HAL
open science

Pedestrian detection and re-identification using interest points between non overlapping cameras

Omar Hamdoun

To cite this version:

Omar Hamdoun. Pedestrian detection and re-identification using interest points between non overlapping cameras. Computer Vision and Pattern Recognition [cs.CV]. École Nationale Supérieure des Mines de Paris, 2010. English. NNT : 2010ENMP0055 . pastel-00566417

HAL Id: pastel-00566417

https://pastel.hal.science/pastel-00566417v1

Submitted on 16 Feb 2011

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.

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

MINES ParisTech

École doctorale n° 432 : "Sciences des Métiers de l'Ingénieur"

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

l'École nationale supérieure des mines de Paris

Spécialité " Informatique temps réel, robotique et automatique "

A présenter et soutenir publiquement par

Omar HAMDOUN

Détection et ré-identification de piétons par points d'intérêt entre caméras disjointes

Directeur de thèse : **Fawzi NASHASHIBI**
Co-encadrement de la thèse : **Fabien MOUTARDE**

Jury

Mme. Bernadette DORIZZI	Rapporteur
M. Frédéric JURIE	Rapporteur
M. François BREMOND	Examinateur
M. Samuel VINSON	Examinateur
M. Fabien MOUTARDE	Examinateur
M. Fawzi NASHASHIBI	Examinateur

T H È S E

MINES ParisTech
Unité Maths et Systèmes

À mes parents

Remerciements

Mes plus sincères remerciements vont à Monsieur Fabien Moutarde qui a dirigé cette thèse. Je voudrais le remercier pour son œil critique pour structurer le travail et pour améliorer la qualité des différentes sections mais aussi pour le temps et la patience qu'il m'a accordé tout au long de ces années.

Je voudrais également remercier de tout mon cœur Monsieur Fawzi Nashashibi, pour ses précieux conseils et son soutien moral tout le long de la thèse. Il a toujours répondu présent quand j'en avais besoin, malgré un emploi du temps très chargé.

Je remercie Madame Bernadette Dorizzi, professeur à Télécom SudParis, pour l'intérêt qu'elle a apporté à mon travail en me consacrant de son temps en tant que rapporteur sur le présent manuscrit.

Je remercie également Monsieur Frédéric Jurie, professeur à l'université de Caen pour le rapport de thèse. Sa gentillesse m'a beaucoup touché. La justesse de ses remarques m'a encouragé à être plus précis dans mes futures recherches.

Je remercie Monsieur François Bremond et Monsieur Samuel Vinson pour leur participation au jury de thèse et leur intérêt pour mes travaux. Leurs questions et remarques très judicieuses ont amenés des discussions très intéressantes.

Je remercie Monsieur Claude Laurgeau et Monsieur Arnaud de la Fortelle de m'avoir accueilli si chaleureusement dans leur laboratoire.

Je remercie également Monsieur Bruno Steux, co-auteur du logiciel RTMAPS et créateur de la bibliothèque Camellia (qui ont été utilisée durant la thèse), pour son aide et ses conseils sur l'utilisation de Camellia.

Je remercie Fatin Zaklouta qui a pris le temps de m'aider pour la relecture de ma thèse et pour mes publications; je lui suis également reconnaissant pour les nombreuses discussions utiles que nous avons eues.

Au cours de ma thèse au laboratoire de robotique, j'ai eu la chance de rencontrer des gens merveilleux et intéressants qui m'ont inspiré ou contribué à la réalisation de ma thèse de doctorat. Je remercie en particulier Safwan Chendeb, Bogdan Stanciulescu, Amaury Breheret et Raoul De Charette, qui ont toujours accepté de répondre à mes questions ; leur aide a été précieuse pour la réalisation de ma thèse.

Enfin je remercie ma famille en Syrie et en France pour leur soutien ainsi que mes amis qui étaient à mes côtés lors de mes aventures au pays des Gaulois.

Résumé

Avec le développement de la vidéo-protection, le nombre de caméras déployées augmente rapidement. Pour exploiter efficacement ces vidéos, il est indispensable de mettre au point des outils d'aide à la surveillance qui automatisent au moins partiellement leur analyse. Un des problèmes difficiles et encore mal résolu de ce domaine est le suivi de personnes dans un grand espace (métro, centre commercial, aéroport, etc…) couvert par un réseau de caméras sans recouvrement.

Dans cette thèse nous proposons et expérimentons une nouvelle méthode pour la ré-identification de piétons entre caméras disjointes. Notre technique est fondée sur la détection et l'accumulation (durant le suivi sur une caméra) de points d'intérêt caractérisés par un descripteur local.

La détection des points peut se faire sur la région d'intérêt obtenue par les approches standards de soustraction de fond et de détection de mouvement au niveau pixel. Mais nous proposons aussi une transposition de cette étape exploitant uniquement les points d'intérêts, ceci dans l'optique de concevoir une chaîne complète de traitement adaptée à de futures caméras intelligentes qui calculeraient elles-mêmes les points d'intérêt sur hardware intégré et les fourniraient en sortie.

Dans notre approche, la ré-identification se fait en collectant un ensemble de points d'intérêt durant une fenêtre temporelle, puis en cherchant pour chacun d'eux leur correspondant le plus similaire parmi tous les descripteurs enregistrés précédemment, et stockés dans un KD-tree. Ceci permet une recherche très rapide même parmi un très grand nombre de points correspondant à de nombreuses personnes suivies antérieurement dans d'autres caméras.

Nous effectuons des évaluations quantitatives de notre technique sur des bases publiques telles que CAVIAR et ETHZ, ainsi que sur un nouveau corpus de vidéos contenant 40 personnes, que nous avons enregistré dans notre laboratoire, et que nous proposons comme benchmark à la communauté. Les performances de ré-identification de notre algorithme sont comparables à l'état de l'art, avec environ 80% d'identification correcte au premier rang.

Nous présentons aussi des comparaisons avec d'autres descripteurs couramment utilisés (histogramme de couleur, HOG, SIFT), qui obtiennent de moins bons résultats. Enfin, nous proposons et testons des pistes d'amélioration, en particulier pour la sélection automatique des instants ou des points d'intérêt, afin d'obtenir pour chaque individu un ensemble de points qui soient à la fois les plus variés possibles, et les plus discriminants par rapport aux autres personnes. Cette variante probabiliste de notre méthode, qui pondère les points d'intérêt selon leurs fréquences relatives dans chaque modèle et globalement, apporte une très importante amélioration des performances, qui augmentent jusqu'à 95% d'identification correcte parmi 40 personnes, ce qui dépasse l'état de l'art.

Plan du mémoire

Ce manuscrit est découpé en quatre chapitres :

- *Le premier chapitre a pour objectif de définir et expliciter les termes scientifiques utilisés dans la suite du document, et de donner un aperçu synthétique de l'état de l'art existant pour les systèmes intelligents multi-caméras.*
- *Le second chapitre introduit les différentes primitives et signatures que nous avons étudiées et implémentées dans nos travaux. Dans un premier temps, nous décrivons les signatures globales et leurs caractéristiques. Puis, nous présentons les primitives locales, en particulier les points d'intérêts et leurs descripteurs.*
- *Le troisième chapitre propose puis évalue une méthode utilisant les points d'intérêts pour la modélisation de scène, puis la détection d'objets mobiles. Afin de modéliser la scène nous utilisons une méthode hybride consistant en la modélisation de l'arrière-plan par un ensemble de points, puis le filtrage par classification AdaBoost des points de premier-plan restant après la soustraction des « points d'arrière plan ».*
- *Le quatrième chapitre détaille la méthode que nous avons mise au point pour la ré-identification de personnes par appariement de points d'intérêts dans plusieurs images. Nous quantifions les performances de notre méthode, et effectuons ainsi une comparaison objective avec les autres signatures existantes (SIFT, Couleur, HOG).*

ABSTRACT

With the development of video-protection, the number of cameras deployed is increasing rapidly. To effectively exploit these videos, it is essential to develop tools that automate the monitoring at least part of their analysis. One of the difficulties and poorly resolved problems in this area is the tracking of people in a large space (metro, shopping center, airport, etc ...) covered by a network of non-overlapping cameras.

In this thesis, we propose and experiment a new method for the re-identification of pedestrians between disjoint cameras. Our technique is based on the detection and accumulation (during tracking within one camera) of interest points characterized by a local descriptor.

Point detection can be done on the region of interest obtained by the standard approaches of pixel-wise background subtraction and motion detection. But we also propose an implementation of this phase using only interest points, to develop a complete processing chain adapted to future smart cameras that would calculate interest points in real-time on integrated hardware and provide them as complementary or alternative output.

In our approach, re-identification is done by collecting a set of interest points during a time window, then looking for each of their corresponding most similar amongst all descriptors previously stored in a KD-tree. This allows a very fast search even among a large number of points corresponding to many people followed previously in other cameras.

We conduct quantitative evaluations of our technique on public databases such as CAVIAR and ETHZ, and a new set of videos containing 40 people, that we recorded in our laboratory, which we propose as a benchmark for the community. The performance of re-identification of our initial algorithm is at least comparable to the state of the art, with approximately 80% of correct identification at first-rank on our 40-persons database.

We also present comparisons with other commonly used descriptors (color histogram, HOG, SIFT), which have a lower performance. Finally, we propose and test possible improvements, particularly for the automatic selection of moments or interest points, to obtain a set of points for each individual which are the most varying and more discriminating to those of other people. This probabilistic variant of our method, which weights keypoints according to their relative frequencies within models and globally, brings tremendous improvement to performance, which rise at 95% correct identification among 40 persons, which is above state-of-the-art.

Manuscript outline

The manuscript is organized in four chapters:

- *The first chapter aims at defining and clarifying scientific terms that will be used in the rest of this dissertation and gives a wide and synthetic overview of the state of the art approaches in distributed smart cameras systems.*
- *The second chapter introduces the concepts and different features that we have studied and implemented in our work. First, we describe the global features, their principles and their issues. In the second part, we present local features defined by interest points.*
- *The third chapter presents and evaluates a keypoints-based method for modeling a scene and detecting new objects in it. The scene is modeled using a hybrid method consisting of background modeling with keypoints, and filtering these points using an Adaboost classifier.*
- *The fourth chapter presents and evaluates our method for identifying a person by matching the interest points found in several images. We produce quantitative results on the performance of such a system to allow an objective comparison with other features (SIFT, Color, HOG).*

Contents

Résumé **1**
Plan du mémoire 6
ABSTRACT **7**
Manuscript outline 8
1 What is IDVSS? **13**
Motivation 13
Definition 13
1.1 Techniques used in surveillance systems 15
1.1.1 Object detection 15
1.1.2 Object classification 16
1.1.3 Tracking 17
1.1.4 Understanding 18
1.1.5 Databases and semantic description 18
1.2 Intelligent Distributed Video Surveillance Systems 19
1.2.1 Methods for tracking with overlapping multi-cameras 27
1.2.2 Methods for tracking with non-overlapping multi-cameras 31
1.3 Conclusion 34
2 Visual features **35**
Motivation 35
Introduction 35
2.1 Feature types 35
2.1.1 Shape features 35
2.1.2 Contour features 36
2.1.3 Texture features 36
2.1.4 Color features 38
2.2 Representation of global features 38
2.3 Local Features 39
2.3.1 The principle of interest points 39
2.3.2 Interest points detector 40
2.3.3 Efficient implementations of IP detectors 46
2.3.4 The interest points descriptors 50
2.3.5 Choice of descriptors and detector 54

2.4 Conclusion 56

3 Pedestrian Detection with keypoints 57

Motivation 57

Introduction 57

3.1 Related works 58

3.1.1 Background substraction 59

3.1.2 Statistical Methods 60

3.1.3 Spatial-based Background Models 61

3.2 The proposed method 61

3.2.1 Construction of the initial background 62

3.2.2 Adaboost Classification 66

3.2.3 AdaBoost training algorithm 68

3.3 Results 70

3.3.1 Background Substraction results 72

3.3.2 Keypoint classification results 73

3.3.3 Evaluation of Keypoints filtering with Adaboost only 78

3.3.4 Evaluation of the cascade filtering 80

3.4 Comparison with HoG detector 83

3.5 Conclusion 84

4 Re-identification 85

Motivation 85

Introduction 85

4.1 Related works 85

4.1.1 Template matching methods 86

4.1.2 Color histogram based methods 88

4.1.3 Local features based methods 90

4.1.4 Kd-Tree search 92

4.2 The proposed algorithm 94

4.2.1 Schema of the algorithm 94

4.3 Experimental evaluation 98

4.3.1 Evaluation metrics 98

4.3.2 Caviar Dataset 98

4.3.3 Our newly built corpus with 40 persons 104

4.3.4 Individual recognition analysis 110
4.3.5 NNDR ratio influance 116
4.4 Proposed improvement for the re-identification 117
4.4.1 Construction of the model 117
4.4.2 Construct the query 121
4.4.3 Experimental Results 124
4.5 Experimental Results using ETHZ 128
4.6 Comparison with other region descriptors 132
4.6.1 Comparison SURF vs. SIFT 132
4.6.2 Comparison SURF vs. COLOR 134
4.6.3 Comparison HOG vs. SURF 140
4.7 Comparison KD-tree vs. hierarchical k-means tree 144
4.8 Conclusions 145
5 Conclusion & Perspective 147
5.1 Summary 147
5.2 Future work. 148
6 Publications 149
7 Bibliography 151

1 What is IDVSS?

Motivation

This chapter is intended to meet two objectives. First, it aims at defining and clarifying scientific terms that will be used in the rest of this dissertation. It is important to define the terms of video surveillance in order to fully understand the scope of this dissertation. Second, it gives a wide and synthetic overview of state of the art approaches in distributed smart cameras systems.

Definition

The Surveillance is the process of collection, collation, analysis, and dissemination of data

Since the late 90s, the digitization of content and the advance of computing power made possible the real-time processing of video images to extract interpretations (what do you see in the image?, what's going on?, who goes where? ...).

The recent interest in monitoring public places increases the necessity to create and to develop intelligent video surveillance systems with some automated analysis. These systems can help monitor and record the situations which affect the public interest. The research in this domain tends to combine with the signal and image processing, telecommunication, database.

The objective of traditional video surveillance system is to give to the operator who surveys: the flow of videos , in real time, and recognize the persistent and transient objects and their actions within a specific environment. The main objective of intelligent video surveillance system (IVSS) is to provide an automatic interpretation of scenes to reduce the redundant information and make it possible to understand the actions and interactions of the objects automatically, and then to predict the next situation of objects observed based on the information acquired by the cameras.

In the state of the art the video surveillance systems have gone through three generations depending on the technology used in these systems, as in (Valera, et al., 2005):

- The first generation figure. 1.1 based on closed circuit TV (CCTV): these systems consist of analog cameras situated in various locations, human operators watch the screens onto which the cameras transmit. The main problem of this technology is that the analog devices need consume a lot of resources. However this technology stays the most reliable technology.
- The second generation figure. 1.2 of video surveillance systems mixes the analog and digital devices to overcome the storage capacity drawback of CCTV, and it then tries to automate these systems by integrating detection and tracking algorithms. This integration requires robust and flexible algorithms.
- The third generations figure. 1.3 has completed the digitization of video surveillance systems by using digital cameras which compress the video data and transmit compressed data via the networks. These networks authorize the distribution of data and its analysis. And the

system trend to include many algorithms of computer vision for object recognition, activities and behavior analysis.

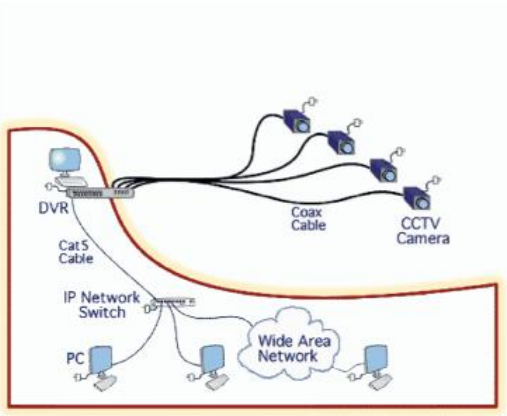


Figure 1-1: First generation of IVSS

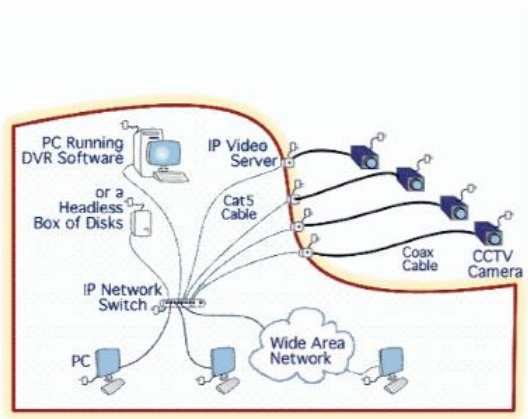


Figure 1-2: Second generation of IVSS

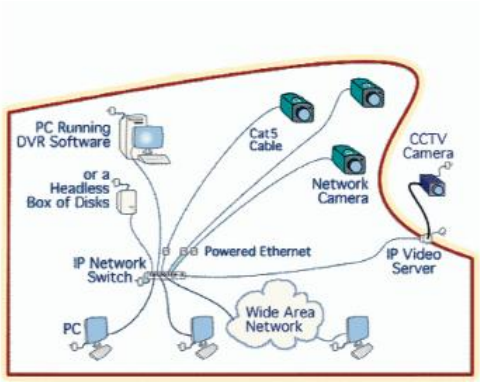


Figure 1-3: Third generation of IVSS (figures extracted from (Bramberger, 2004) *)*

As it is well known, the video surveillance systems differ in their use of computer vision, due to the particularity of the application. For example some systems identify moving objects and then classify them to select only useful targets (e.g., vehicles and people). Others systems, start with target detection and then track selected objects only. Apart from some initial image enhancement and image processing steps, there are five main steps in any surveillance system (Dedeoglu, 2004).

- Object detection.
- Object recognition.
- Tracking.
- Action recognition.
- Semantic description of scenes.

These steps are the themes of computer vision, artificial intelligence, data management and communication. The traditional pattern of treating a surveillance system is shown in figure 1-4

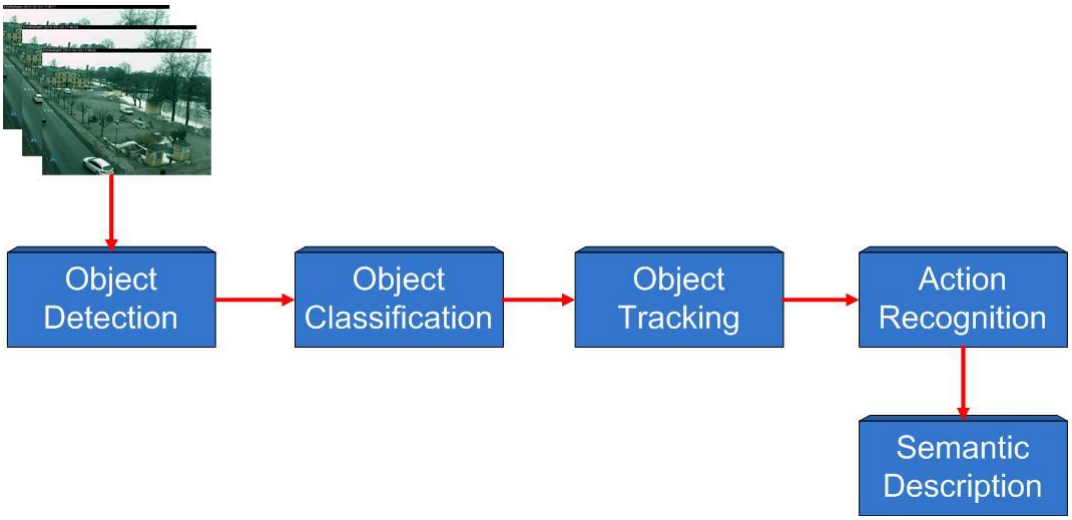


Figure 1-4: A generic framework for video surveillance system (figure extracted from (Dedeoglu, 2004))

1.1 Techniques used in surveillance systems

This section summarizes research that addresses the main image processing tasks which constitute the low-level building blocks necessary for any distributed surveillance system. It is crucial to note that the five perceptual tasks presented in this section are the five facets of the same global problem. The computation of any given task is therefore highly dependent of the other perceptual tasks.

1.1.1 Object detection

Determine the regions of interest in the image which belong to a predefined object in the scene.

The object detection is the first step of video surveillance systems. It plays a very important role in the video surveillance system since the result of this step will affect all subsequent steps. The objective of this step is to extract the region of interest.

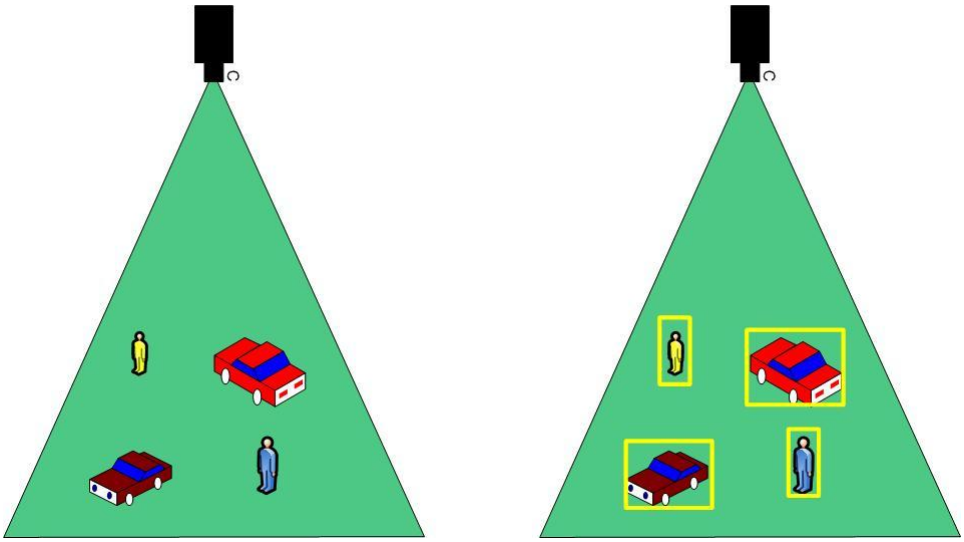


Figure 1-5: A schematic view of a detection process from Image.

Motion detection is a non-trivial problem and many methods have been proposed. The common step of all these method is background modeling. It consists of a training stage which tries to adapt with the dynamic changes in natural scenes such as variation in illumination, weather changes and repeated motions that cause clutter (waving of tree leaves). However, these techniques differ in the methods of representation of the modeling in the scale of pixels, blocks of pixels or the cluster of pixel. On the other hand these methods are classified according to the way they update the background model.

1.1.2 Object classification

Determine the "type" of all the objects present in the scene.

Once the objective of our video surveillance application has been fixed, the types of objects are often predefined, such as vehicles on the road or people in an office, before using an algorithm to classify the objects. It is very important to recognize the type of a detected object in order to track reliably and analyze their activities correctly. Object classification can be considered as a standard pattern recognition task. There are two main categories of approaches for classifying moving objects (Wang, et al., 2003):

- **Shape-based classification.**
- **Motion-based classification**.

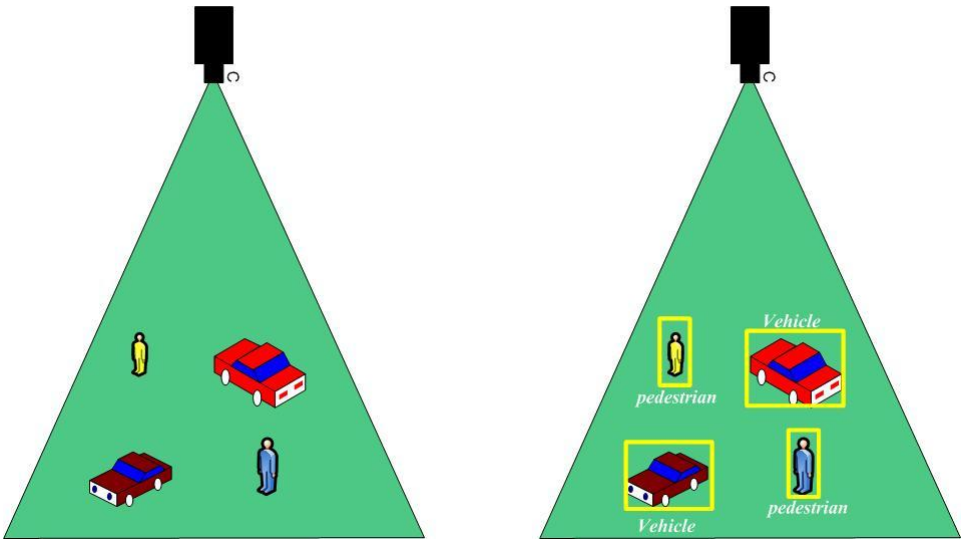


Figure 1-6: Schematic view of a classification process from image

These approaches differ in the nature of the features used in the classification. Shape-based classification uses the shape information such as points, boxes, silhouettes and blobs. This approach depends heavily on the accurate computation of the detection task. Motion-based classification uses temporally tracked features of objects for the classification decision.

Object classification can be performed by a supervised learning mechanism, such as Adaptive boosting *(Adaboost)* (Viola, et al., 2001) or support vector machines *(SVM)* (Dalal, et al., 2005).However these methods usually require a large collection of samples from each object class.

1.1.3 Tracking

Estimate the trajectory of an object in the image as it moves in the scene

The objective of an object tracker is to find the trajectory of an object by locating its position in every frame of the video. Another objective of video tracker is to give the complete region in the image that is occupied by the object at every time instant (Yilmaz, et al., 2006). There are two important factors in the tracking methods: the features used to model the object (object model) and the alignment of these features at every time (correspondence methods). Tracking in video can be categorized according to the model used or according to the correspondence methods.

Object modeling plays a critical role in tracking. Object modeling is closely related to the needs of the application it is used in. For example, the skeletal model is used for outdoor video surveillance whereas the articulated shape model is necessary for some indoor surveillance and high level action recognition applications. In some cases, the methods of tracking use explicit a priori geometrical knowledge of the objects to follow, which in surveillance applications are usually people, vehicles or both (Valera, et al., 2005).

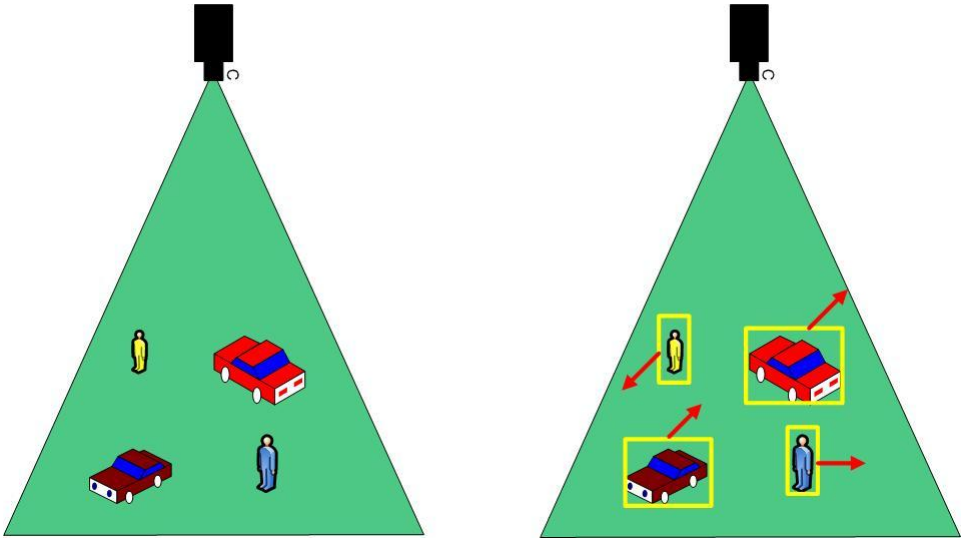


Figure 1-7: Schematic view of a tracking process from image.

Correspondence methods are the second important factor to predict the position of the object and its velocity. They reduce the search space and the probability of ambiguity. These methods are split into two categories: deterministic and statistical methods. Deterministic methods attempt to minimize the cost function, associating each object in frame $t-1$ to a single object in frame t using a set of motion constraints. These methods assume the location of the object does not change notably from one frame to other. The statistical method use same constrains. However, the object properties like position, velocity and acceleration are modeled as state space (Yilmaz, et al., 2006).

There are still many unsolved problems in tracking and even more partially solved issues, such as handling occlusions, variable lighting conditions, low or uneven frame rates, insufficient resolution and tracking very deformable objects.

1.1.4 Understanding

Exploit the world model to recognize actions, events and behaviors

The next step in a surveillance system is to recognize and interpret the activities and behaviors of tracked objects. This step focuses on actions and does not explicitly consider the context such as the environment and interactions between persons or objects. It is based on classification of time-varying signatures that are provided by the previous steps. There are two important factors in the behavior understanding methods: learning the reference behavior sequences from training samples, and development a robust method to deal effectively with small variations of the feature data within each class of motion patterns (Hu, et al., 2004).

There are several approaches to match the temporal data. For example, DTW (dynamic time warping) is widely used in speech recognition, in image recognition (Kovacs-Vajna, 2000) and recently in recognition of the pedestrian movement (Junejo, et al., 2008). Although this is a reliable technique, it is now less favored than the network of dynamic probabilistic models such as HMM (Hidden Markov Models) in (Meyer, et al., 1998) and Bayesian networks (Chen, et al., 2007).

The methods of action recognition are categorized into two main approaches according to the representation: global and local. The first is powerful since they encode much of the information. However, it relies on accurate localization, background subtraction or tracking. It is also more sensitive to viewpoint, noise and occlusions. Whereas local representations such as spatiotemporal features are less sensitive to noise and partial occlusion, and do not require background modeling or tracking. However, it depends on the extraction of a sufficient amount of relevant interest points (Poppe, 2010).

1.1.5 Databases and semantic description

Build and describe the behavior of each object in the scene

The last step in a video surveillance system is the storage. Little research has been done in the field of efficient storage of information obtained by a surveillance system. The semantic description aims to reduce the large amounts of video data stored for future or immediate use. The methods of semantic description are divided into two main approaches (Hu, et al., 2007): motion-based video retrieval and the semantic description of object motions.

1.2 Intelligent Distributed Video Surveillance Systems

Intelligent distributed surveillance systems are real-time systems, based on techniques from computer vision, distributed computing, and distributed sensor networks for monitoring large areas .

A distributed system is a system where necessary treatments or calculations are not centralized (i.e. on a single computing unit), but spread over several computational units connected by a communication medium such as an Ethernet network. Figure 1-8 represents the general structure of the video surveillance system proposed in (Hu, et al., 2004).

This system has to be able to detect and track objects of interest and reliably, taking into account the lighting conditions, weather, change in camera view angle and the presence of non-overlapping cameras (i.e. with view field that are not intersecting).

This system is called intelligent when these units are able to cooperate with a communication protocol level. The result is the output of the organization of communication between different programs running on different computing units.

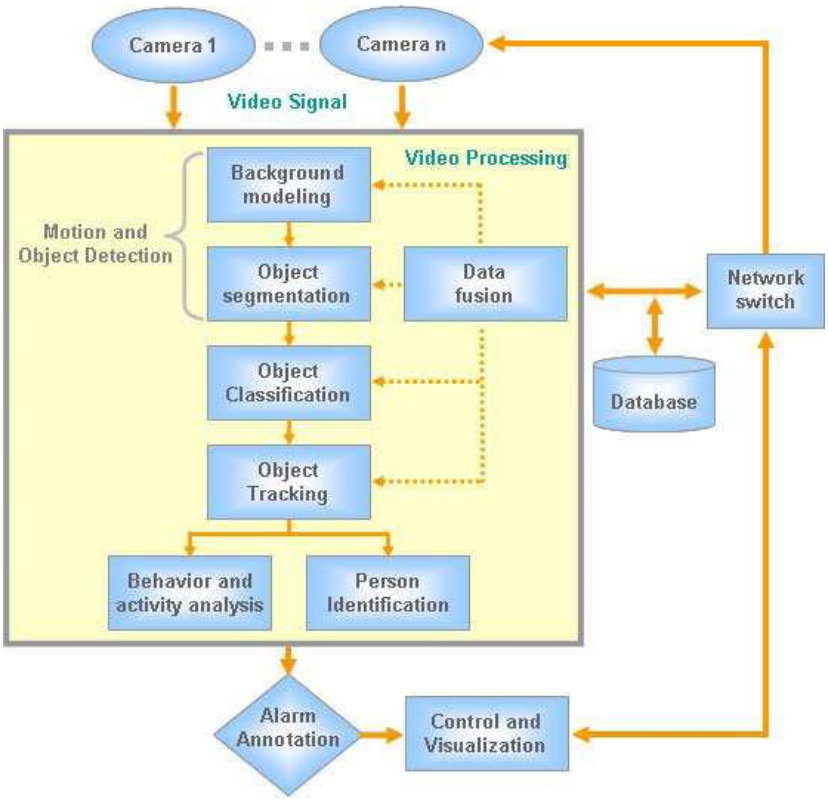


Figure 1-8: A general framework of an automated visual surveillance system (Ko, 2008)

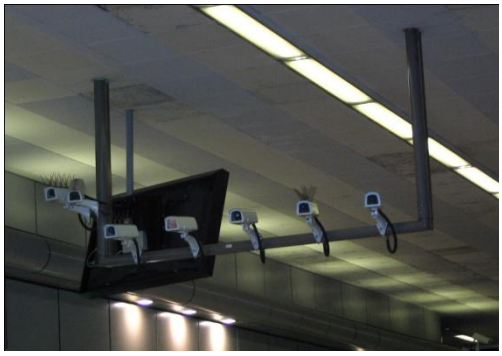


Figure 1-9: Due to the exhaustive use of cameras, there is an information overload: The cost of relevant sequences from a large number of cameras is too high. For example, a 200 camera installation at 6 images/second generates 100 million images per day and 2.1 billion images in the database within 3 weeks.

The previous section presented some computer vision techniques necessary for the detection and understanding of the activity in the context of monitoring. It is important to emphasize that the existence of these techniques is necessary, but not sufficient to serve a potentially vast surveillance system, which consists of networks of cameras and distributed computing units, as in airports or highways. The exhaustive coverage of the entire region of interest is typically impossible due to the large size of the field monitored and the limited number of surveillance cameras. This leads to two main problems:

- On one hand, the attention of most operators falls bellow an acceptable level after only 20 minutes of watching and analyzing video surveillance as it is showed in several studies[1] (figure 1-9). The maximum period during which a person can attentively monitor 9 to 12 cameras is 15 minutes. The ratio between the number of screens and the number of cameras can be between 1:4 and 1:78. The studies show that the probability of immediately dealing with an event captured by a surveillance camera network is approximated to 1 over 1000 (Gouaillier, et al., 2009).

- On the other hand, just because a surveillance system is depending on digital network technology (IP camera, Network video recorder…), it does not mean the architecture is suitable. Consider for example, Aéroports de Paris Group[2] , Europe's second largest airport group, managing airports, and aerodromes including Paris-Orly, Paris-Charles de Gaulle, and Paris-Le Bourget. As the airport authority for the Paris region, Aéroports de Paris view security operations as an integral part of their customer services and ensure a high level of security to safeguard passengers, airline companies and partners. In 2003, Aéroports de Paris launched a CCTV upgrade program to improve the level of security and safety throughout their airports that still continues today. Since 2003, Aéroports de Paris have supplied over 5,600 video ports helping to create one of the largest CCTV networks in Europe. Operators are able to access, share, and view live and recorded video surveillance sequences from thousands of cameras. Another example of the scale of the redundancy is the system deployed in Athens for the 2004 Olympics. Figure 1.10 shows the Command Center Structure used during the games. There were 1250 operators in 63 command centers monitoring 47 sporting events spread out over an area of 250 square kilometers. Operators worked for many different agencies each with their own interest in the CCTV video feeds. Law enforcement, emergency services, military, traffic management, coastguard and local security all required, to some degree, access to all or part of the system and so every operator had to be given unique access rights to particular components. This provided a high degree of redundancy.

[1] US National Institute of Justice study, quoted
[2] http://www.controlware.co.uk/en/case-studies/aeroports-de-paris.html

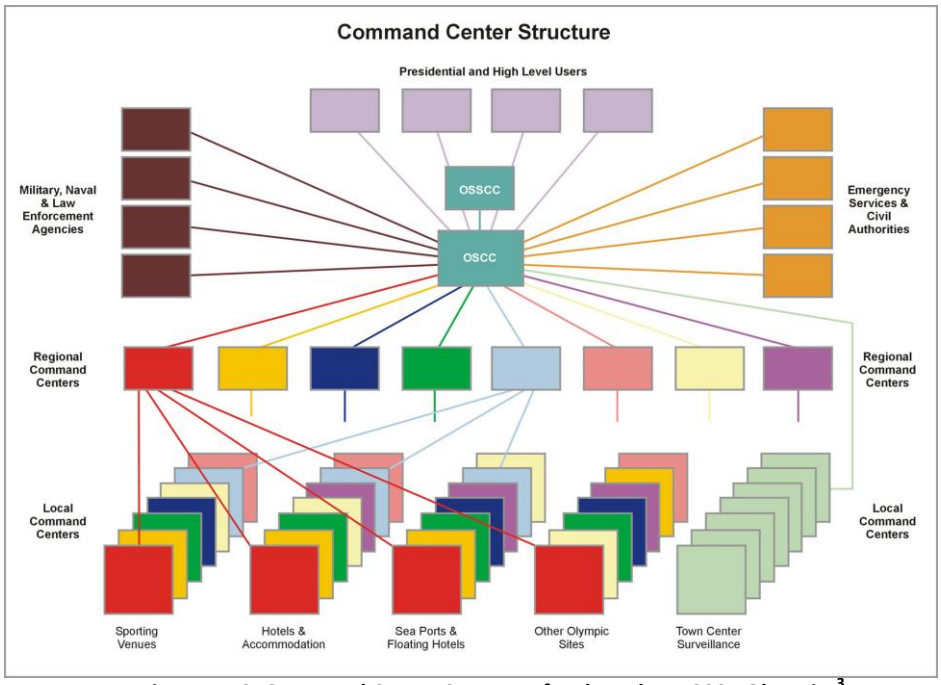


Figure 1-10: Command Center Structure for the Athens 2004 Olympics[3]

These systems pose an additional problem considering the appearance of an object in several video streams. When the cameras observe the same scene, the object can appear simultaneously in the sequences and the association may be simplified by the techniques of camera calibration ((Jaynes, 2004), (Tu, et al., 2007), (Jain, et al., 1995), (Mittal, et al., 2003)). But in general, each camera observes a scene disjoint from the scene observed by the other cameras .Each field-of-view covers only a relatively small scene. So the total scene will not be contiguous. In this case, the problem of association is generally more difficult and has not been studied as widely.

The work presented in this thesis focuses on the problem of the re-identification of persons for the distributed systems of non overlapping cameras. More precisely, we aim at recognizing objects by matching sequences when different observations show large variations. Conventionally, in pattern recognition, there is a first step of characterization or recognition learning. The object is tracked and characterized on a first sequence. The objective of this learning is to select a set of information (radiometry, shape information, dynamics...) concerning the identity of the object. The second step, called the test step, intervenes with the recognition. We must determine whether or not an object has previously been characterized, despite the changes in camerawork conditions:

- Change in the appearance of the object due to its nature like the silhouette.
- Variance of camera settings like the sensitivity of the camera sensor, resolution and focal length.
- Illumination variation.
- Different poses of an object in different instances of tracking.

The research on intelligent distributed surveillance systems focuses on two principal problems in the domain of computer vision. The first problem is the multi-camera tracking or reacquisition, which

[3] http://www.indigovision.com/learnabout-olympics.php 2009

depends on the time for tracking the object between cameras. These algorithms try to predict and associate the object based on its movement. Most of the work concentrates on methods of movement estimation and predicting the trajectory of the object. The second problem which is not different from the reacquisition, tries to associate the object between the cameras without taking into account the movements of the object. In both cases, there is no way to associate the objects between any two cameras without comparing the appearance of the objects in both cameras. Therefore, it is very difficult to distinguish between the two problems. We will start with the state of art of multi-camera tracking although our work is the re-identification. In both cases, we will focus on the features and the association between these features.

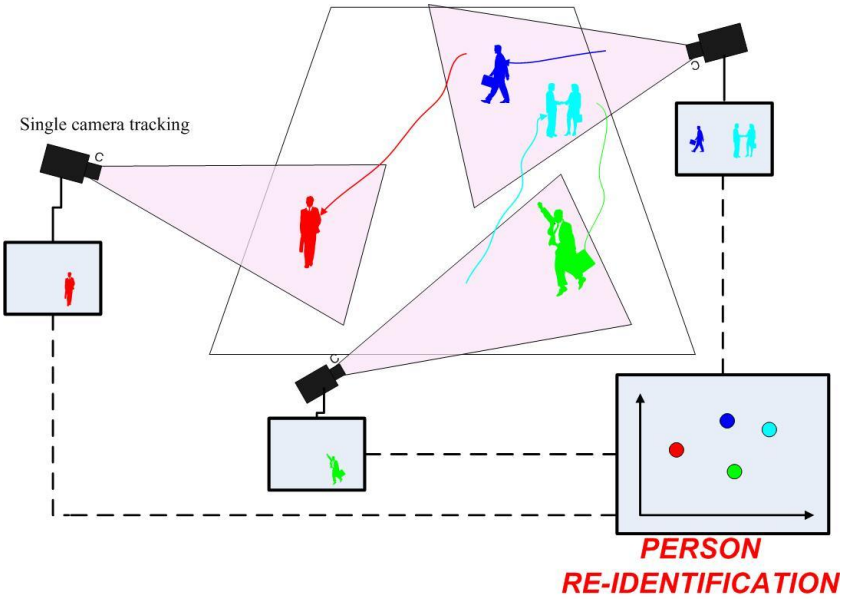


Figure 1-11: A Diagram of person re-identification

The first category of person re-identification methods relies on biometric techniques (such as face or gait recognition). Face identification in the “cooperative” context on high-resolution images with well-controlled pose and illumination can now be done with very good performance see (Belhumeur, et al., 1997) or (Chellappa, et al., 2007). However, on-the-fly face matching in wide-field videos is still a very challenging problem. Gait recognition aims at identifying the people by their manner of walking. Most works assumed that people walk perpendicularly. These methods of identification are out of the focus of this thesis.

The second category of person re-identification is based on the global appearance of the object, which we are interested in studying in this thesis. Most systems for tracking in multiple cameras or re-identification work in three stages as we can see in figure 1-12. The first step is the detection of the object in each camera. The second is the extraction of the discriminative features of these objects or the construction an object model, which consists of its spatial and temporal information. Finally, the data association is performed. In some application, the location of object is necessary to detect and construct the object model. Hence, we can predict its trajectory. Some applications start by modeling the environment using calibrated cameras

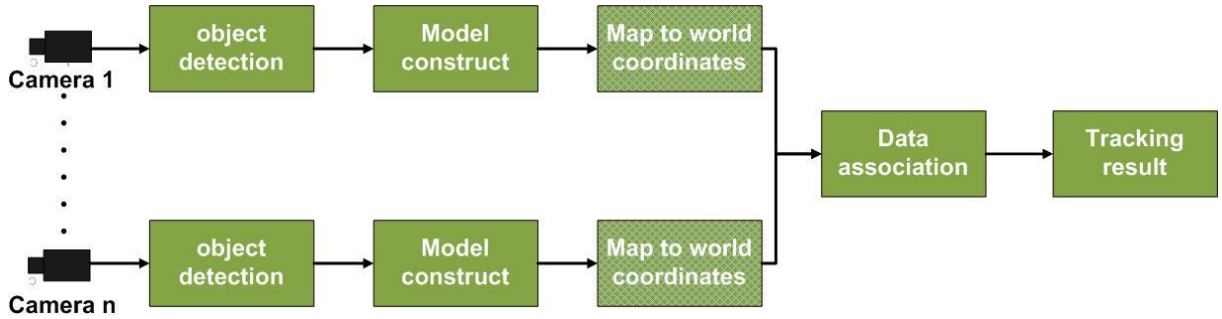


Figure 1-12: A block diagram for tracking in multiple cameras. This block diagram consist of three stages: First object detection, second stage model construction and third stage data association

The methods of identification in multi-camera video surveillance systems are differentiated according to their basic assumptions about the overlapping of the field of view. Fig.1.14 shows four different types of overlapping relationships that a network of cameras could have: using calibration or learning the inter-relationship cameras, using of 3D position of objects, and / or signatures used to establish connections. We divide the literature of multi-camera tracking into two main categories based on the specifications of the relationship between the fields of cameras. It is important to define the perception vocabulary first in order to fully understand the methods of tracking.

Environment and field of view

The environment refers to the context within which cameras acquire the images. Note that cameras will rarely be able to collect images of its whole environment. This is why we define the regions that an individual camera will survey by its field of view. In the cases where the camera view extends to the horizon, a practical limit on the view range is imposed by the finite spatial resolution of the camera or by the minimum size of reliably detectable objects.

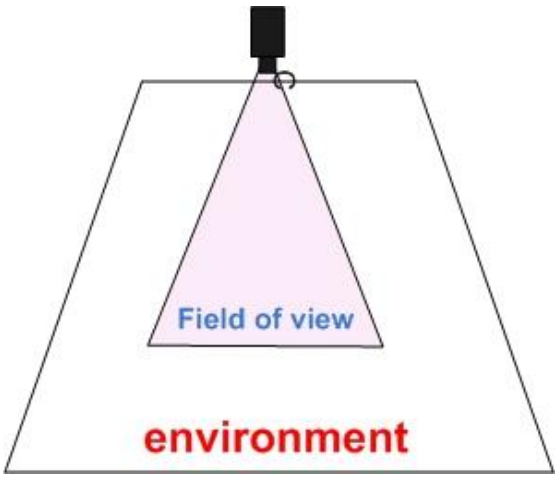


Figure 1-13: The portion of the environment that is visible from a camera is referred to as scene in this dissertation.

Type of overlapping

1. Non-overlapping fields of view: it is assumed that no two cameras observe the same parts of the scene like in the system of video surveillance installed in buildings. In this case, it is very difficult to observe the entire building using cameras. There are some parts of the environment which are not covered by the cameras. We have to estimate the trajectory of the objects in these cases.

2. Overlapping fields of view: it is supposed that the cameras observe different parts of the same scene as figure 1-14 (b) illustrates. The overlapping views increase the target's pixels. Our ability to analyze a subject is limited by the amount of information, measured in pixels, which increases when we see the same object from different view angles (Rinner, et al., 2008).

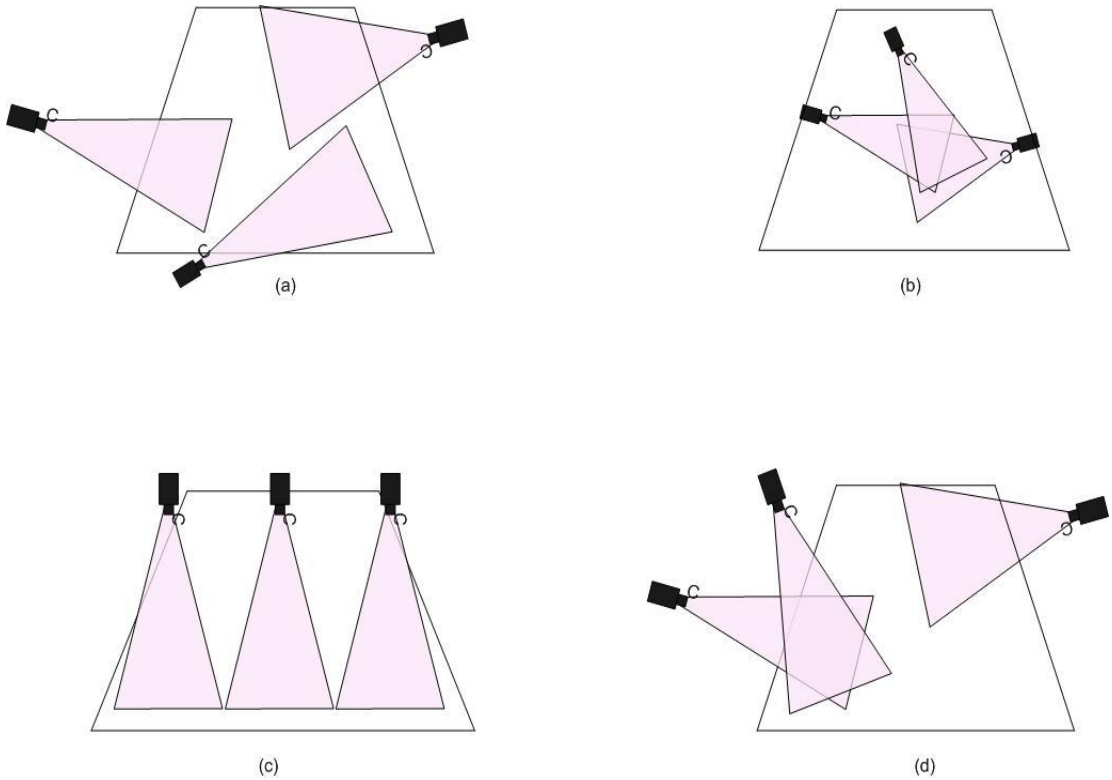


Figure 1-14: Four different cases overlap in tree cameras: (a) no region of overlap between any cameras, (b) every camera overlapping with every other camera, (c) disjoint cameras and (d) the most general case with all previous types of overlap.

Camera calibration

Camera calibration gives the parameters that model the relation between the points lying on the plane of camera to 3-D points in a fixed world coordinate system to 2-D points on an image plane. These calibration parameters can be divided into Intrinsic Parameters of camera such as image center, aspect ratio, lens distortion, and focal length, pixel sizes, and extrinsic parameters namely the rotation angles and translation vector that find projective transform from the camera coordinate plane to the world coordinate frame .

Homography Transformation

The objective of homography transformation is projecting the points from the plane of one camera viewpoint to another by using standard projective geometry notation. The homography transform between cameras is available when the ground plane is visible in each view as shown in figure 1-15.

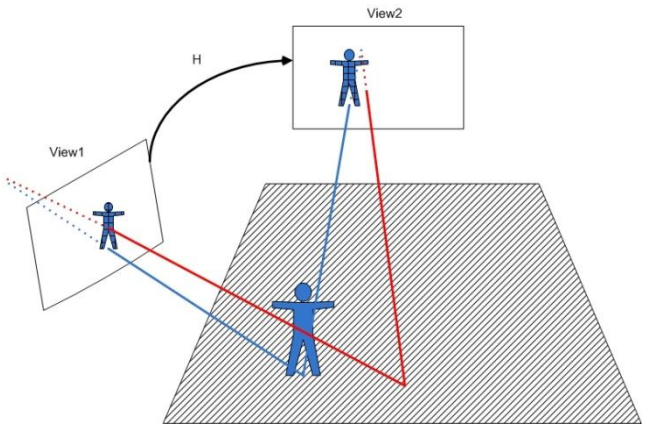


Figure 1-15: the figure shows an object standing on top a planar surface. The scene is being viewed by two cameras. H is the homography of the planar surface from view 1 to view 2. (Khan, et al., 2006)**.**

Epipolar Geometry

"The epipolar geometry is the intrinsic projective geometry between two views. It is independent of scene structure, and only depends on the cameras' internal parameters and relative pose. The epipolar geometry between two views is essentially the geometry of the intersection of the image planes with the pencil of planes having the baseline as axis" (the baseline is the line joining the camera centers). This geometry is usually motivated by considering the search for corresponding points in stereo matching (Hartley, et al., 2004) . This has the benefit that the ground plane constraint is not valid but this method has increased ambiguity if several objects lie on the epipolar line. The epipolar geometry is graphically shown in figure 1-16.

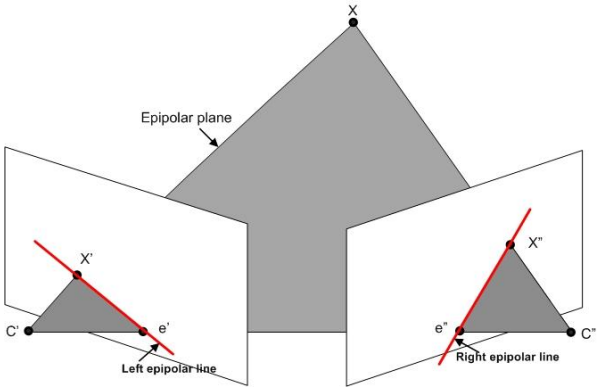


Figure 1-16: The epipole geometry between a pair of camera views.

1.2.1 Methods for tracking with overlapping multi-cameras

A lot of works on multi-camera surveillance requires overlapping fields. Using multi-camera with overlapping fields of view allows to see the object from several different angles. This, in turn, helps to solve the problem of occlusion. A synthesis of methods is presented in table 1-1, and more details are gives below.

- Jain and Wakimoto (Jain, et al., 1995) develop an approach for person tracking using calibrated cameras and computing 3D environments model. The alignment of the viewpoints of the same person depends on the absolute position of objects in 3D environments.
- Cai et al (Cai, et al., 1999) use calibrated cameras to track the object. Their approach consists of tracking the person using a single camera and the switching across cameras is done when these no longer have a good view of the object. Selecting the switched cameras depends on a simple threshold of the tracking confidence. They model three types of features: location, intensity and geometry using a multi-variant Gaussian. The Mahalanobis distance is used for matching.
- Chang and Gong (Chang, et al., 2001) overcome the occlusion by matching objects across several cameras. They use a Bayesian to fuse the geometric constraints: epipolar geometry, landmarks and homography, with the appearance color model and the height of the object.
- Lee et al (Lee, et al., 2000) use overlapping cameras to track the objects in urban scenarios. They use the intrinsic camera parameters and the alignment of moving object centroids to map these points onto a common frame.
- Dockstader and Tekalp (Dockstader, et al., 2001) use the Bayesian net to iteratively fuse the independent observations from multiple cameras to produce the most likely vector of 3D state estimates for the tracked objects. They use also the Kalman filter to update this estimation. Sparse motion estimation and foreground region clustering are extracted as features.
- Mittal and Davis (Mittal, et al., 2003) segment images in each camera view and track the object using Kalman filter. Then they use epipolar geometry to match the segment of the object to find the depth of the object points. The points are mapped onto the epipolar plane

and the correspondences between the points are achieved with a Bayesian network. The foreground objects are modeled by their color.

- Khan and Shah (Khan, et al., 2003) avoid the calibration using the constraints of the lines of the field of view (FOV). The information of FOV is learned during a training phase. Using this information. They can predict the label of an object that has been followed in a camera, in all other cameras in which the object was visible.
- Kim et al (KIM, et al., 2006) build a colored part-based appearance model of a person. Then the models across cameras are integrated to obtain the ground plane locations of people. The correspondence of a model across multiple cameras is established by mapping the model onto homographies planar to construct a global top view. The result of the projection is extended to a multi-hypothesis framework using particle filtering.
- Gandhi and Trivedi (Gandhi, et al., 2007) use calibrated cameras and a model of the environment for the 3D position of a person (see figure 1.17). This approach captures the information to look up a number of angles and azimuths around the person. It produces a panoramic appearance of the person (Panoramic Appearance Map) which is used for matching.
- Calderara et al (Calderara, et al., 2008) present methods for consistent labeling (tracking). They extend the approach of Khan et al (Khan, et al., 2003) by an offline training process computing homography and epipolar geometry between cameras. The consistent labeling is solved using Bayesian statistics. They use the color appearance model consisting of the centroid, the color template and the mask of object.
- Kayumbi and al (Kayumbi, et al., 2008) use homographic transformation to integrate different views into the large top-view plane. Then they transform the detected player locations from the camera image plane into the ground plane. Finally, the tracking on the ground plane is achieved using graph matching of trajectories of objects.
- Dixon and al (Dixon, et al., 2009) avoid the calibration by using the sheet (sheet is an image constructed from video data extracted along a single curve in the image space over time) to decompose a 2D tracking problem into a collection of 1D tracking problems, exploiting the fact that in urban areas vehicles often remain within lanes. The data association is expressed as MAP (maximum-a-posteriori) estimation by using the min-cost flow formulation.

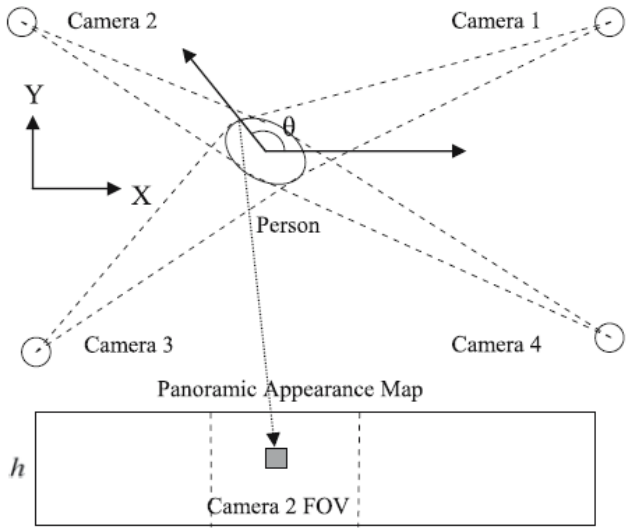


Figure 1-17: **Multi-cameras with overlapping fields of view**
(figure extracted from* (Gandhi, et al., 2007)*)

Most of these methods of monitoring require large overlapping fields of views of the cameras. This need is obviously costly in material resources for monitoring large areas. Moreover, it is preferable that the tracking system does not require the camera calibration or modeling of the whole site, because the luxury of calibrated cameras or a site template is not available in all situations (Khan, et al., 2003). Moreover, for outdoor cameras, it is practically challenging to accurately calibrate them. Therefore, 3D points at a large distance from the camera are difficult to measure accurately (Mittal, et al., 2003).

Table 1-1: Overview of Methods for tracking with overlapping multi-cameras

Reference Paper	Color information	Features	Geometric Information	Calibration
(Jain, et al., 1995)	-	-	3D positions	Yes
(Cai, et al., 1999)	-	intensity and geometric characteristics	-	Yes
(Lee, et al., 2000)	-	centroid of the silhouette	-	Yes
(Chang, et al., 2001)	Color appearance	Object apparent height	Epipolar geometry	NO
(Dockstader, et al., 2001)	-	Patches elliptical	3D projection	Yes
(Mittal, et al., 2003)	Color Region	-	Epipolar geometry and 3d projection	Yes
(Khan, et al., 2003)	-	-	constraints of lines of the field of view (FOV)	NO
(KIM, et al., 2006)	Color region	-	homographies planar	Yes
(Gandhi, et al., 2007)	-	Panoramic Appearance Map	-	Yes
(Calderara, et al., 2008)	Color	centroid	homography and epipolar geometry	NO
(Kayumbi, et al., 2008)	-	trajectories of objects	Homography transform	Yes
(Dixon, et al., 2009)	-	Sheet	-	NON

1.2.2 Methods for tracking with non-overlapping multi-cameras

To track people with non overlapping FOVs, most researchers propose solutions that make assumptions about the movement of the objects between cameras to get a rough estimate of object locations and orientations. They model the knowledge about the environment and locations of cameras to reduce the matching space of features. An overview of method is presented in table 1-2, and more detailed descriptions are given below.

- Collins et al in (Collins, et al., 2001) develop a system consisting of multiple calibrated cameras and a site model. They track the objects from one camera to another using the object location with respect to a 3D site model followed by object matching using the color of the object, its class and the geolocation of the object.
- Huang and Russell (Huang, et al., 1997)present a probabilistic approach to vehicle tracking using two cameras on a motorway. The similarity between two objects is calculated using the probabilities. They use only the transition time (the time to move from one camera to another) is modeled as a Gaussian distribution which learned offline then these probabilities are modified during the tracking. They use the average color of whole object to model the object.
- Kettnaker and Zabih (Kettnaker, et al., 1999) develop a system to track the objects using a Bayesian formulation to compute likely paths of multiple people seen occasionally from cameras. They assume that objects moving through the monitored environment are likely to pass by several cameras, and that their movement is constrained to follow certain paths. They use the position, speed and time transition in their formulation. The topology of these allowable paths as input is given manually, together with information about transition probabilities and transition times. The appearances of objects are represented using histograms.
- Markris et al propose in (Makris, et al., 2004)a method to automatically create tempo topographical model of the network and the entry/exit zones of a network of non-calibrated cameras with overlapping as well as non-overlapping FOVs. This model allows associating the object which leaves a camera view field from a particular point or region to be matched only within the appropriate associated region of the adjacent camera. They used the appearance model to perform the correspondence inter-cameras.
- Rahimi et al (Rahimi, et al., 2004) propose a method to reconstruct the full path of an object when the object is observed by the cameras, and compute simultaneously the calibration parameters of the cameras. This approach is based on a probabilistic model of the location and velocity of the object. They have modeled the dynamics of moving object as a Markovian process. Then for each camera they map the image coordinate of the object to a local coordinate system lying on the ground-plane, and where all the data on the trajectory of the objects are available.
- Javed et al (Javed, et al., 2003) use a Bayesian formulation of the problem of reconstructing the paths of objects as in (Kettnaker, et al., 1999) but they used in their model the positions, speeds and transition time of the detected objects in cameras, and system is able to know the paths of movement automatically after a learning phase.

- Gilbert and Bowden (Gilbert, et al., 2006)use an incremental learning method, to model both the color variations and distributions probability of spatio-temporal links between cameras simultaneously to determine the main network of entry and exit areas in each camera probabilistically. They learn the network using the repetition of the trajectories of object across camera and then they use these temporal links inter camera can be used to link camera regions together and producing a probabilistic distribution of an objects movement across cameras.
- Van de Camp et al (de, et al., 2009) assume that the topology of the camera network is known. Then they model the whole camera network as a Bayes' Net. They create one HMM for each person they would like to track. Each HMM will only be used to determine the corresponding person's location. The topology of each HMM is identical and resembles the real camera network as it consists of states that represent cameras and states that represent paths' between cameras. The alphabet consists of all possible observations that can be made. Since the states of the HMM represent all possible locations for a person, either the person is at a camera or on a path from one camera to another.

Most research dealing with multi-tracking cameras with disjoint fields of view focuses on the maintenance of a coherent identity between multiple targets when they emerge from a field of view and enter another. This is called the data association problem (Rahimi, et al., 2004). These techniques provide a mechanism for establishing correspondences across disjoint views of fields, according to information from both space-time and the appearance of objects.

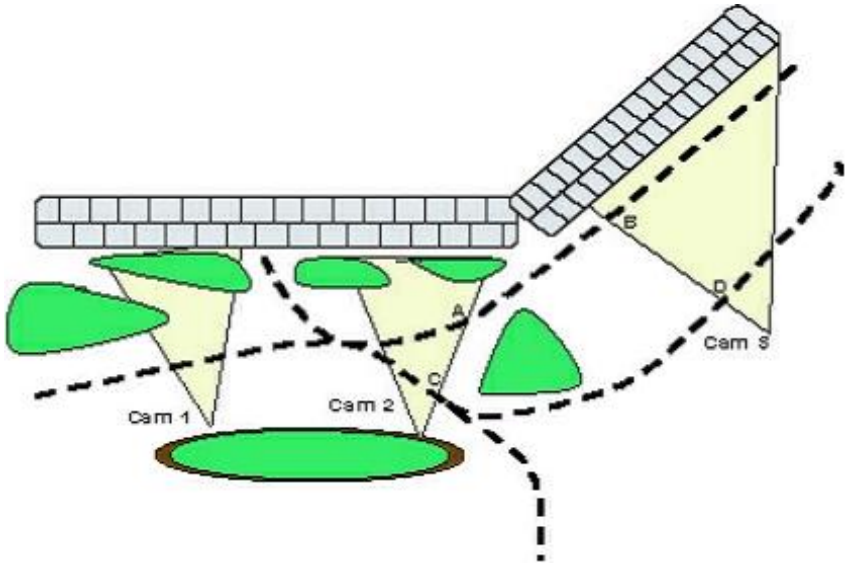


Figure 1-18: multi-camera tracking with disjoint fields of view. The dashed lines represent the trajectories (figure extracted from (Javed, et al., 2003)**)**

Table 1-2: Overview of Methods for tracking with non-overlapping multi-cameras

Reference Paper	Color information	Features	Association method	Calibration
(Huang, et al., 1997)	Average color	-	Probabilistic approach	-
(Kettnaker, et al., 1999)	Histogram of color	-	Bayesian formulation	-
(Collins, et al., 2000)	color	centroid of the silhouette	3d model	Yes
(Javed, et al., 2003)	Color	-	Epipolar geometry	No
(Makris, et al., 2004)	-	Patches elliptical	3D projection	No
(Rahimi, et al., 2004)	-	-	Probabilistic approach	Yes
(Gilbert, et al., 2006)	-	-	Topology of cameras	No
(de, et al., 2009)	Color	High features like glass , gender	HHM	-

1.3 Conclusion

In this chapter we have presented an overview of intelligent distributed video surveillance systems (IDVSS). It consists in an architecture based on cameras and computing servers and visualization terminals. The surveillance application is adapted to the monitored scene domain. Then, we described a basic intelligent surveillance architecture which contains: motion detection, tracking, behavior understanding, and personal identification.

These algorithms discussed above can be realized by single camera-based visual surveillance systems. Multiple camera-based visual surveillance systems can be extremely helpful because the surveillance area is expanded and multiple view information can overcome occlusion. We discussed the two approaches of multi-camera tracking depending on the type of overlapping. Tracking with a single camera easily generates ambiguity due to occlusion or depth. This ambiguity may be eliminated from another view.

However, visual surveillance using multi cameras also brings problems such as camera installation (how to cover the entire scene with the minimum number of cameras), camera calibration, and automated camera switching and data fusion, object matching. This last function heavily depends on type of feature used for matching. In the next chapter, we will present the various types of visual features that are commonly used for such object recognition or identification. The representation of object and method of association of features between two cameras is called the re-identification of object, and will be addressed in chapter 4.

2 Visual features

Motivation

The purpose of this chapter is to present concepts and different features commonly used for object recognition. We shall detail in particular those that we have studied and implemented in our work. First, we describe the global features, their principles and their issues. In the second part, we present local features defined by interest points that we have studied and used in this thesis.

Introduction

In computer vision, a feature is defined as a notable property of an object which specifies its overall appearance or its parts. It is a quantitative representation of the distinctive characteristics of an object

Features are a fundamental theme in many fields of computer vision, including detection, tracking, and recognition. Object features give a suitable representation of the real data of an object. These features have to reduce the redundant information of an object as much as possible and simplify the comparison of objects. The most important property of these features is to extract the most distinctive information of an object. This means that the feature should have several desirable properties such as robustness to changes in illumination and affine distortion as well as invariance to rotation and scale. Features can be described by different primitives such as the shape of the object, its contour, its texture and color. Many features can be used for representing an object.

These features differ from each other based on the primitives used. The extraction of these primitives is done by defining the way in which these feature describe the object. The feature is either **global**: we have one vector representing the whole object, or **local**: the feature describes small regions “blobs” or local positions “corners” of the object, in which case we have more than one vector to model the object.

In the following, we will first describe the different types of primitives used to specify the overall appearance of the object then we will elaborate the methods used to extract the local features of the object.

2.1 Feature types

We classify in the paragraph the global features depending on the primitives which are used to describe the objects such as its shape, contour, texture, color, region, interest points, etc. This section describes the methods used to extract the features and their advantages / inconveniences. The applications are elaborated in the related work of the third and fourth chapters.

2.1.1 Shape features

The use of moments is widespread in pattern recognition domain. Since their introduction by Hu (Hu, 1962) in 1961, which proposed a set of moments invariant to translation, scaling and rotation

using the theory of algebraic invariant, Fulsser and Suk (Flusser, et al., 1993) extended it to the affine moments invariant(AMI) which is invariant to affine transform, and Van Gool (Gool, et al., 1996) suggest a set that is additionally invariant to photometric condition. These moment based are more sensitive to noise than the moment which based on an orthogonal basis like Zernike moments (Khotanzad, et al., 1990) which are invariant to rotation and scale. Wang et al (Wang, et al., 1998) extended these moments to be invariant to illumination, with good experimental, but the method involves a high computational complexity. The work (Adam, et al., 2001)showed that the Fourier-Mellin transform gives better results than other signatures generally used in the literature for character recognition with multi-oriented multi-scale rotations on their images up to 180^0 and robust against noise.

2.1.2 Contour features

The second approach typically referred to is the Fourier descriptor (Rui, et al., 1996) . It involves a characterization of the contours of the shape. The curvature scale space descriptors (CSSDs) (Abbasi, et al., 1999) are also widely used as shape descriptors which detect the curvature points of the contours at different scales using a Gaussian kernel to convolve successive silhouettes. The experimental results in (Zhang, et al., 2003) show that the Fourier descriptors are more robust to noise than CSSDs. The main drawback is the need to obtain an object with a clearly segmented contour. It is difficult to obtain a full close contour of the object. Furthermore, the detection of all the edges contained in an image can be disturbed by internal or external contours of the object.

2.1.3 Texture features

It is difficult to give a mathematical definition of the texture. Generally, “a region in an image has a constant texture if a set of local statistics or other local properties of the picture function are constant, slowly varying, or approximately periodic” (Sklansky, 1978). Several methods have been used to extract the feature based on the texture. Scheunders et al in (Scheunders, et al., 1997) propose using wavelets to analyze the texture and extract features invariant to rotation. Khotanzad et al (Khotanzad, et al., 1990) use a bank of orientation selective Gabor filters to obtain texture filters. Newsam et al in (Newsam, et al., 2004) show that the performance of features based on Gabor filters give better results than features based on wavelets.

Figure 2-1: What kinds of features might one use to establish a set of correspondences between these images?

2.1.4 Color features

There is no ideal color space which is more distinctive than others. Several color spaces (RGB, HSV, Lab, Luv…) have been used in the re-identification or retrieval. The RGB (red, green, blue) color space is often used, although this space is sensitive to changes of illumination. The hue component of the HSV (Hue, Saturation, Value) space or the LUV space provide some invariance. However, Garcia et al in (Garcia et al., 2006) prove that the image from the camera is not completely invariant to changes in color of the light it receives. Indeed, the hue component changes with the shadows, and (Finlayson, et al., 2005) avoid the change of illumination by using histogram equalization. (Mindru, et al., 1998) propose color based moments which are invariant to illumination using the RGB space and invariant to affine transformation at the same time.

After an introduction to the main primitives used for recognition, we will now elaborate the methods for structuring the information of these features. As mentioned in the preceding paragraph, there are several types of primitives for recognition: the pixel, the pixel block, the region, contour, etc. Two approaches are possible to structure these feature: the global and the local approach. The first is the accumulation of information of any object to form a so-called global feature. The simplest example of this type of feature is the histogram.

2.2 Representation of global features

The structure most widely used to model the features is the histogram, which represents the probability density associated to visual primitives of an object, seen as random variables. The main advantage of histogram is its invariance to rotation and its slight robustness against occlusion. If the histogram is normalized, it is invariant to changes in scale. In addition, the histogram is simple to calculate and use. The histogram is not only used to describe the color of the object but could also be used to describe another primitive like the EOH descriptor (Edge Orientation Histogram) introduced in (Levi, et al., 2004). Levi et al propose to use EOH as a feature to detect faces. A further feature, the HOG (Dalal, et al., 2005), will be explained in the fourth chapter.

The representation of features as a histogram suffers from two main problems: first when a feature is described as a histogram, the spatial information, which is very important in many applications, is lost. This can be improved by splitting the object into a number of sub-regions and computing a histogram for each sub-region. Further modifications have been suggested to avoid this loss such as Huang et al who propose a new image feature in (Huang, et al., 1999) called the color correlogram. This feature is expressed as the correlation of color depending on the distance. The second problem of using histograms is its sensitivity to quantization. The number of bins used to describe the feature effect the discriminative of this feature.

Another example of global representation is the 2D appearance of the object is “the template matching” which exploits the spatial information and appearances at the same time. Its disadvantage, however, is that additional pre-processing steps are needed to make the representation invariant to position, rotation, scale, and illumination. Because all pixels are treated independently, the objects need to be correctly aligned and the pixel values normalized.

2.3 Local Features

The main problems of global features are essentially are that it cannot handle occlusions of the object, and that the influence of external condition such as pose, view point, lighting is difficult to model ,these difficulty makes global features are more sensitive to these conditions. To solve these problems, in recent years the objects are described by a set of local features (Harris, et al., 1988); (Lowe, 1999)) (Schmid, et al., 1997).

2.3.1 The principle of interest points

We now focus on interest points. In 1977, H. Moravec (Morevec, 1977)introduced the concept of interest points. According to him, certain points in an image may have characteristics more significant than others and therefore be of greater interest. Then P.R. Beaudet (Beaudet, 1978)formalized the concept of corners in an image. He was the first to propose a detector. First, the "interest points" are chosen at distinctive places in the image such as corners, blobs and T-junctions. The important quality of an interest point detector is its reproducibility, that is to say, it must find the same interest points under different conditions. Then the neighborhood of every interest point is represented by a vector of signatures. This descriptor should be distinctive and at the same time, robust to noise, error detection and geometric distortions and photometric.

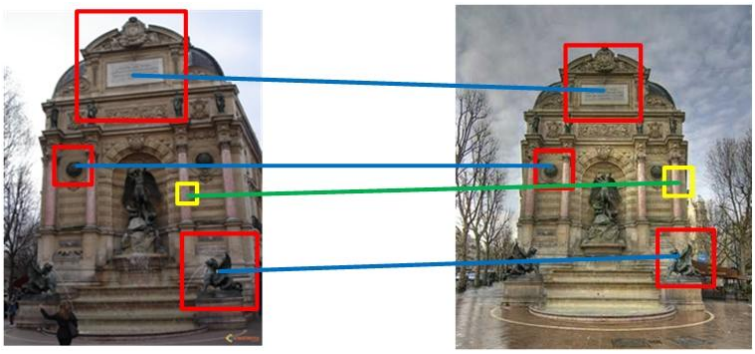


Figure 2-2: Notice how some patches can be localized or matched with higher accuracy than others

Once we calculate the vectors of descriptors. We use the distance between these vector to associate the images, such as Mahalanobis or Euclidean distance. The computation time is affected by the size of the descriptor directly. Hence, a small size is desirable. A wide variety of point detectors and interest point descriptors has already been proposed in the literature. Furthermore, a detailed comparison and evaluation was performed in (Mikolajczyk, et al., 2004). The detectors most commonly used are:

- Harris detector.
- Laplace of Gaussian detector.
- Difference of Gaussian Points (DoG) detector.
- Harris/Hessian Laplace detector.
- Entropy Based Salient Region detector (EBSR)
- Maximally Stable Extremal Region detector.
- Intensity Based Regions and Edge Based Regions

- Affine Harris/Hessian detector.
- SURF detector.

2.3.2 Interest points detector

The detection of interest points is a necessary step for descriptions of local feature of an object. It locates the most expressive points and regions of the object. It is generally able to reproduce the performance levels similar to those of human observers. The objective of a detector is not only to find the locations of the interest points, but also to provide the robustness of the local feature, the photometric and geometric deviations. Most detectors can be classified into two categories:

I. Corner detectors.
II. Region detector.

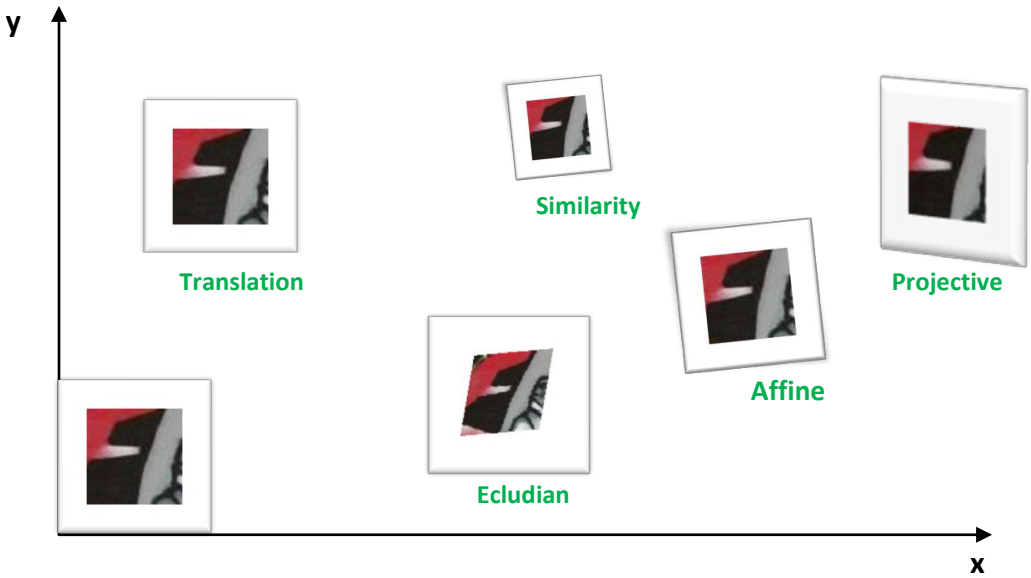


Figure 2-3: Basic set of 2D planar transformations

2.3.2.1 Definitions

An image is defined as a function $I(\overrightarrow{P})$ where the domain of I is the set of all positions $\vec{P} = (x, y)^t$ of image pixels. There are I_i derivatives of I with respect to i ($i \in [x, y]$). A Gaussian kernel with a local scale parameter σ is defined:

$$G(\vec{P}, \sigma) = \frac{1}{2\pi\sigma} \exp\left(-\frac{\vec{P}^t \vec{P}}{2\sigma}\right)$$

We introduce the following definitions we use for the detectors in this chapter.

The scale representation

For a given image $I(\vec{P})$ the scale representation $L(\overrightarrow{P;\sigma})$is a set of images of a scene represented at several levels of resolution, obtained by the following equation:

$$L(\vec{P},\sigma) = \int I(\vec{p}-\vec{q})G(\vec{q})\overrightarrow{dq} = G(\vec{p},\sigma) * I(\vec{p}) \qquad (Eq.2.1)$$

The derivatives of the image are computed by convolution with Gaussian derivatives:

$$L_x(\vec{P},\sigma) = \frac{\partial G}{\partial x}(\vec{P},\sigma) * I(\vec{P}) \qquad (Eq.2.2)$$

$$L_y(\vec{P},\sigma) = \frac{\partial G}{\partial y}(\vec{P},\sigma) * I(\vec{P}) \qquad (Eq.2.3)$$

Harris matrix

The Harris matrix is defined in the Harris detector and the eigenvalues of this matrix determined whether or not a point is a corner. The Harris matrix of a point $\vec{P}$ in an image is defined by:

$$A = G(\vec{P},\sigma) * \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x \\ I_y \end{bmatrix}^t \qquad (Eq.2.4)$$

Hessian

For a given image $I(\vec{P})$, the Hessian matrix is the matrix of second partial derivatives:

$$H = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{yx} & L_{yy} \end{bmatrix} \qquad (Eq.2.5)$$

$with$:

$$L_{xx} = \frac{\partial^2 G}{\partial x^2}(\vec{P},\sigma) * I(\vec{P})$$

$$L_{yy} = \frac{\partial^2 G}{\partial y^2}(\vec{P},\sigma) * I(\vec{P})$$

$$L_{xy} = \frac{\partial^2 G}{\partial x \partial y}(\vec{P},\sigma) * I(\vec{P})$$

Laplacian

The laplacian operator is invariant to the rotation of a given image $I(\vec{P})$

$$\Delta I = \nabla^2 I = (\nabla.\nabla)I = L_{xx} + L_{yy} \qquad (Eq.2.6)$$

Harris matrix adapted to the scale

$$M(\vec{P},\sigma_D,\sigma_I) = \sigma_D^2 * G(\vec{P},\sigma_I) * \begin{bmatrix} L_x^2(\vec{P},\sigma_D) & L_x(\vec{P},\sigma_D)L_y(\vec{P},\sigma_D) \\ L_x(\vec{P},\sigma_D)L_y(\vec{P},\sigma_D) & L_y^2(\vec{P},\sigma_D) \end{bmatrix} \quad (Eq.2.7)$$

With σ_I , σ_D are the parameters of local scale

2.3.2.2 Corner detector

For two decades, many corner detectors (Fig. 2.4) have been developed. Schmid and Mohr (Schmid, et al., 1997) compare the performance of several of them. The most popular interest point detector is that of C. Harris and Stephens (Harris, et al., 1988), also called Plessey detector. It is considered robust, reliable and invariant to rotation. It is based on the work of H. Morevec (Morevec, 1977)who had the idea of using the auto-correlation to determine the best position of the window. Hence, every neighboring position contains less information. This means that if one moves from the center of the window, it should be easy to distinguish the current position from the previous one. There are three cases that arise:

I. If the window is in a homogeneous area, the auto-correlation will give a low response in all directions.
II. If the response is strong in a predominant direction, the location cannot be distinguished in other directions. Hence it is an edge.
III. If the response is strong in all directions, we are in a zone which can be characterized: textured patterns, corners and blobs

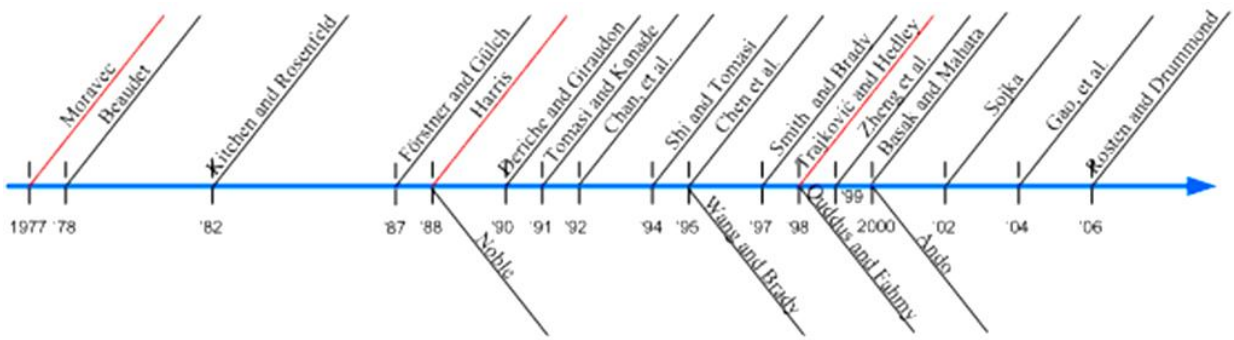


Figure 2-4: The chronology of the corner detectors

Harris and Stephens have shown that the calculation of the auto-correlation was reduced to the study of eigenvalues of the Harris matrix (Eq.2.4). Since the Eigenvalues of this matrix represent Harris principal curvatures of the auto-correlation as we can see in Figure 2-5, we have three cases:

a) If both Eigenvalues are small, the region is considered homogeneous.
b) If one Eigenvalues is clearly dominant, there is an edge.
c) Finally, if both Eigenvalues are high, there is no preferred direction. Hence there is an interest point.

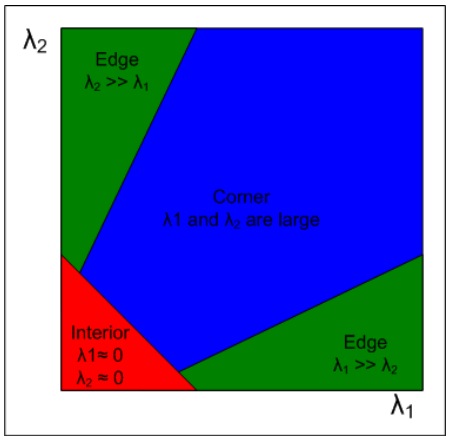


Figure 2-5: Classification of image points by the eigenvalues.

The problem is the evaluation of these Eigenvalues. To avoid their explicit calculation, C. Harris and Stephens proposed calculating a measure based on the determinant and the trace of the Harris matrix. Note that these values are invariant to isometric changes. Therefore the invariance to rotations of the image is preserved. It then evaluates the following measure:

$$R = \det(A) - K * trace(A)$$

If R> 0 then there is a point of interest, and A is a Hessian matrix K is generally between 0.04 and 0.06.

The Harris corner detector is widely used due to its high repeatability. However, it is computationally expansive and sensitive to noise, because it relies on the gradient information. Further, it provides an incorrect location on many types of junctions and is not scale invariant (scale corner detector is not well defined). In (Dufournaud, et al., 2000) Dufournaud et al proposed a multi-scale version of the Harris operator. Interest points are local maxima detected in the Harris detector applied on multiple scales.

The scale properties were studied by Lindeberg in (Lindeberg, 1998). The points are automatically indicated by the maxima in the scale space built from standard derivatives using Gaussian filters. Several functions have been proposed to construct the representations of images in different scales. These functions depend on the type of information to be extracted from an image (i.e., blobs, edges). The general representation is a set of images of a scene represented at several resolution levels.

2.3.2.3 Blob detector

In addition to the corners, the second type of "points", the blobs, is the most intuitive. We start with the methods based on derivatives.

Harris Laplace

The region detector of Harris-Laplace in (Mikolajczyk, et al., 2004) locates potential point "landmarks" with the Harris detector. It then selects the point with a characteristic scale, which is the extremum of the Laplacian on different scales. To describe the Harris-Laplace detector, we must define the measures and multi-scale Harris and Laplacian operators. The measure of the multi-scale Harris is given by:

$$m_l = \det\left(M(\vec{P}, \sigma_D, \sigma_I)\right) - \alpha.trace\left(M(\vec{P}, \sigma_D, \sigma_I)\right) \quad (Eq.2.8)$$

With $M(\vec{P}, \sigma_D, \sigma_I)$ is Harris matrix adapted to the scale, and α is a predefined scalar.

The operator of Gaussian Laplacian is given by:

$$|LoG(x, \sigma_n)| = \sigma_n^2 \left|L_{xx}(x, \sigma_n) + L_{yy}(x, \sigma_n)\right| \quad (Eq.2.9)$$

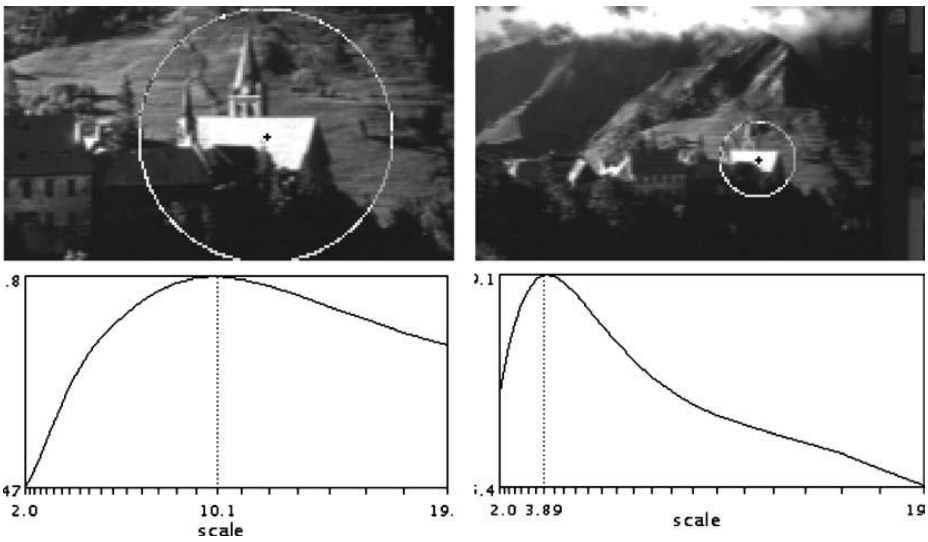


Figure 2-6: Set of responses for the function of the Laplacian (trace of scale) (Mikolajczyk, et al., 2004).

Given the two previous definitions, the Harris-Laplace detector contains two steps:

1. We obtain a set of points by detecting at each level of the pyramid of maxima in the surface of the image using $(Eq.2.8)$.

$$\vec{X} = \arg_{\vec{P}} \max m_l(\vec{P}, \sigma_D, \sigma_I), \sigma_D = \epsilon\sigma_I$$

2. A characteristic scale is selected for each detected point $(Eq.2.9)$.

$$\sigma_D = arg_{\sigma_D} \min \left| LoG\left(\vec{X}, \sigma_D\right) \right|$$

Harris affine and Hessian affine

The Hessian-Affine detector is similar to the Harris detector, except that they use the determinant of the Hessian instead of the Harris detector. The Eigenvalues of the matrix of second moment (see equation 2.5) are used to measure the affine transformed shape of the point neighborhood. This shape is estimated by three parameters. The simultaneous optimization of all three affine parameters: the point location, scale and shape, is too complex to be practically useful. Thus Mikolajczyk and Schmid (Mikolajczyk, et al., 2004) propose the following iterative process to estimate the parameters:

1. Detect the interest points using the measurement of multi-scale Harris Eq. (2.8) or the Hessian based detector.
2. Select the distinct scale for each interest point in the second stage of the Harris-Laplace detector.
3. Determine the form associated with an interest point by the Eigenvalues and Eigenvectors of the second moment matrix A $(Eq. 2.4)$.
4. Normalize the form of a circle according to:

$$\vec{P}' = A^{1/2}\vec{P}$$

5. Repeat steps 2 to 4 while the Eigenvalues are not constant.

The main disadvantage of the affine adaptation algorithms is the increase in runtime due to their iterative nature, but as shown in (Mikolajczyk, et al., 2005) for example, the performance of those shape-adapted algorithms is excellent.

Maximally Stable External region detector

Maximally Stable External Region (MSER) proposed by (Matas, et al., 2004) is a region-based segmentation detector which computes the partitions of the image where local threshold stable over a large range of thresholds. Therefore, this technique only works on grayscale images. This operation can be efficiently implemented by first sorting all pixels by their gray value, and then incrementally adding pixels to each connected component as the threshold is changed, and monitoring the rate of area change of this component. The regions are defined as maximally stable when this rate is minimal (Nistér, et al., 2008).

Salient regions detector

Another method, which does not use the derivatives of the image, is the salient region detector proposed by Kadir and Brady (Kadir, et al., 2001). It is based on the probability density function (pdf) of the intensity values $P(I)$ calculated on the elliptical region Ω. The procedure is as follows:

1. For each pixel $\vec{P}$, calculate the entropy of the $pdf\ p\ (I)$ of an ellipse centered at $\vec{P}$, where s is the scale, θ is the direction and λ is the ratio between the two axes of the ellipse.

2. All entropy extrema at different scales is registered as a candidate salient regions and the entropy is given by:

$$H = -\sum_{I} P(I) Log\big(P(I)\big) \qquad (Eq. 2.10)$$

3. The salient regions are classified according to the amplitude of the derivative of the pdf for each extremum on s, which is defined by:

$$\mathcal{W} = \frac{s^2}{2s-1} \sum_{I} \left| \frac{\partial P(I, s, \lambda, \theta)}{\partial s} \right| \qquad (Eq. 2.11)$$

4. The saliency γ is then computed as

$$\gamma = \mathcal{W}.H \qquad (Eq. 2.12)$$

The candidate salient regions over the entire image are ranked by their saliency γ, and the top P ranked regions are retained. The main shortcoming of the algorithm is its long runtime.

2.3.3 Efficient implementations of IP detectors

The common step to most recent detectors implements the convolution process to calculate the first order derivatives for Harris detector or second derivatives for Hessian detector. In some cases these processes are iterative like the Affine-Hessian or Affine-Harris. This process needs a very high computation effort which is why these detectors are not suitable for real time applications. We will present the interest points detectors that have been developed to be computationally efficient.

2.3.3.1 Difference-of-Gaussians

Lowe (Lowe, 2004)found that the convolution of the image using different Gaussian filters constitute the principal calculation time because the others methods use two Gaussian filters, one for the derivative calculation and another for image scaling. Therefore, he uses the same filter for both and integrates the two steps. Hence, the calculation time is reduced. The difference of Gaussian "DoG" was proposed by Crowley and Parker (Crowley, et al., 1984). It detects local extrema in the scale space using the convolution of the image with the DoG function:

$$\begin{aligned} D(\vec{P}, \sigma) &= \Big(G(\vec{P}, k\sigma) - G(\vec{P}, \sigma)\Big) * I(\vec{P}) \\ &= L(\vec{P}, K\sigma) - L(\vec{P}, \sigma) \end{aligned} \qquad (Eq. 2.13)$$

Lowe uses this operator to handle the change of scale in the SIFT detector by comparing the Laplacians $L_{xx} + L_{yy}$ which are the derivatives of the image relative to the scale. The simple difference between adjacent points in a given direction approximates the derivative in this direction

$$\sigma\nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(\vec{P}, k\sigma) - G(\vec{P}, \sigma)}{k\sigma - \sigma} \qquad (Eq. 2.14)$$

As result

$$G(\vec{P}, k\sigma) - G(\vec{P}, \sigma) \approx (k-1)\sigma^2\nabla^2 G \qquad (Eq. 2.15)$$

The image is convolved with multiple Gaussian kernels. The prior selection of the interest points and their scale is performed by detecting the local extrema of the difference of Gaussians $G(\vec{P}, \sigma)$ as shown in figure 2-7 Then the extrema are found in small neighborhoods around the position and the scale (typically 3 x 3 x 3). An interpolation step is used to improve the location of the interest points in space and scale. Finally, an analysis of the Eigenvalues of the Hessian 2 x 2 matrix eliminates interest points in insufficient areas or with contrasting edges with a bend that is too low. The next step is to assign a direction to each point. This orientation corresponds to the orientation of the majority of spatial gradients of intensity calculated in a neighborhood of the interest points around the previously determined ones. One interest point may be included in several directions.

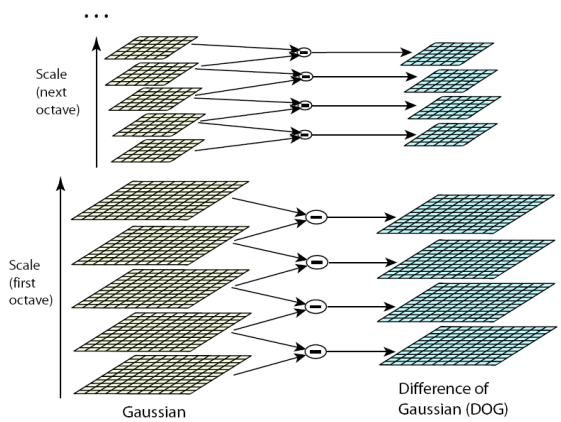


Figure 2-7: Multi scale Approach extracted from (Lowe, 2004)

The interest points detected by DoG are shown in figure 2-8.The main property of SIFT is that it is invariant to scale and that several frames per second can be treated with this method. Despite the success of the SIFT method, it can be insufficient for not being fast enough.

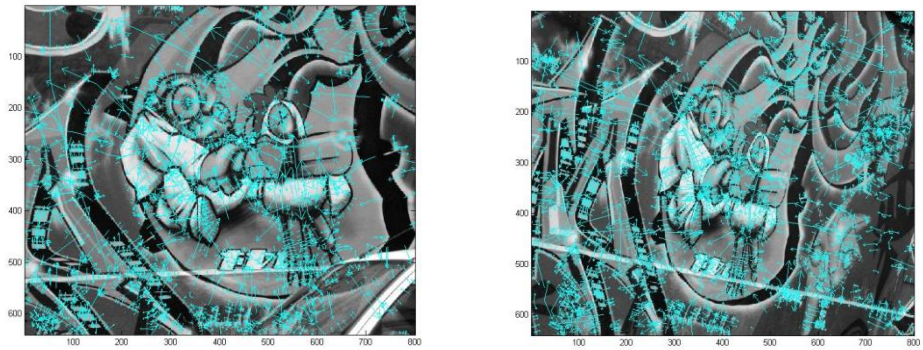


Figure 2-8: The local signatures detected with the DoG-detector

2.3.3.2 SURF: Speeded Up Robust Features

Bay et al (Bay, et al., 2006)use the integral image to obtain a fast implementation of a box-type filter convolution. Then they use this filter to approximate the Hessian Matrix. The integral image is used in the implementation of the widely known face detector by Viola and Jones (Viola, et al., 2001). The integral $I_w(x,y)$ at location (x,y) contains the sum of pixels restricted by the origin and the location(x,y).

$$I_w(x,y) = \sum_{\substack{x'<x \\ y'<y}} I(x',y') \;\; where \;\; I(x,y) the\ value\ of\ pixel \quad (Eq.2.16)$$

We can obtain the integral image in one scan using the recurrence relation:

$$\begin{aligned} s(x,y) &= s(s,y-1) + I(x,y) \\ I_w(x,y) &= I_{w(x-1y)} + s(x,y) \end{aligned} \qquad (Eq.2.17)$$

The main advantage of the integral image representation is that the calculation time of the box filter is independent of its size. It needs four addition operations, whatever the size, as we can see in figure 2-9.

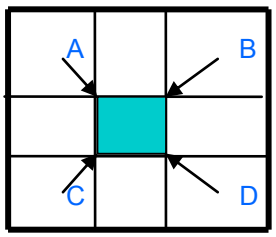


Figure 2-9: Just four addition operations to calculate the sum of intensities on any rectangular region

SURF (speed up robust feature): Bay et al (Bay, et al., 2006)propose using the determinant of the Hessian matrix to detect the interest points for selecting the locations and scales. They extended the approximation of Lowe using the box filters instead of Gaussian filters to approximate the Hessian matrix as shown in figure 2-10. They avoid using the convolution filter during the transition from one

scale to another. Despite these large approximations, the performance of the detection points is comparable to the results obtained with the Gaussian filter.

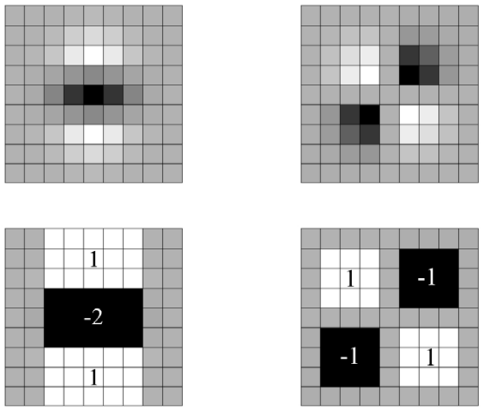


Figure 2-10: Left to right: The Gaussian second order partial derivative in the y direction and xy direction respectively extracted from (Bay, et al., 2006).

The estimated determinant of the approximated Hessian represents the response of the blob in the image at the position (x, y) and is given by the relation:

$$\det(H_{approx}) = D_{xx}\, D_{yy} + (0.6D_{xy})^2 \geq threshold \quad (Eq. 2.18)$$

where $D_{xx}, D_{yy}\ and\ D_{xy}$ represent the results of the box-type filter, and $threshold$ is a constant selected empirically.

These responses are stored and the extrema are found in small neighborhoods around the position and scale (typically 3 x 3 x 3). An interpolation step is adapted to improve the locations the interest points in space and scale like the implementation of SIFT. Bauer et al (Bauer, et al., 2007)have shown that the number of points detected by SURF is fewer than the points produced by the SIFT detector over real images. The performance of SURF is hereby slightly lower than that of SIFT. This property is important for many applications which do not need many points which take time in matching. As a conclusion, SURF compromises the performance and the calculation time which make it popular. The results of the SURF detector are shown in figure 2-11.

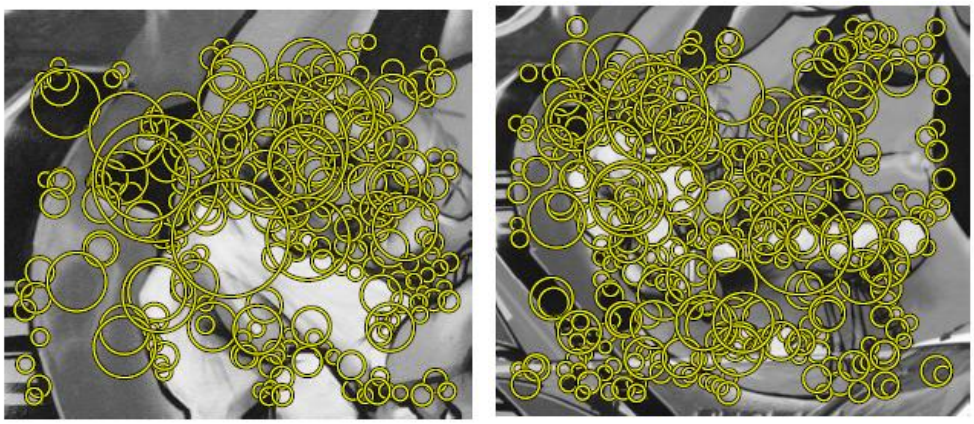


Figure 2-11: the interest points detected with SURF detector extracted from (Tuytelaars, et al., 2008)

2.3.3.3 Camellia Keypoints

The Camellia Keypoints detector functions available in the Camellia image processing library[4] developed in our lab proposed by Bruno STEUX. Their detection and characterization functions implement a very fast variant of SURF. SURF uses 4 octaves in the phase of detection, while Camellia keypoints use only 8 image pyramids. Another main difference between Camellia keypoints and SURF is that the Camellia implementation uses integer-only computations – even for the scale interpolation – which makes it even faster than SURF**.** Figure 2-12 shows the interest points detected using the Camellia keypoint detector.

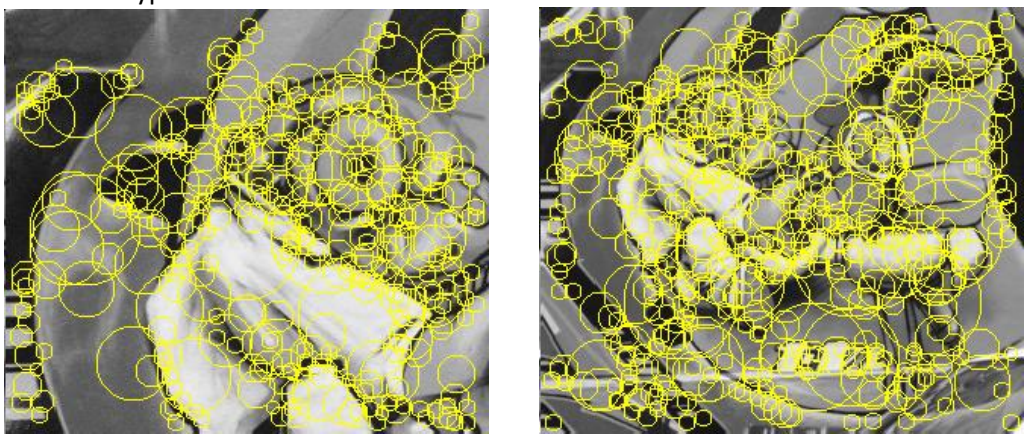


Figure 2-12: the interest points detected using the detector by Keypoint Camellia.

We use Camellia keypoints instead of SURF in several of our applications like the detection of pedestrian (Bdiri, et al., 2009).

2.3.4 The interest points descriptors

Once we have detected the interest points, we can determine the location of points, the scale of the points, and in some cases the orientation, of these points. We can benefit from the repeatability property of the detector to describe the zone limited by the scale of the points to compare and match the points. The descriptor is a one dimensional or multi-dimensional vector containing the

[4] http://camellia.sourceforge.net

signatures which should be invariant to brightness and robust against noise, compression artifacts, and scale change. Further, these signatures should be robust to the main drawback of the blob detector which is the inaccuracy of the location. A variety of descriptors of signatures has been proposed, such as the gradient location-orientation histograms (GLOH) (Mikolajczyk, et al., 2005), Spin Images (Mindru, et al., 2004), differential-invariants (Schmid, et al., 1997), Gabor filters, SIFT (Lowe, 1999), SURF (Bay, et al., 2006). In (Mikolajczyk, et al., 2005), Mikolajczyk and Schmid proposed to divide the descriptor into the following three categories:

- Filter based descriptors
- Distribution based descriptors,
- other methods.

The following descriptors will be discussed more detailed:

- Differential-invariants.
- SIFT descriptor.
- SIFT PCA descriptor.
- CSIFT Descriptor.
- SURF descriptor

2.3.4.1 Differential-invariants

Schmid et al. (Schmid, et al., 1997) propose the characterization of local contours by calculating differential invariants after the detection of the interest points by Harris detector. Each region associated is described by a combination of differential operators in order to obtain rotational invariance. The descriptor v is based on the combined set of local jet calculated up to the third order (Florack, et al., 1996).

$$v[0..8] = \begin{bmatrix} L \\ L_i L_j \\ L_i L_{ij} L_j \\ L_{ii} \\ L_{ij} L_{ji} \\ \varepsilon_{ij}(L_{jkl} L_j L_k L_l - L_{jkk} L_i L_l L_l) \\ L_{iij} L_i L_k L_k - L_{ijl} L_i L_j L_k \\ -\varepsilon_{ij} L_{jkk} L_i L_k L_l \\ L_{ijk} L_i L_j L_k \end{bmatrix} \quad (Eq. 2.19)$$

With

$$L_i = \sum_i L_i = L_x + L_y$$
$$L_{ij} = \sum_i \sum_j L_{ij} = L_{xx} + L_{yy} + L_{xy} \quad (Eq. 2.20)$$

$$\epsilon_{12} = -\epsilon_{21} = 1$$

$$\epsilon_{11} = \epsilon_{22} = 0$$

The indices $i;\ j;\ k$ are the corresponding derivatives of the image in the two directions in the image. The main drawback of this descriptor is its dependence on the degree of Gaussian derivative of the image for obtaining distinct information which makes it sensitive to noise.

2.3.4.2 SIFT descriptor

The principle of the SIFT descriptor is to calculate the orientation of the gradient at each point of the image patch to describe. This region is then divided into (for example) four or sixteen sub-regions. An 8 quantization level histogram summarizes the directions of the gradient intensities within the sub-regions as shown in figure 2.13. The descriptor SIFT is a vector of 4 x 4 x 8 = 128 values.

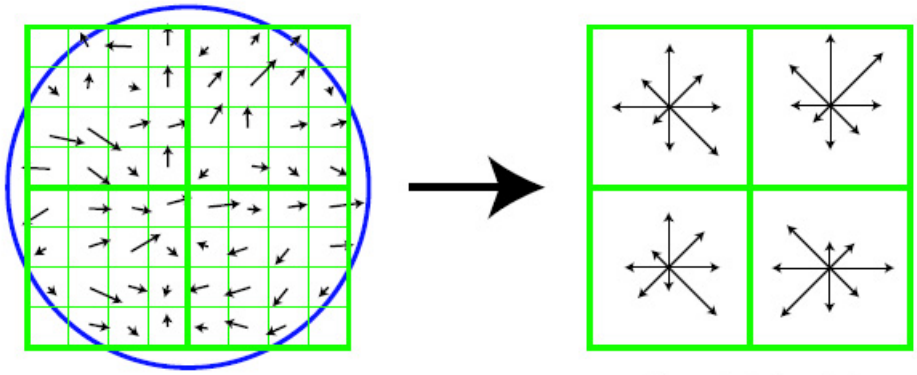


Figure 2-13: Illustration of the SIFT descriptor calculation partially taken from (Lowe, 2004)

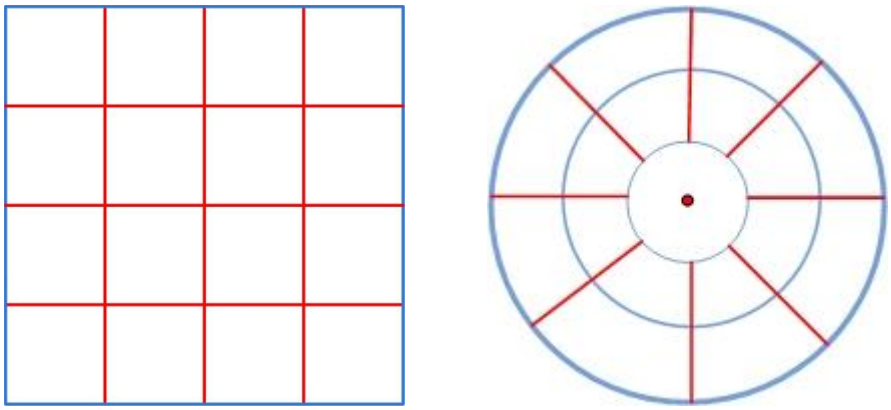
The invariant scale property of this descriptor is achieved by the output of the detection process and the rotational property concerning the orientation assignment calculated in the recent process. This descriptor is not fully invariant to the affine transform (Lowe, 2004), but the method described allows for small change in the descriptor when the regions evolve by a small affine transform and make the descriptor robust against changes in 3D viewpoints.

Several improvements are proposed to modify the SIFT descriptor like PCA SIFT (Yan, et al., 2004) and GLOH (Mikolajczyk, et al., 2005). Ke and Sukthankar (Yan, et al., 2004) applied PCA on the gradient image instead of the calculation of gradient orientation. They computed the Eigenspace off-line, then they projected the gradient of the patch which is determined by SIFT into the Eigenspace. They empirically determined the dimensionality of the feature. It was shown that the 20 dimensional feature vectors for each patch are sufficient to obtain a descriptor which is faster for matching but it is less distinctive than SIFT (Mikolajczyk, et al., 2005).

Another improvement is GLOH (Gradient location-orientation histograms) which is proposed by Mikolajczyk and Schmid in (Mikolajczyk, et al., 2005). The main difference between GLOH and SIFT is using the log polar grid instead of the Cartesian grid as shown in 2-14. They divide the patch into 17

cells and the gradient orientation of each cell is quantized in 16 bins histograms, which produce 272 bin histograms. Then PCA reduces the size of the descriptor to 128. The disadvantage of this descriptor is that it is more expensive in computation yet it outperforms SIFT.

Abdel-Hakim et al in (Abdel-Hakim, et al., 2006), propose a colored local invariant descriptor which combines the color and the gradient orientation. They use the invariance model developed by Geusebroek et (Geusebroek, et al., 2001) to obtain the color invariance but the scale and rotation are calculated using the SIFT structure.



(b) Cartesian grid (a) Log-polar grid

Figure 2-14: Cartesian & log-polar location grids. The log-polar grid shows nine location bins used in shape context (four in angular direction).

2.3.4.3 SURF descriptor

We describe here the SURF descriptor (Speedup Robust Features (Bay, et al., 2006)). The main motivation is to provide a local description which is inexpensive in computing time and has the scale and rotation invariance properties. Bay et al proposed the SURF descriptor based on the idea of SIFT. The rectangular regions around the interest point are cut into 4 × 4 and these sub-regions are described by the 4 Haar features defined by the vector v:

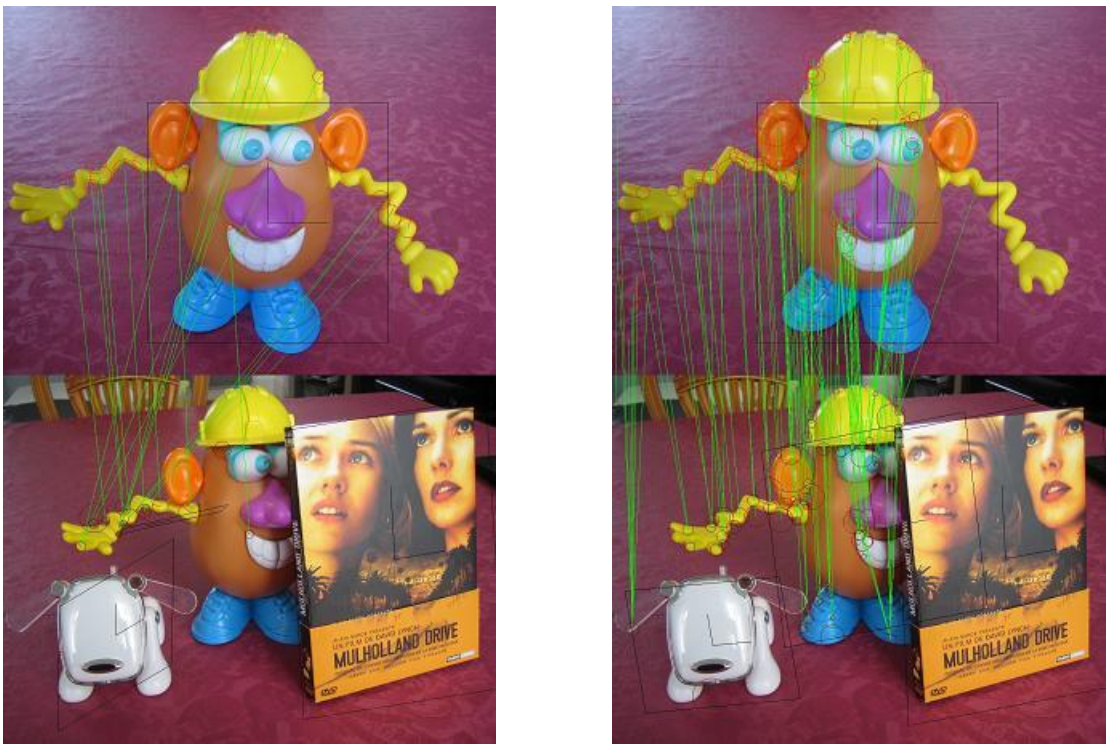
$$v = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|\right) \quad (Eq. 2.21)$$

d_x and d_y are the responses of the Haar wavelet analysis in the horizontal and vertical. The advantage to using the Haar feature is the possibility to use the integral image to calculate this feature which reduced the time of computing. This descriptor performs slightly best than SIFT descriptor (Bay, et al., 2006). The main drawback of this descriptor is that it is not invariant under brightness change.

The Camellia descriptor is a modified version of the SURF descriptor. The tri-linear interpolation is used to distribute the value of every single feature in the adjacent histogram bins. There is another version of Camellia descriptors which use the Yuv color space. The dimension of this descriptor is 128, the first 64 bins represent the descriptor of the gray component (Y) and the last 64 bins represent the u, v color components respectively.

2.3.5 Choice of descriptors and detector

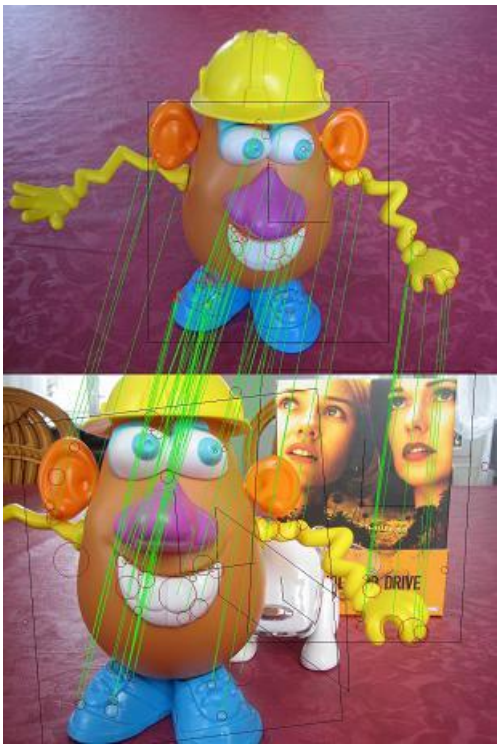
The choice of detector is based on the application and the context of the application. The corner detectors are based on the first derivative which allows them to be more accurately localized in the image than the detectors which depend on the second derivative, such as the blobs detector. Therefore, this accuracy makes the corner detectors more suitable for 3D reconstruction applications and camera calibration. For the object recognition, the robustness against the changes scale is necessary, which makes the blob detector more suitable (Tuytelaars, et al., 2008). Another factor which influences the choice is the number of extracted region. The number of interest points given by SIFT is bigger than the number given by SURF. The principle of SIFT is to give many points, of which the majority is eliminated in the matching process or by another constraint. However, this causes a problem of processing time when we compute the high dimensional 128 bin descriptor and establish the feature matches. On the other hand, SURF is a compromise between the number of points and the calculation efficiently. The performance of SURF is better than SIFT in two comparative studies (Bay, et al., 2006) and (Bauer, et al., 2007). In our re-identification application, we use the Camellia Keypoints (figure 2-15) which are similar to SURF as an interest-point detector and descriptor.



(a) SIFT (b) CamKeypoint

Figure 2-15: Comparisons between SIFT and CamKeypoints: note that the number of points matched between the interest points of two images extracted by SIFT is relatively small compared to the number of points extracted by CamKeypoints. We try to detect the same number in both cases by changing the threshold.

(a) SURF



(b) CamKeypoint

Figure 2-16: Comparisons between SURF and CamKeypoints we note that the number of points that match between the interest points which extracted by SURF is relatively small compared to the number of points extracted by the CamKeypoints. We try to detect the same number in both cases by changing the threshold.

In Figure 2-15 and 2-16, each circle indicates an interest point. The center of the circle is the location of the interest point. The radius of the circle corresponds to the magnitude of the gradient. Note that the number of matches between the interest points extracted by SIFT and SURF respectively are relatively small compared to the number of correct matches extracted by the Camellia keypoints. Second, we note that the interest points extracted on the object are often located at the edges. In table 2-1, we note that the Camellia Keypoints are calculated almost twice faster than SURF and 6 times faster than SIFT. The averages are calculated over 100 executions of the implementation of point extraction and the descriptor calculation.

	camKeypoints	SURF	SIFT
Computing Time (ms)	**270**	**469**	**1500**

Table 2-1: Computing the average time over 100 times for the implementation of extraction of points and then calculation of descriptors. The thresholds are adapted to detect the same points of interest to all methods. We note that the Camellia Keypoints are calculated almost 2 times faster than SURF and 6 times faster than SIFT

2.4 Conclusion

The goal of the feature extraction is to define an optimal object description. Many local and global approaches have been developed amongst which we have selected the local ones, which describe the object by a set of local features. The local approaches provide some robustness against partial occlusion while the global approaches are more sensitive to changes in illumination, pose, viewpoints, etc than the local features.

In chapter four we will compare the performance of SURF, SIFT and the Camellia keypoints on pedestrian re-identification. We will also compare the local and global features such as color and HOG histogram. Before presenting re-identification, we will show in next chapter that interest points can be used not only for object recognition, but also for the pedestrian detection (or other moving objects) from background.

3 Pedestrian Detection with keypoints

Motivation

The following presents a keypoints-based method for modeling a scene and detecting new objects in it. The scene is modeled using a hybrid technique consisting of background modeling with keypoints, and filtering these points using an Adaboost classifier. New objects are detected by feature-based detection and background substraction. The motivation for doing this with keypoints (instead of using classical pixel-based approaches) is the idea that future smart cameras could compute keypoints efficiently on integrated hardware and output keypoints as an alternative (more abstract) description of the scene.

Introduction

Detection of moving objects is usually the first step in every video surveillance system. Background modelling is not a trivial process. When using a fixed camera, the background modelling algorithm has to cope with the problem of dynamic changes in natural scenes such as illumination changes, weather changes, and repetitive motions that cause clutter (waving trees). A further difficulty arises when an object, which was initially classified as part of the background, moves or when an object moves during the background training. These challenges are overcome based on the possibility of the algorithm to update the model. The speed of these updates depends on the application.

The method most widely used for detecting moving objects is the background substraction, which works by subtracting the intensities of pixels or the color of these pixels in the current frame from the background model. The main steps in background substraction are the background modeling and the foreground detection. The background modeling step provides a pixel-wise statistical representation of the entire image and uses new frames to update a background model. Foreground detection then classifies pixels in the video frame that cannot be adequately explained by the background model and outputs them as a candidate foreground binary mask. On the other hand, our background modelling algorithm provides an interest-points-wise statistical representation. The foreground detection then uses the statistical representation to compare the interest points of the new frame. Our aim is to adapt this processing chain to the case where, instead of full video streams, only interest point descriptors from various smart cameras would be collected. We present here a method combining background modelling and substraction using interest points.

Figure 3.1 shows the flow chart for the moving object detection procedure. This procedure starts with the background modeling step. The next step is detecting the change in the image using the background model and the current frame from the video stream. The learning of the background model and the change detection are straight-forward. The foreground detection step is generally followed by a pixel-level post-processing stage to perform some noise removal filters. Once the result is enhanced, the foreground pixel-map is ready. In the next step, a connected components analysis is used to label the connected regions. The labeled regions are then filtered to remove small regions

caused by noise. A number of object features are then extracted, like the centroid of the object and its bounding box.

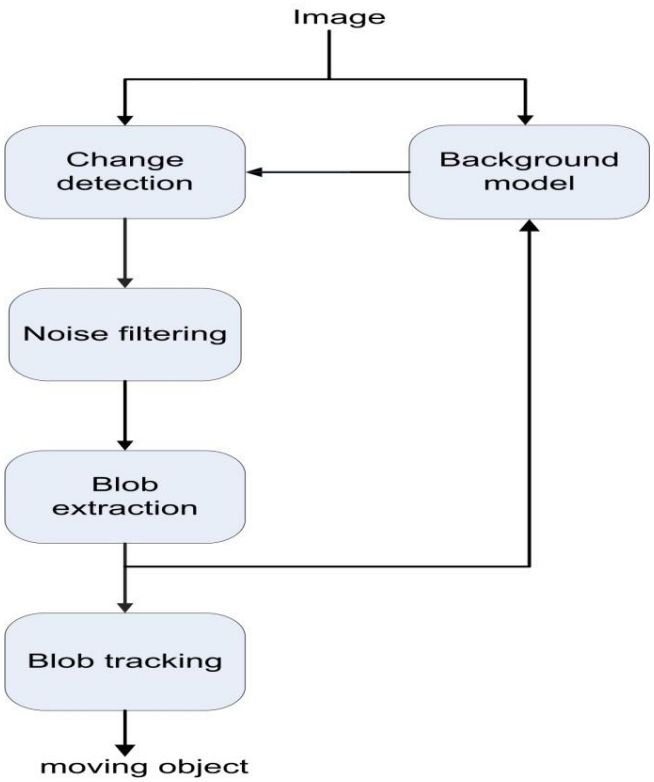


Figure 3-1: motion detection flow chart

3.1 Related works

To our knowledge there is no published work presenting adaptation of background modeling and substraction using only keypoints. We therefore present below the related works using other methods for the same objective. We will focus on the problem of background modeling, which can be considered as a special case of motion detection in video surveillance. We classify the background modeling methods depending on the representation of the background. We can distinguish three categories: background substraction methods, statistical methods and block-based methods. In the background substraction methods, the background is modeled as a single frame and the classification of the foreground is estimated by a pixel-wise thresholding of the absolute difference between the background image and the current frame. On the other hand, the statistical methods estimate the pixel variation in the background using a statistic model. They test if a pixel in the input image is compatible with this learned model. The spatial-based background model uses spatial invariant features to detect monotonic changes in a local area. We summarize these models in the following.

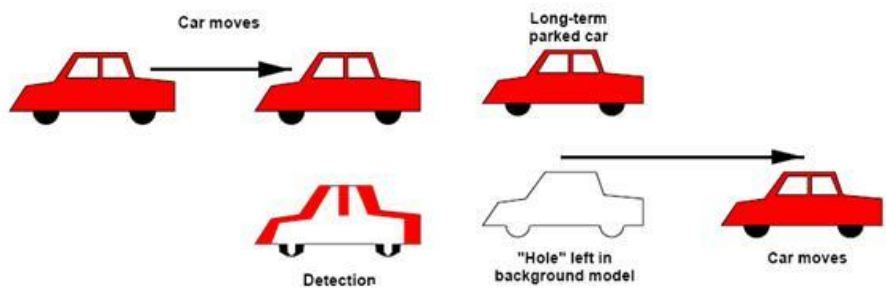
3.1.1 Background substraction

There are different approaches to the basic scheme of background substraction in terms of background and foreground classification. These methods detect objects by computing the difference between the current frame and the background. Let B^t be the current background image and I^t the current image. The difference image D^t is estimated by:

$$D^t(x,y) = |I^t(x,y) - B^t(x,y)| \quad (3.1)$$

The pixels are assumed to contain motion if the absolute difference $D^t(x,y)$ exceeds a predefined threshold τ. Motion pixels are labeled with 1 and non-active ones with 0. The resulting binary image usually contains a significant amount of noise. A morphological filtering step is applied to eliminate isolated pixels.

There are several approaches to update the background. The simplest method is the temporal difference method, which attempts to detect moving regions by exploiting the pixel-by-pixel difference in (two or three) consecutive frames in a video sequence. The temporal difference method produces good results in dynamic environments because it is highly adaptive, although it cannot extract all the pixels of moving objects. An example of inaccurate motion detection is shown in figure 3.2-b. The temporal differencing algorithm fails to extract all of the pixels of the monochrome region of the moving vehicle.



(a) (b)

Figure 3-2: problems with standard detection algorithms. (a) Substraction of the background images leaves "holes" when stationary objects move. (b) Temporal differencing fails to detect all moving pixels of the object on the left hand side since it is uniform color (Figures extracted from (Collins, et al., 2000)).

Another approach estimates the background model using the running average of the image (e.g., (Friedman and Russell 1997)):

$$B^{t+1}(x,y) = \alpha I(x,y) + (1-\alpha)B^t(x,y) \quad (3.2)$$

Where α is the learning rate. Unlike temporal differentiation, this approach performs well in extracting most of the pertinent pixels of moving regions even when they stop. However, they are usually sensitive to dynamic changes. For instance, when stationary objects uncover the background

(e.g. a parked car moves out of the parking lot) or sudden illumination changes occur. An example for inaccurate motion detection is shown in 3.2-b.

In order to overcome the shortcomings of both approaches, Collins et al. developed a hybrid method that combines three-frame differencing with an adaptive background substraction model for their VSAM project (Collins, et al., 2000).

3.1.2 Statistical Methods

The main drawback of the background substraction technique is its sensitivity to repeat motions that cause clutter (waving trees) because the background model is changed periodically. Therefore this method is not suitable for outdoor images. To overcome this sensitivity, the statistical methods use the pixels variations to construct the background model. The foreground pixels are identified by measuring the likelihood between the pixel and the scene model. Wren et al (Wren, et al., 1997) propose using a single Gaussian distribution to model the pixel intensity. However, a single-mode model cannot handle multiple backgrounds like waving trees. Thus, Friedman and Russel (Friedman and Russell 1997) applied a Mixture of Gaussians to estimate the background model. They use the EM-algorithm to update the Gaussian mixture, which makes this approach unsuitable for real time applications. This approach was extended by Stauffer and Grimson (Stauffer, et al., 1999). They propose to use a more efficient on-line approximation of k-means to update the Gaussian mixture model. This approach is reliable in scenes that contain noise, illumination changes and shadow which makes it is popular (Hu, et al., 2004). It should, however, be noted that modeling backgrounds with few Gaussian may be not robust to the fast variations. So this method may fail to provide a sensitive detection to moving objects. To overcome this problem, Elgammal et al (Elgammal, et al., 2000) propose a non-parametric technique to estimate the probability density function (pdf) at each pixel from many samples using a kernel density estimation technique. It is able to adapt very quickly to changes in the background and detect targets with high sensitivity/accuracy.

Another approach is proposed by Haritaoglu et al in (Haritaoglu, et al., 2000). In this method, the background scene is modeled by representing each pixel by three values: minimum intensity (m), maximum intensity (M), and the maximum intensity difference (D) between consecutive frames. Non parametric methods cannot be used when long-time periods are needed to sufficiently sample the background. To overcome this shortcoming, Kim et al (Kim, et al., 2005) suggest representing the background model by a codebook, where each pixel is represented by a number of code words. During run-time, each foreground pixel creates a new codeword. A codeword not having any pixels assigned to it throughout a certain number of frames is eliminated.

In (Rittscher, et al., 2000), Hidden Markov Models (HMMs) are used to model pixel intensity. In this method, the pixel intensity variations are represented as discrete states corresponding to scene models. The pixel states represent the foreground, background and shadows. Kalman filters have also been used for background modeling. In (Zhong, et al., 2003), Zhong et al the background is modeled using an Autoregressive Moving Average Model (ARMA). They use a Kalman filter to iteratively update the background model as well as to detect the regions of the foreground objects.

3.1.3 Spatial-based Background Models

A pixel-based background model is very robust when the pixel value constantly changes, for example due to changing weather conditions. However the main disadvantage of these methods is that the module is sensitive to noise e.g camera noise. Hence, recently several techniques were proposed that use block-based approaches instead of single pixel representation. Eng et al. (Eng, et al., 2003) divide a background model learnt over time into a number of non-overlapping blocks which are classified into, at most, three classes according to their homogeneity. The background block is represented by the means of these classes. Heikkila and Pietikainen (Heikkila, et al., 2006) suggest using local binary patterns (LBP) to describe the texture operator for a spatio-temporal block-based (overlapping blocks). This model makes the background model invariant to repetitious illumination changes. Oliver et al (Oliver, et al., 2000) propose using the Eigenspace to represent the background so that new objects are detected by comparing the input image with an image reconstructed via the Eigenspace. Pham and Smeulders (Pham, et al., 2006) proposed to estimate a background model by computing HOG descriptors on a grid of overlapping cells and blocks.

Javed et al. (Javed, et al., 2002) process images at three levels: the pixel, region, and frame levels. At the pixel level, they use a mixture of Gaussians to perform background substraction in the color domain. They also use the statistical model of gradients at the same time. The two models are separately used to classify each pixel as belonging to background or foreground. On the region level, foreground pixels obtained from the color based substraction are grouped into regions. Gradient based substraction is then used to make inferences about the validity of these regions. Shimada and Taniguchi in (Shimada, et al., 2009) propose to use hybrid type of background model consisting of two different kinds of background models. One is the Mixture of Gaussians which is robust to long-term illumination changes. The other is a spatial-temporal background model which is robust to short-term illumination changes. The spatial based background model allows detecting coarse motion depending on the size of the block used to divide the image. To avoid these shortcoming (Orten, et al., 2005) use a multi-scale approach to model and detect the objects.

The performance of any approach is not only measured by the accuracy of this algorithm. Hence, there are other factors which distinguish the algorithms, like the speed of implementation and the application of the algorithm. The classification and updating are two important elements in any motion detection method. So the overall accuracy of the background depends on the combination of this element with the background representation.

3.2 The proposed method

The goal of our method is to identify the interest points of a frame which belong to moving objects. The method extracts the interest points of N frames. Then the clustering of points is performed depending on the descriptors and the location of the points as illustrated in figure 3-3. By using the clustering result, it is possible to extract the moving interest points robustly.

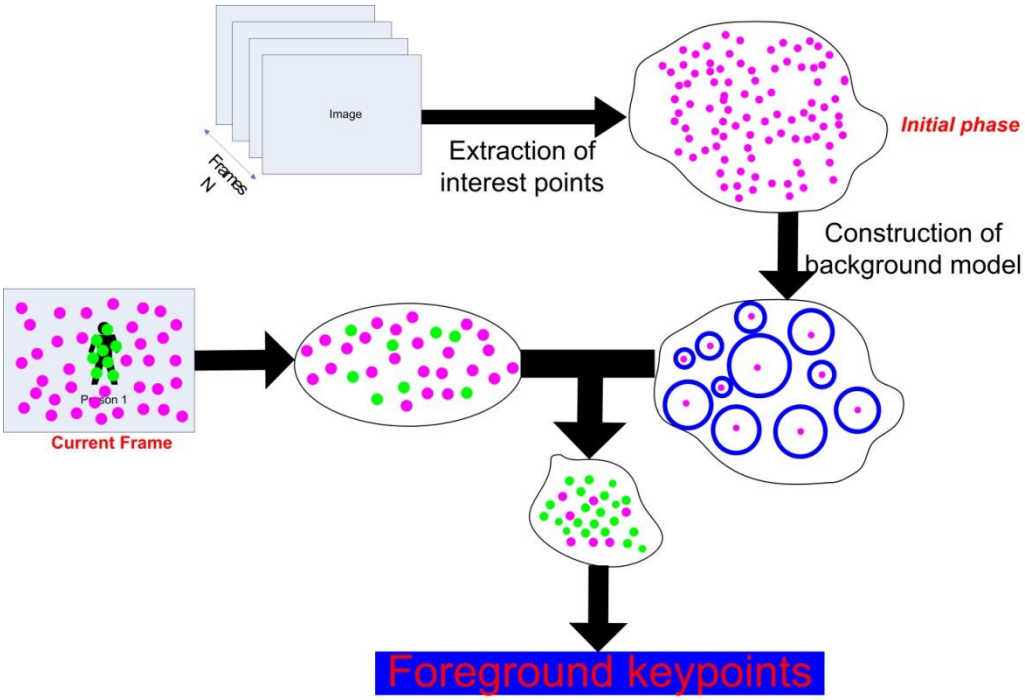


Figure 3-3: the block diagram of the method proposed for motion detection

3.2.1 Construction of the initial background

Let $P = \{p_1, p_2, \ldots\ldots, p_N\}$ be a training set of interest points. Let $B = \{b_1, b_2, .., b_L\}$ represent the interest points of the background. Each interest point consists of a descriptor vector= $[d_0, d_1, \ldots, d_{63}]$, and three parameters $(x_{p_i}, y_{p_i}, scale_{p_i})$ which are the location and scale of the point. For the background points, we add also f, the number of times with which the interest point has been matched, during the training period. Each point P_t at time t is compared to the current background points to determine which interest point of background B_m (if any) it matches (m is the matching background interest point's index). When determining the best match, we use spatial constraints to reduce the search space. The nearest neighbor in the background for each interest point is identified by using the sum of absolute difference (SAD). We then employ a nearest neighbor distance ratio matching (NNDR) which is proposed by Lowe (Lowe, 2004), we can define the NNDR as

$$NNDR = \frac{d_1}{d_2} = \frac{\left\|D_{p_i} - D_{B_m}\right\|_\infty}{\left\|D_{p_i} - D_{B_{m1}}\right\|_\infty} \leq r_d$$

Where d_1, d_2 are the nearest and second nearest neighbor distance, D_{p_i} is the descriptor of query and D_{B_m} and $D_{B_{m1}}$ are the closest two neighbors. This measure provides a dynamic threshold which will adapt to a variety of feature spaces due to the change of illumination. The disadvantage of using a fixed threshold is that it is difficult to determine. The useful range of global thresholds can vary depending on the change in illumination. The detailed algorithm to construct the background is given below.

Algorithm for background construction

1. $L \leftarrow 0(number\ of\ interest\ points\ of\ background), B \leftarrow \phi\ (background\ set)$
2. $for\ t = 0\ to\ N(The\ number\ of\ frames)$
   I. Extract interest point of image t.
   II. $for\ each\ points\ in\ image_t$
      a) Find the B_m in $B = \{b_j | 1 \leq j \leq L\}$ matching to p_i based on tree conditions
         - $(x_{p_i} - x_{B_m})^2 + (y_{p_i} - y_{B_m})^2 \leq \varepsilon_1$
         - $\left(scale_{p_i} - scale_{B_m}\right)^2 \leq \varepsilon_2$
         - $\left|D_{p_i} - D_{B_m}\right| < r_d * |D_{p_i} - D_{B_{m1}}|$
      b) If L= $\phi\ or\ there\ is\ no\ match\ then\ L = L + 1.$
         $Create\ a\ new\ backgound\ point\ by\ setting$
         - $D_{B_L} = D_{P_i}$
         - $f = 1, x_{B_L} = x_{p_i},\ y_{B_L} = y_{p_i}\ ,\ scale_{B_L} = scale_{p_i}$
      c) $otherwise, update\ the\ matched\ background\ point\ B_m, according\ to:$
         - $D_{B_m} = \frac{D_{B_m} * f + D_{p_i}}{f+1}$
         - $f = f + 1$
         - $x_{B_m} = \frac{x_{B_m} * f + x_{p_i}}{f+1}\ \ , y_{B_m} = \frac{y_{B_m} * f + y_{p_i}}{f+1}\ \ , scale_{B_m} = \frac{scale_{B_m} * f + scale_{p_i}}{f+1}$
      $end\ for\ (one\ image)$

End for (stage of training)

Algorithm 1: algorithm of building background using interest points

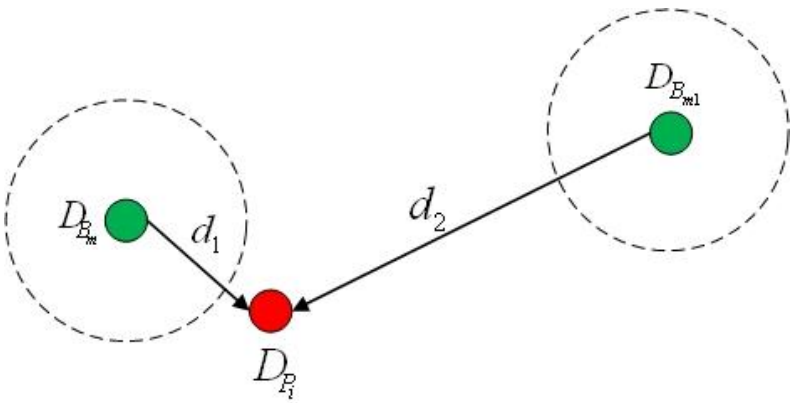


Figure 3-4: global threshold and NNDR modeling: we have 2 points in background D_{B_m} and $D_{B_{m1}}$ and new D_{P_i}, at a distance>global distance threshold (dashed circles), the descriptor of query point D_{p_i} fails to match D_{B_m} but using nearest neighbor distance ratio (NNDR) matching, the small NNDR= $\frac{d_1}{d_2}$ correctly matches D_{B_m} with D_{P_i}.

Classification: substracting the current image from the background model is straightforward. We simply compute the distance of each new interest point from the nearest interest point in the background and then use the NNDR measure. If there is no match, the point is considered as foreground. Otherwise, the new keypoints which match the interest point of the background are also used to continuously update the background model. The new keypoints in the image are added to the model, while the keypoints that have not been present for a while are removed. This allows for objects that quickly move in and out of the scene to be considered foreground, while new objects that come into the scene and remain stationary will slowly be blended into the background.

In figure 3-5 we show the result of the background construction. We used the first 20 frames to construct the background. The number of points at the end of construction the background is 410 points. We can see that the background keypoint substraction method extracts not only the points on the pedestrians but also new points from the background. There are two reasons for this problem: first when the pedestrian enters the background, some blobs in the background are divided by the pedestrian. The sub-blobs may be considered new points if the determinant of the Hessian in these points is bigger than the threshold which we used when we extracted the interest points.

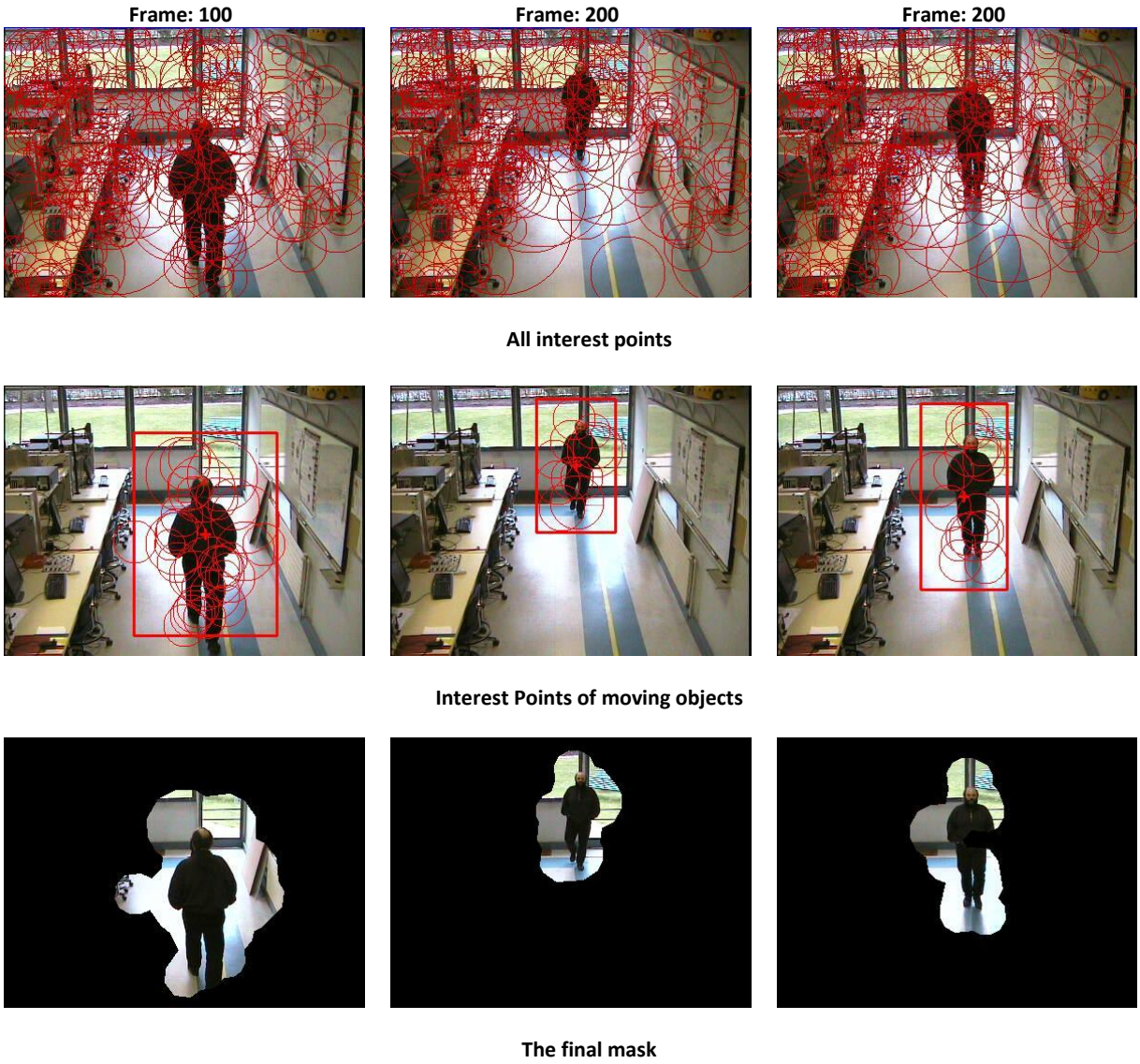


Figure 3-5: the result of background substraction using interest points.(Top line, all intereset points, middle line background points, bottom line the result mask).

The second problem of interest points is their instability at smaller scales, because they are more sensitive to changes in lighting and to camera noise and are often on edges. In addition, most of them describe background clutter or ambiguous objects, making them difficult to match. These unstable keypoints cause some false alarms.

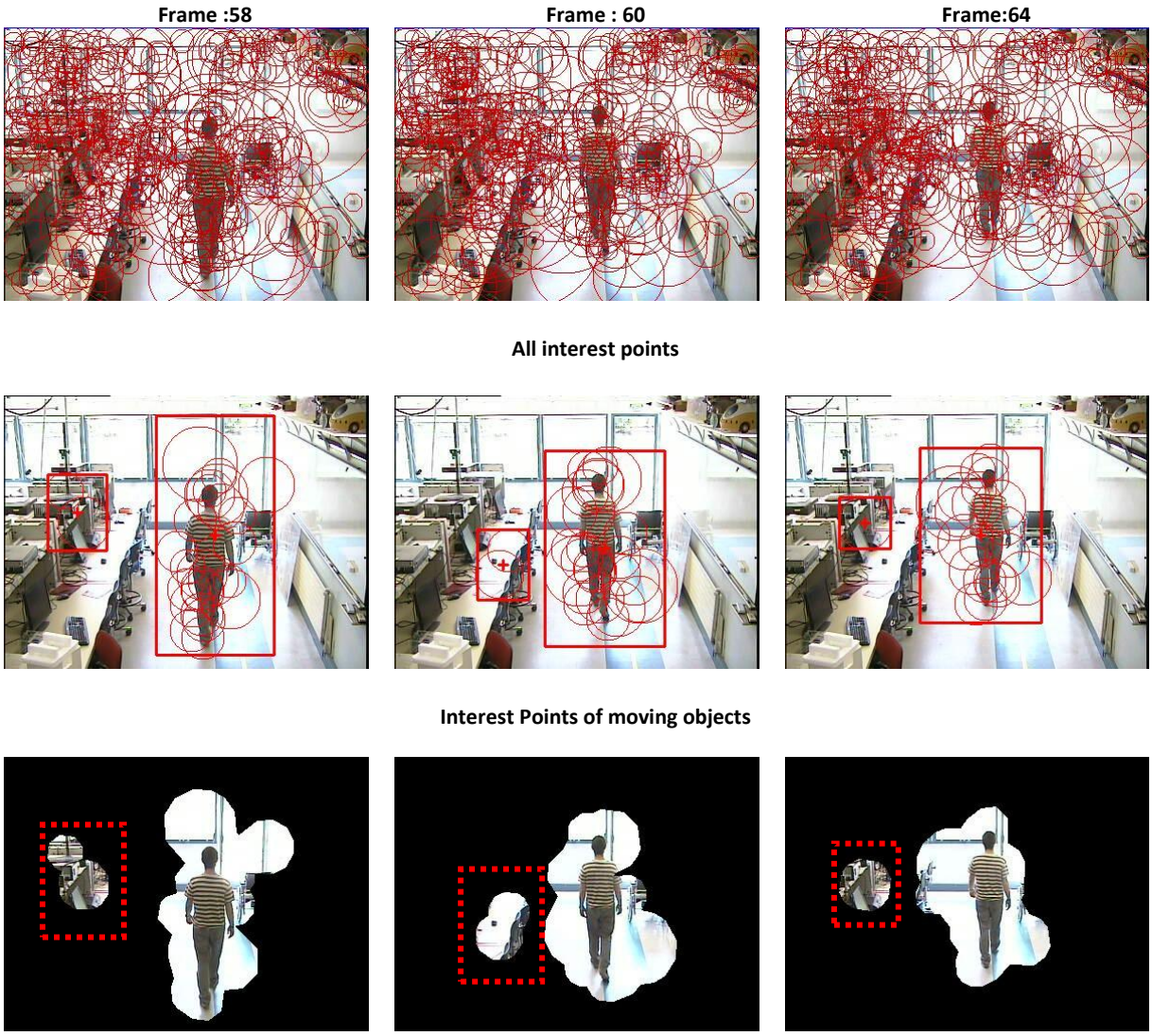


Figure 3-6: the result of background substraction using interest points, we note that the parasite points in the red dashed line rectangle.

In Figure 3-6, we construct the background using 15 frames. We can see that there are parasite points (determined by red dashed-line rectangle in Figure 3-6.) These points are unstable, i.e. they appeared and disappeared according to the noise of the camera. (The luminance of this camera is high which makes the image sensible to noise). In Figure 3-7 we can see the normalized histogram of interest points determinant of Hessian for the background and the foreground, the first bin which represents the smallest value of parasite interest points and the rate between the biggest and second biggest is increased for foreground due to the effect of these parasite points. The value of the determinant of Hessian is very close to the threshold we use (THR=400). A quantitative evaluation of the (rather poor) resulting foreground extraction is given later in §3.3.1.

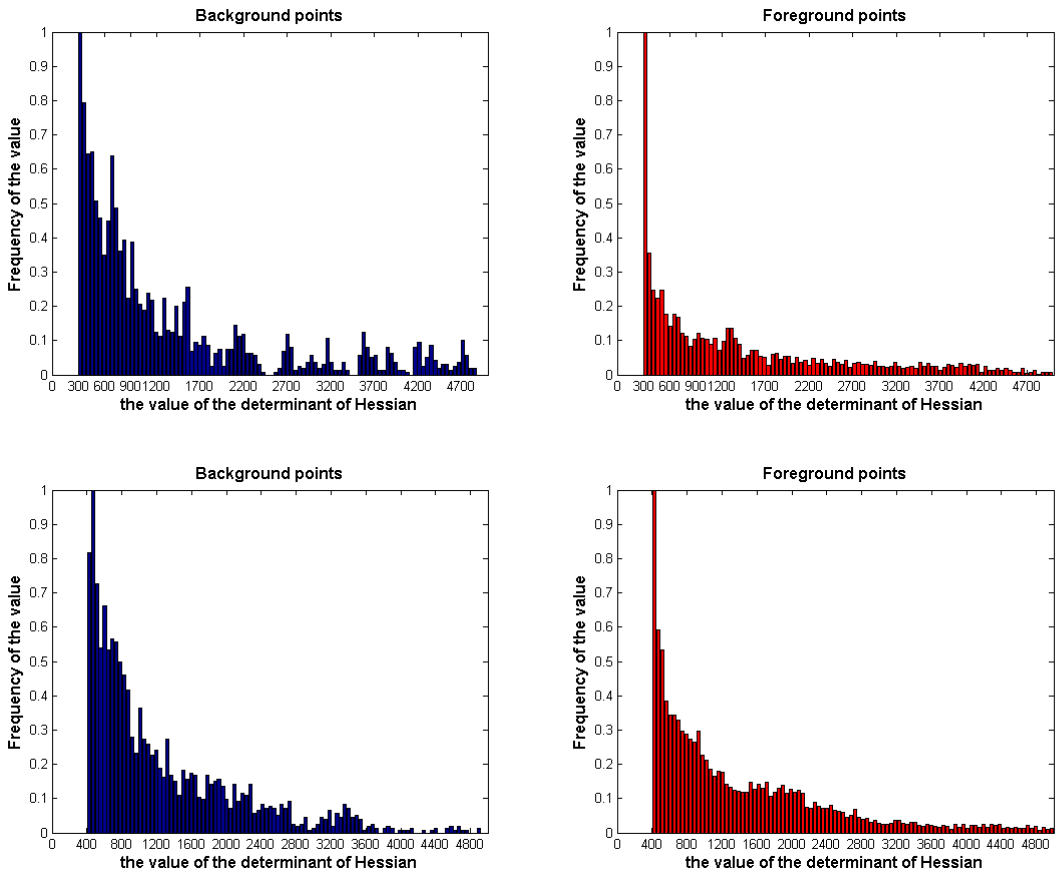


Figure 3-7: the histogram of the determinant of Hessian for two threshold (THR=300,400)

To overcome this problem, we can use a hybrid algorithm for change detection which integrates the pixel level algorithm as a codebook model (Kim, et al., 2005) with our algorithm. In this case, the mask of the movement pixel is more robust. In our case we want to use only the interest points. Therefore, we apply a second filtering step based on the keypoints classification using "classification and regression trees" (CART).

3.2.2 Adaboost Classification

The idea of this work is that when we extract keypoints from an image containing interesting objects, some of these keypoints may lie on the objects. As a consequence, the descriptors of some of these keypoints may contain information about the shape of the interesting objects. Hence, one should be able to infer the presence of a pedestrian in an image by studying the descriptors of the keypoints extracted from this image. Therefore we use CART to classify these descriptors.

CART "classification and regression tree" or decision tree proposed by Breiman et al (Breiman, et al., 1984) is a tree graph, with leaves representing the classification result and nodes representing some predicate. Branches of the tree are marked true or false based on this prediction. The classification process in the case of the decision tree is a process of going through the tree. We start from the root and descend until we reach the leaf. The value associated with the leaf is the class of the presented sample and the interior node represents the predicate. The general principle of

decision trees is to decompose the classification problem using a series of tests (rules in data mining) corresponding to a partition of the data space into subsets. At each step, the classification data is split to maximize the class purity within the two resulting subsets. Each subset is split recursively based on the independent tests.

The global training sample form the root node, when a selected statistical measure is performed, this samples are divided into two descendent node. The same process is applied to each node. These sequential splits will construct a tree decision space with the leaves node representing the assignment of classes. The unknown sample is assigned to the same class like leaf node which it falls finally into after scanning the decision tree. Different approaches can be used for the tree construction. In figure 3-8 we present the general algorithm used to construct the decision tree.

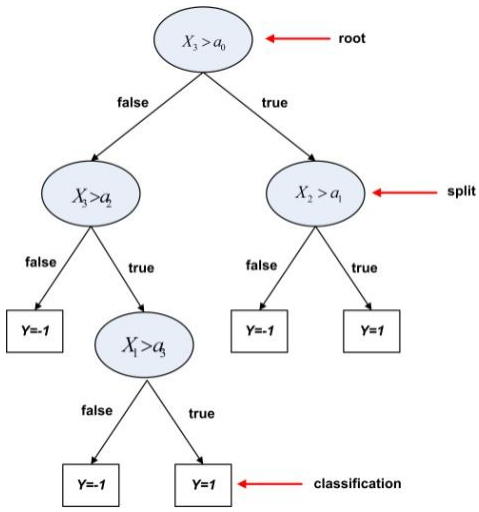


Figure 3-8: a decision tree classifier where the input dimension is three. At each of the root and internal nodes (splits), a statistical measure is applied. The values $a_0, \ldots, a_n$ are thresholds for splitting. A data set is subdivided into smaller groups down to the leaves which contain the class labels.

CART Construction

Input: sequence of m examples $\langle (x_1, y_1), \ldots, (x_m, y_m) \rangle$ where $y_i \in Y = \{-1, 1\}$ and with x_i belong to domain$X \in \mathcal{R}^n$.

1. ***Call construct node to build node root.***
2. ***Calculate the classification according to the prediction (threshold) of previous node.***
3. ***Find the leaf which has largest error.***
4. ***Call construct node using only subset data associated with leaf***
5. ***Split the previous data according to the predicate of the new node.***
6. ***Repeat 2-5 until all leafs have zero error, or predefined number of split (level in the tree)***

Construct node:

1. ***For each element of set and each dimension of $x_i^j \in \mathcal{R}$ find (l, θ) that divide the set of data with least error, where l is the associated dimension and θ is the threshold.***
2. ***Classify this set according to both (l, θ).***

Algorithm 2 the algorithm of CART construction

Freund and Schapire (Freund, et al., 1995) propose to use the AdaBoost algorithm to improve the performance of CART. The boosting algorithm iteratively constructs a series of trees (weak classifiers). The data set which constructs each tree has been filtered and weighted by the previously trained tree. Hence, the weights of incorrectly classified samples are increased so that the next tree (weak learner) is forced to focus on the hard samples in the training set.

3.2.3 AdaBoost training algorithm

The AdaBoost training algorithm has three steps which are repeated until a certain number of weak classifiers have been selected as shown in figure 3-9. Initially, the weights of the examples are assigned so that the sum over negative examples and the sum over positive examples are each equal to $\frac{1}{2}$. Then, for each cycle, the current best weak classifier, in terms of lowest weighted classification error on the training set, is selected and aggregated to form the strong classifier. At the end of every cycle, the weights are modified in order to make the training focus on examples that are misclassified by previous weak classifiers. The last two steps, aggregation and weight update, are generic. Only the first step depends on the context in which the AdaBoost algorithm is used. This will be described below. In particular, the relative weight assigned to the newly selected weak classifiers is automatically computed as $\alpha = \frac{1}{2} \cdot \log\left(\frac{1-\varepsilon}{\varepsilon}\right)$, where ε is the weighted classification error made by the weak classifier.

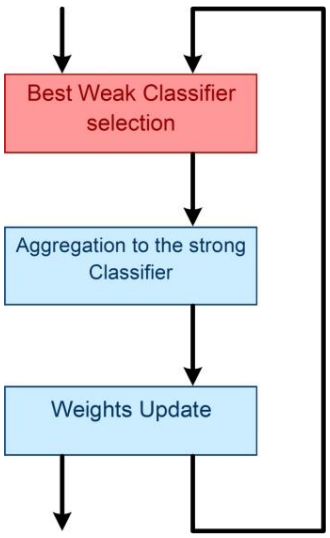


Figure 3-9: The steps of adaboost training algorithm

The keypoints filtering and learning algorithm consists of a classical AdaBoost algorithm. The idea of this algorithm is to combine several CART trees, which have poor performances, into one strong classifier which performs very well, using various weights. The key to the AdaBoost algorithm is to assign different weights to each example and to modify these at each step in order to enhance the relative weights of misclassified examples. The choice of the weak classifiers depends on the application.

The classification of keypoints in each decision tree is a process of tree traversing. We start at the root and descend. At each node, we compare the element of the keypoint descriptor with the threshold associated in this node. We descend to right or left depending on the result of the comparison, until we reach a leaf, which gives the label of a class. Hence, the result value is equal to 1 or 0. A weak classifier could be represented as in figure 3-10.

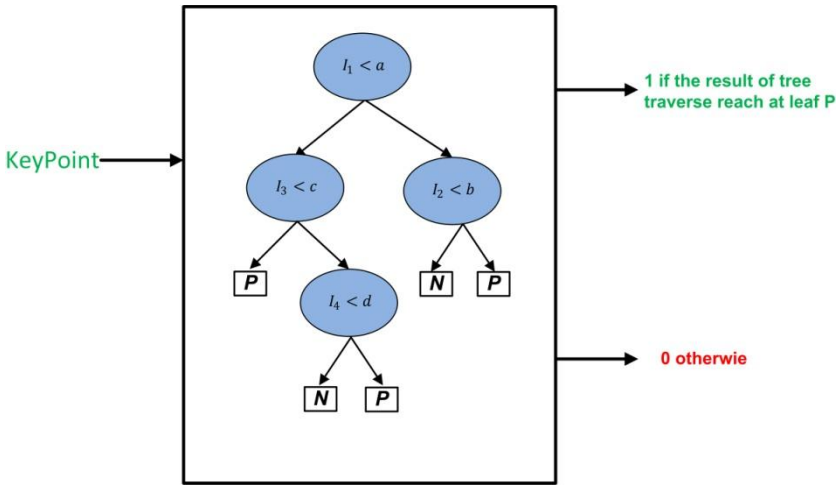


Figure 3-11: Synthetic representation of a weak classifier where $I_1, I_2, I_3, \ldots$ are the indexes of tested component descriptors and $(a, b, ..)$ are the thresholds

The strong classifier is characterized by a threshold T, a set of Weak Classifiers $\{CART\}$and their corresponding reliabilities $\{\alpha_i\}$. The strong classifier associates a label in $\{0,1\}$ to an input keypoint, where 0 means the negative class and 1 means the positive class. If the weighted sum of the values resulting from the weak classifiers $\{Cart\}$ is greater than the threshold T, the resulting label is 1 and otherwise 0. A strong classifier could be represented as in figure 3.12.

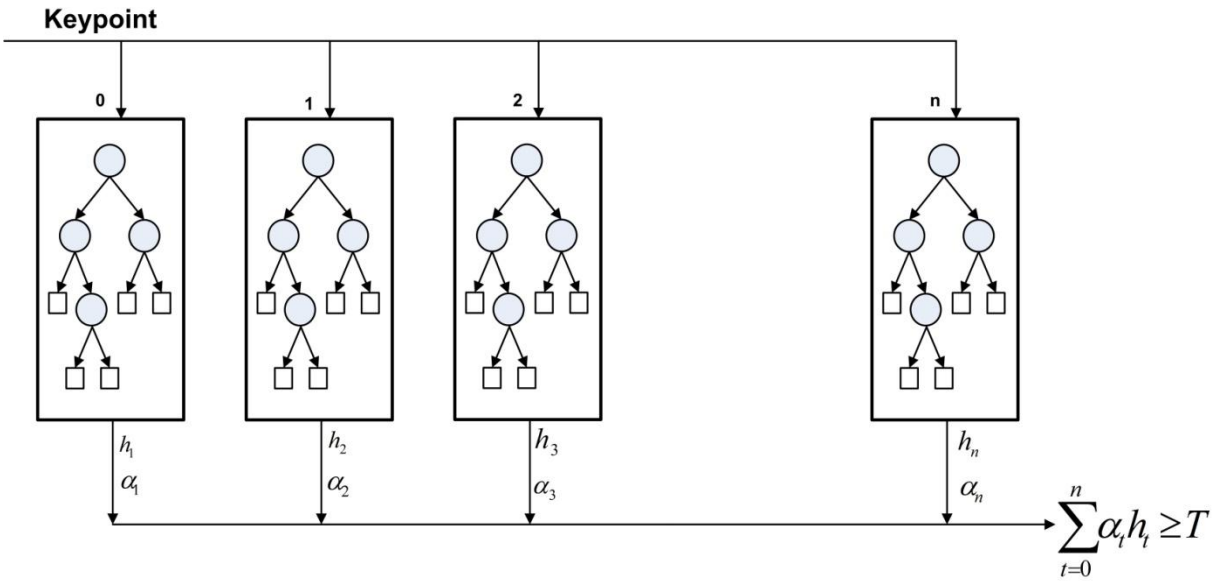


Figure 3-12: Synthetic representation of a strong classifier.

3.3 Results

To evaluate our detection and re-identification algorithms, we constructed a corpus containing 40 persons. Figure 3.15 shows the first 24 pedestrians, known as ID 01 to ID 24. No clothing constraint was imposed. Video sequences were recorded by two cameras with different perspectives and different illumination conditions (see figures 3-14 and 3-14). Two video sequences (one from each camera) were recorded for each person. Each person followed a similar path, walking through the scene in many directions, to enable the retrieval of images of the same person at different scales and angles (see figure 3-16).

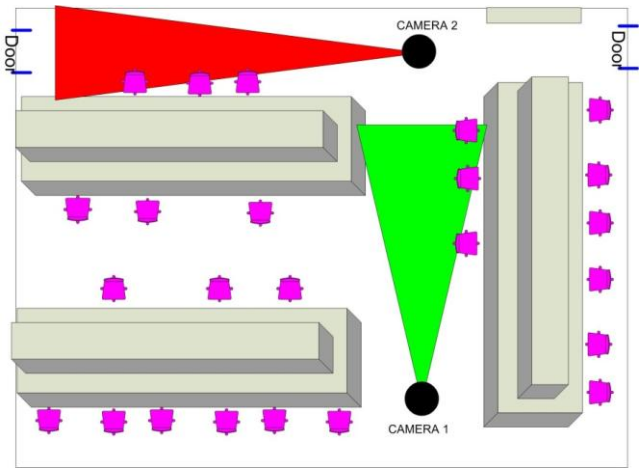


Figure 3-13: the position of two cameras and the field of view of two cameras. We note that the field of views of the cameras are not overlapping.

The video sequence has a rate of 15 frames per second. The images obtained by the first camera form the models. The evaluations were formed using images taken by the second camera. The silhouettes were extracted manually. We construct the ground truth using SamGT[5], which can display ground truths of several types (pedestrian, vehicle…). In the ground truth of our corpus we note the position of the pedestrian and the window containing the pedestrian. The window's height is twice as large as its width, and the pedestrian is surrounded by margins of about 10%.

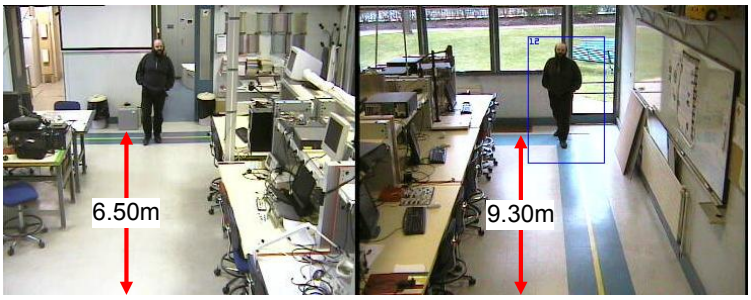


Figure 3-14: Images illustrating the different illumination conditions for video sequence acquisition. (Left: the camera sequence used for extracting training points for the Adaboost classifier, right: that of the second camera used for validation)

[5] SamGT: an internal tools to build geometric ground truth

The twenty four pedestrians from first camera (training & validation set)

The last sixteen pedestrians from second camera (test set)

Figure 3-15: indoor image sequences taken by two cameras collected in our laboratory

Figure 3-16: illustrating different scales and pedestrian poses.

3.3.1 Background Substraction results

To evaluate the performance of the classifier, we used the dataset of the second camera which contains 11.000 images with 6.500 images of 16 pedestrians which are not included in the training dataset. We used the criteria used in the Pascal VOC challenge to compare the ground truth and the estimated bounding box based on the ratio of intersecting and their union area:

$$\frac{\text{area}(BB \cap BB_e)}{\text{area}(BB \cup BB_e)} \geq \frac{1}{2}$$

Where BB is the ground truth bounding box and, BB_e is the estimated bounding box. This criterion is often used in object detection literature in order to compare new techniques. The number of estimated bounding boxes matching one and only one ground truth bounding box allows for the computing of a precision value p and a recall value r using these formulas: $p = \frac{TP_{loc}}{TP_{loc}+FP_{loc}}$ and $r = \frac{TP_{loc}}{TP_{loc}+FN_{loc}}$ where TP_{loc} is the number of True Positives, FP_{loc} is the number of False Positives (i.e. the number of false alarms) and FN_{loc} is the number of False Negatives (i.e. the number of misdetections). Groups of people cannot be determined only by analyzing the spatial relations between individuals, which makes the detection and tracking very difficult for artificial systems. For this reason, we decide to restrict the evaluation to individual bounding boxes only.

We can see in Fig3-17 and Fig3-18 the performance of the background substraction for different values of the threshold of determinant of Hessian. We note that the background substraction performs poorly because of the influence of parasite interest points. The best performance is obtained when we use a threshold of 500 and 15 images. We get the precision of 53% and a recall 60% for NNDR=0.9.

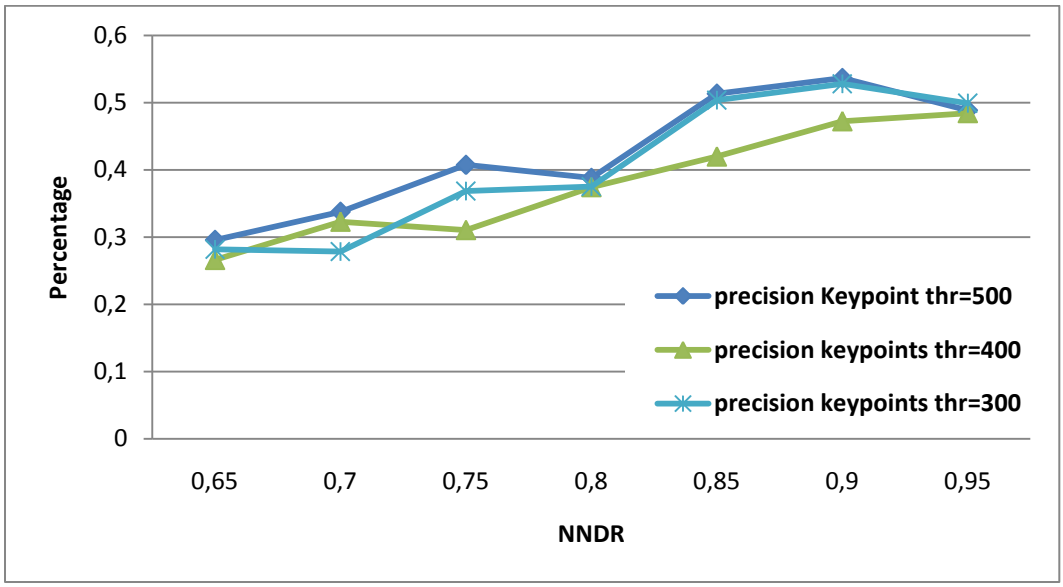


Figure 3-17: chart shows the influence of NNDR on the precision using only background substraction

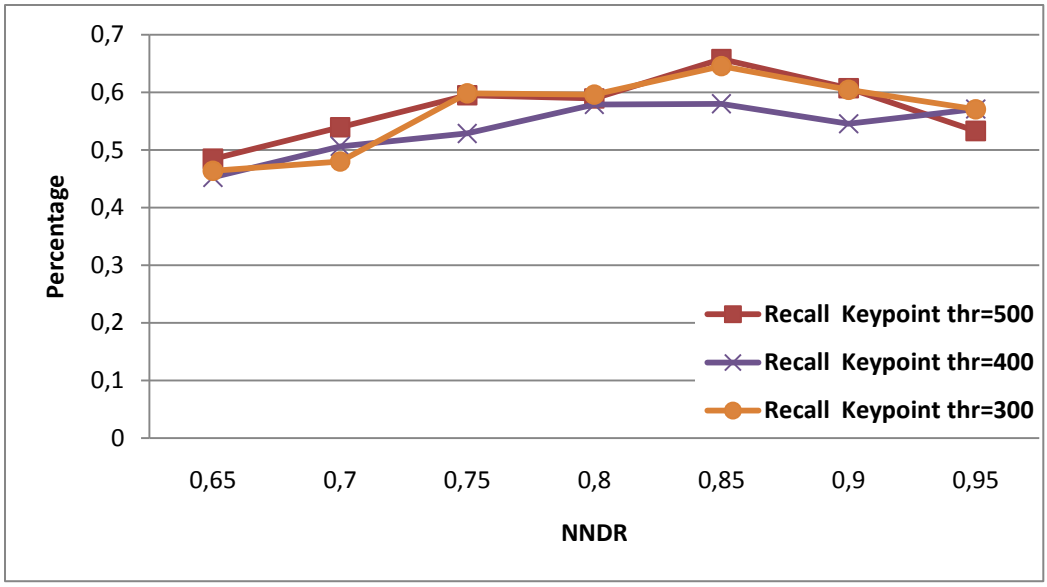


Figure 3-18: chart shows the influence of NNDR on the recall using only background substraction

3.3.2 Keypoint classification results

We collected keypoints from the first 24 pedestrian images from the first camera (camera 1 in figure 3.13). There are about 50.000 keypoints for pedestrians recorded by this camera with different illumination, scale and pose and 250.000 keypoints from the non-pedestrian samples taken with same camera. Figure 3-19 represents the positive interest points (red circles) and the other interest points considered as negative. We change the parameter of camera's focal length and the iris to capture the most variance of non-pedestrian interest points.

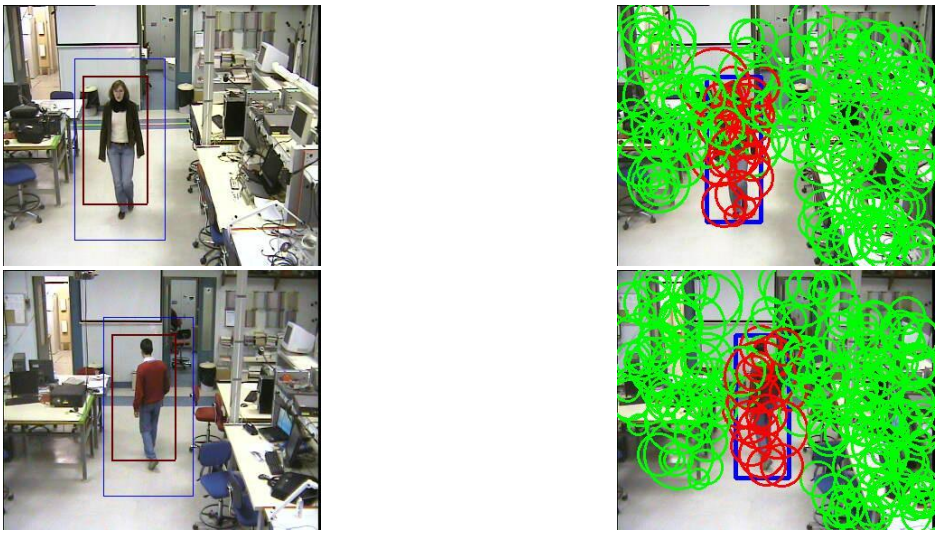


Figure 3-19: the ground truth used for classification of keypoints. We rescale the bounding box (red rectangle) to take only the interest points of the pedestrian.

For training and classification, we adopt the GML Adaboost Matlab Toolbox which was provided by the Graphics and Media Laboratory of Computer Science Department at Moscow State University and Matlab 7.2 for training/constructing our strong classifier. During the training, we use the Gentle AdaBoost (GAB) and Real AdaBoost algorithms to train the classifier. We can see in the figures 3-20 and 3-22 the evolution of the strong classifier's classification training and validation errors for the two algorithms. The performance of Gentle AdaBoost is slightly better than Real AdaBoost.

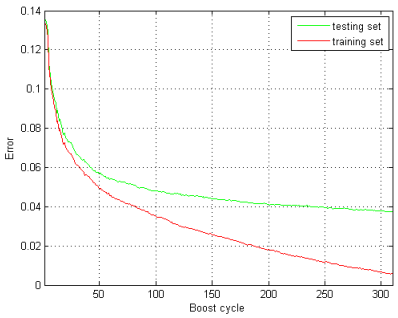


Figure 3-20: Evolution of the strong classifier classification error on the training(red) and validation (green) set using real AdaBoost algorithm

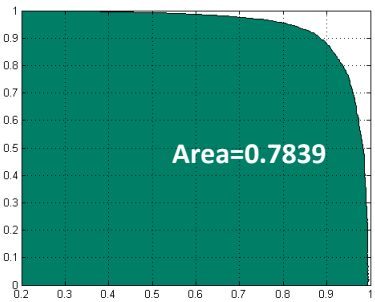


Figure 3-21: Precision/Recall curve on the validation set using real AdaBoost algorithm

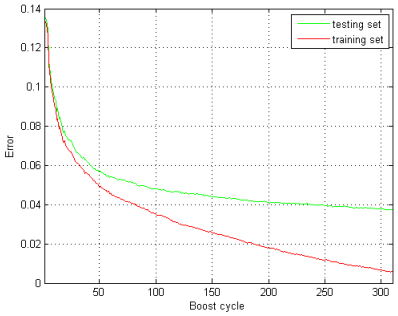


Figure 3-22: Evolution of the strong classifier classification error on the training(red) and validation (green) set using Gentle AdaBoost algorithm

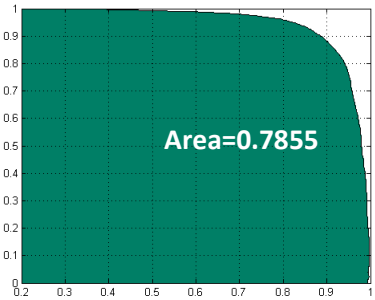


Figure 3-23: Precision/Recall curve on the validation set using gentle AdaBoost algorithm

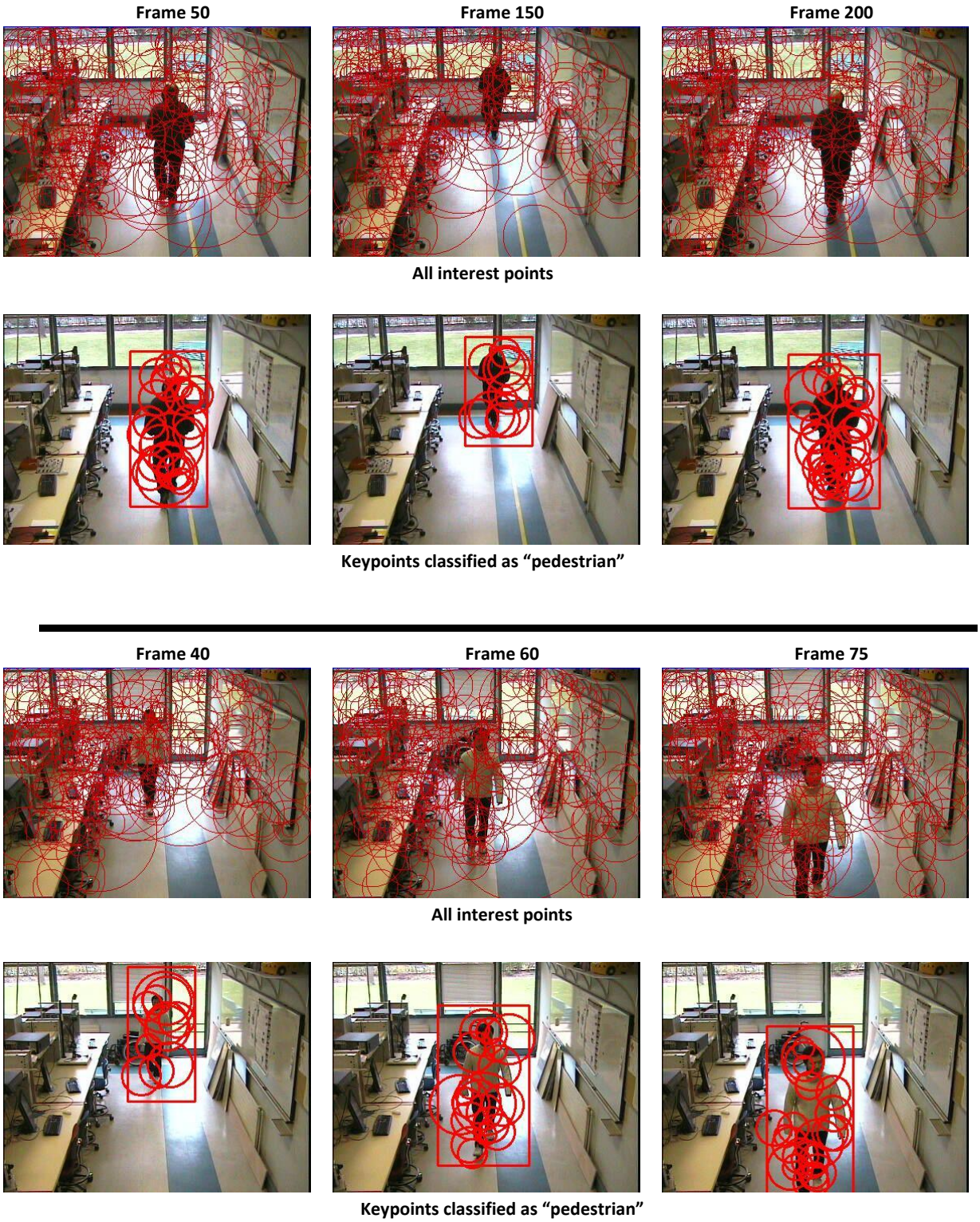


Figure 3-24: Some results of adaboost classification to all keypoints.

We note in the figure 3-24 that the classification filtered the interest points which were associated to the pedestrian. We use the Gentle AdaBoost classifier and the same threshold for keypoints which were used to construct the background (TH=400). We can see the result of classification using the images of the second camera. In figure 3-25 we can see the result of Adaboost classification on different illumination and scale conditions.

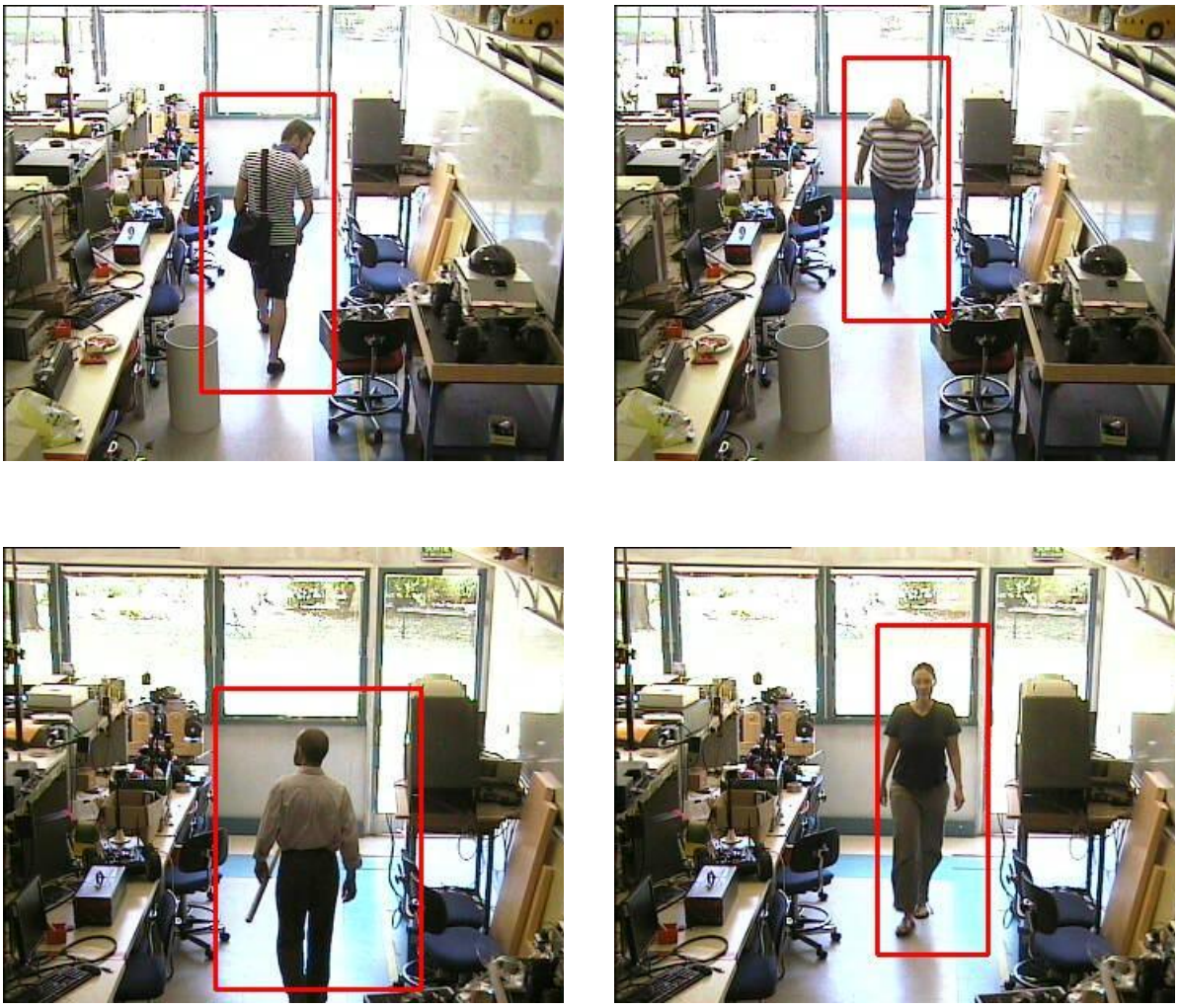
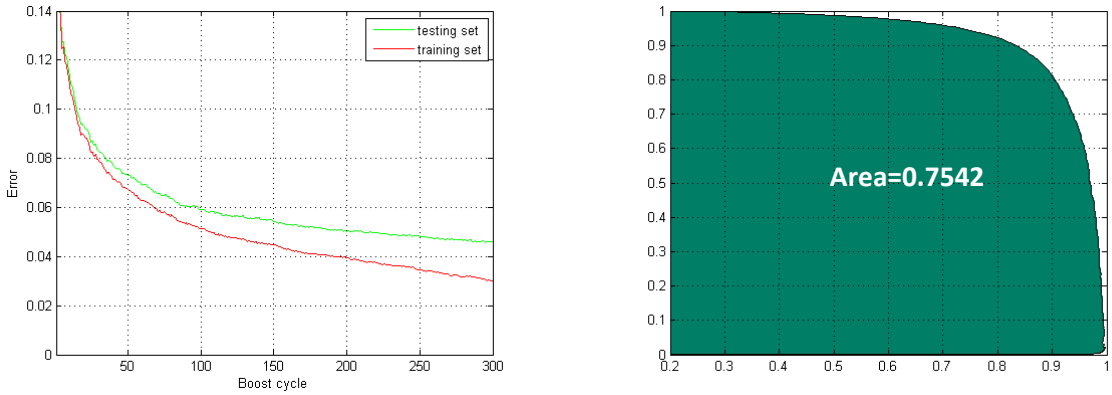


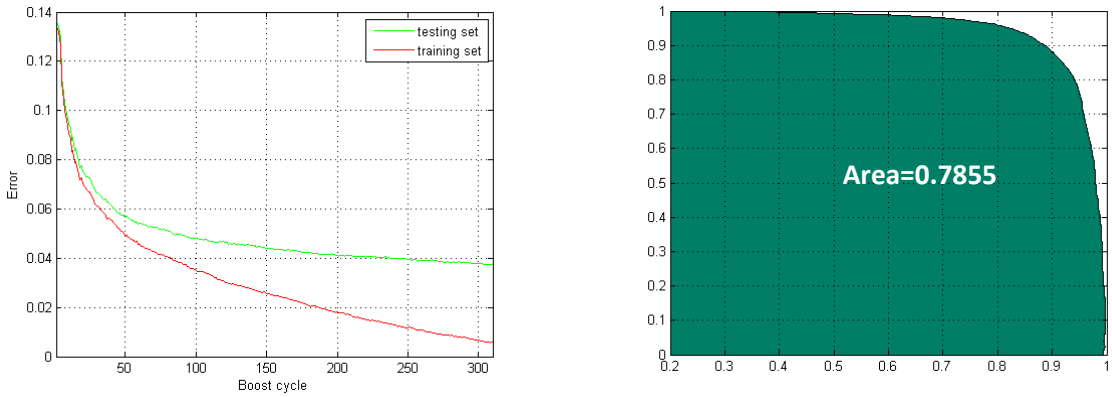
Figure 3-25: the results of the proposed method on different illumination and scale conditions

We also study the optimal number of splits (the maximum number of levels in each weak classifier). We tested 4-split, 8-split and 16-split. We note that when the number of splits is increased, the error is decreased but the calculation time is also increased. In figure 3-26, we can see the classification error of the strong classifier using Gentle AdaBoost with various values of split:

Evaluation of the strong classifier classification error and precision recall curve using 4-split



Evaluation of the strong classifier classification error and precision recall curve using 8-split



Evaluation of the strong classifier classification error and precision recall curve using 16-split

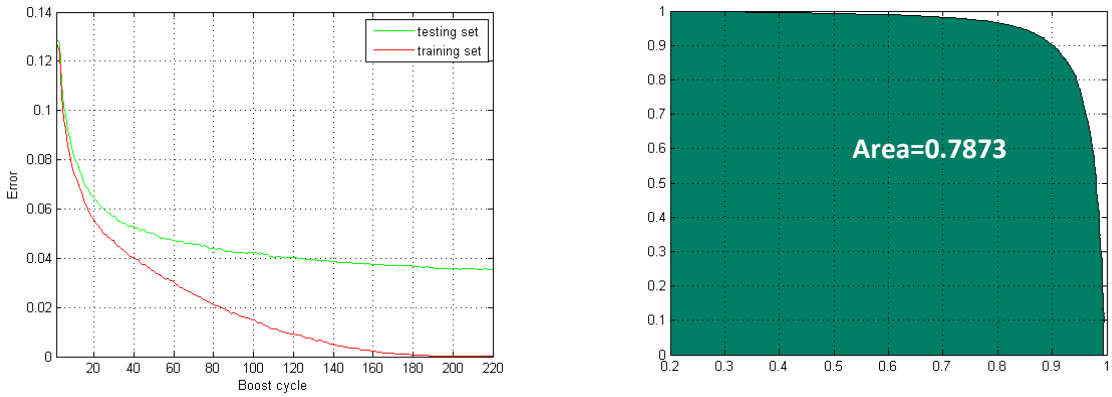


Figure 3-26: Evaluation of the strong classifier classification error and precision recall curve different value of split

3.3.3 Evaluation of Keypoints filtering with Adaboost only

In the previous section, we looked only at the keypoint classification result on validation set. Now we evaluate the performance of keypoints filtering using this classifier on an independent test set,.To evaluate the performance of the classifier, we used the dataset of the second camera which contains 11.000 images with 6.500 images of 16 pedestrians not included in the training dataset. The confusion matrix (Table 3-1) is used to analyze the relationship between the real input and the output of the classifier. The quality of the classifier is measured by the number of correct and incorrect classifications in each cell of the confusion matrix:

Table 3-1: the confusion matrix

		Input	
		Pedestrian keypoints	**Non pedestrian keypoints**
Output	**Pedestrian keypoints**	TP	FP
	Non pedestrian keypoints	FN	TN

where TP is the number of True Positives (i.e. the number of good detections), FP is the number of False Positives (i.e. the number of false alarms), FN is the number of False Negatives (i.e. the number of misdetections) and TN is the number of true negatives. We can see in the next three tables the influence of increasing the number of levels (weak classifiers)

Table 3-2: the confusion matrix for 4 split

		Input	
		Pedestrian keypoints	**Non pedestrian keypoints**
Output	**Pedestrian keypoints**	**60531 (84.0%)**	**26383**
	Non pedestrian keypoints	**11519**	**590098 (95.7%)**

Table 3-3: the confusion matrix for 8 split

		Input	
		Pedestrian keypoints	**Non pedestrian keypoints**
Output	**Pedestrian keypoints**	**62458 (86.6%)**	**25265**
	Non pedestrian keypoints	**9592**	**591216 (95.9%)**

Table 3-4: the confusion matrix for 16 split

		Input	
		Pedestrian keypoints	**Non pedestrian keypoints**
Output	**Pedestrian keypoints**	**63481 (88.1%)**	**25682**
	Non pedestrian keypoints	**8569**	**590799 (95.8%)**

We note that the performance of 16 split is more accurate. The number of false negatives decreases when the number of splits increases, yet the computation time and the number of iterations which depend on the number of splits increases. Table 3-5 shows the time needed to classify 81000 points using a portable computer equipped with a dual core CPU 2.00 GHz and 2 GB RAM. We empirically choose the 8-split CART to study the effect of the iteration number.

Table 3-5: classification time

Type of split	Computing Time
4 split	112µs each point
8 split	237µs each point
16 split	800µs each point

The quality of the trained classifiers was evaluated using recall and precision. These measures are defined as follows:

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{Total\ \#\ positives}$$

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{TP + False\ alarms}$$

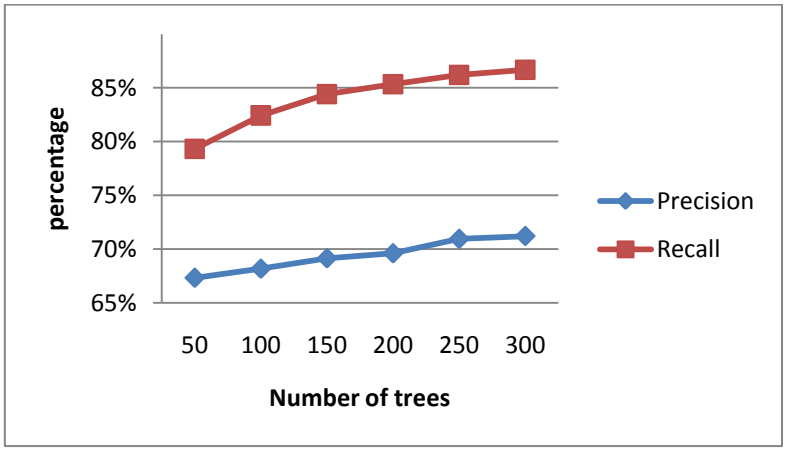


Figure 3-27: the effect of the number of trees on the precision and recall

As shown in fig 3-27 when the number of trees increases, the precision and recall also increase. However, the amount of increase of the recall is higher than that of the precision. In this case, increasing the number of trees augments the number of true positives but the effect on the false negatives is less significant. We can augment the precision (decrease the false alarms) by changing the ratio of 1:5 (50000:250000) positive to negative keypoints to 1:10, when we train the detector. In figure 3-28 we can see the effect of Adaboost threshold on precision and recall.

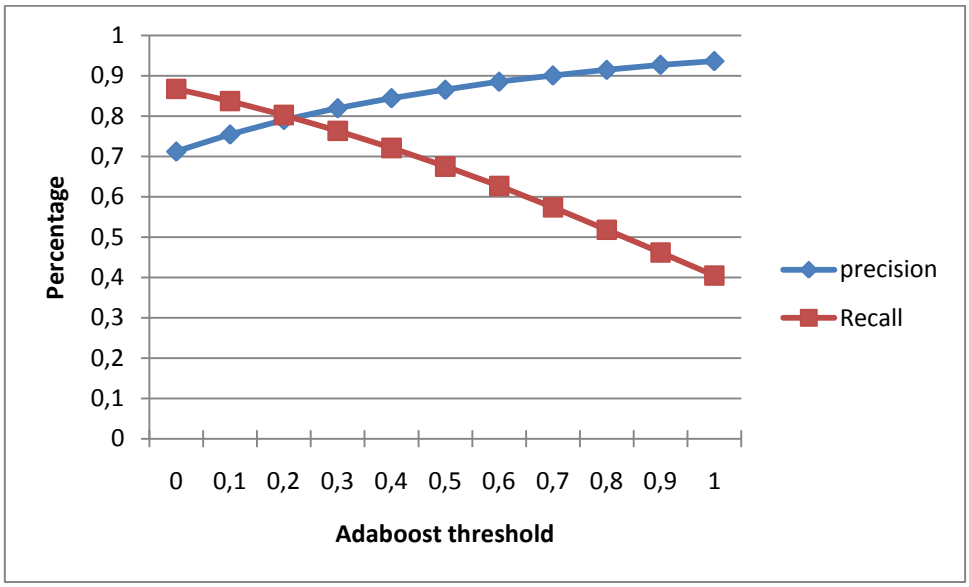


Figure 3-28: the effect of Adaboost threshold on precision and recall. When we use threshold 0.2, the precision is 80% and the recall is 80%.

3.3.4 Evaluation of the cascade filtering

In order to obtain the best result, we use a cascade approach, where the background substraction is applied first. Then the keypoint classification is done on foreground keypoints. Figure 3-29 illustrates the corresponding block diagram of pedestrian detection by cascade processing of keypoints. The role of the background substraction is to decrease the false alarms as shown in figure 3-30.

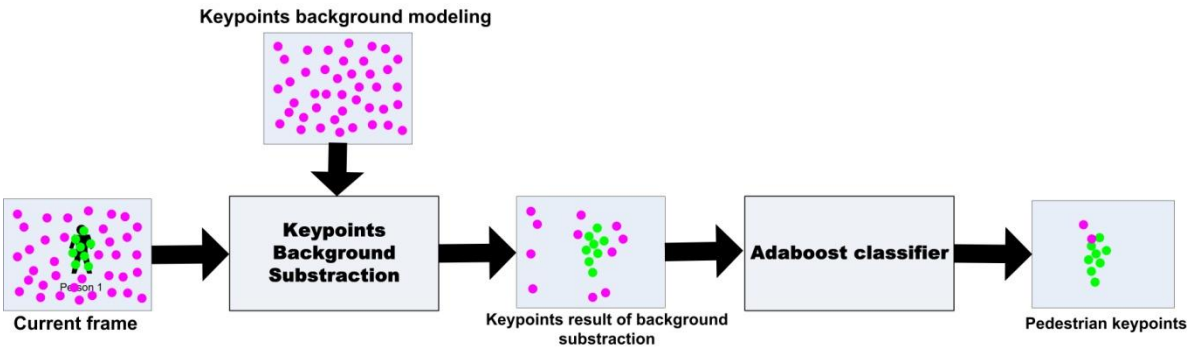


Figure 3-29: the block diagram of the cascade proposed for keypoints filtering

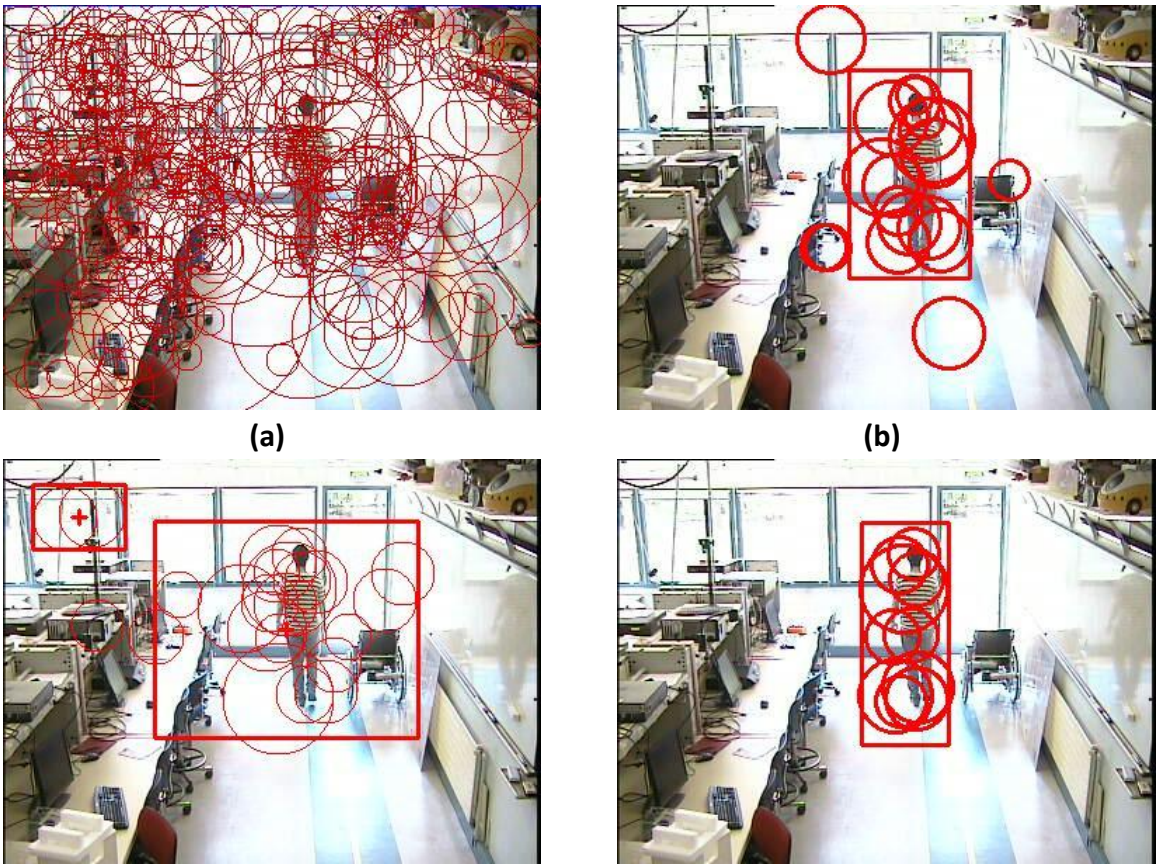


Figure 3-30: the result of the cascade: (a) the keypoints of current frame (b) the result of AdaBoost classifier (c) the result of background substraction (d) the result of the cascade

To evaluate the performance of cascade we used the same dataset and applied also the criteria used in the Pascal VOC challenge (see §3.3.1). We study the influence of using the cascade (Background substraction, adaboost classifier), as shown in figure 3-31 and figure 3-32. The recall decreases slightly when we use the background substraction, however the precision increases in most cases. For Adaboost only we obtain the best performance (a precision=90% for recall=87%) when we use the Adaboost threshold =0.5. When using the cascade, the best performance (a precision=93% for recall=90%) is for use Adaboost threshold=0. It is worth noting that the number of points for the 16 persons is 2,5 millions points, and using of background substraction decreases by 10 fold the number of points which pass to the second stage, down to 240000. So the computing time is decreased significantly.

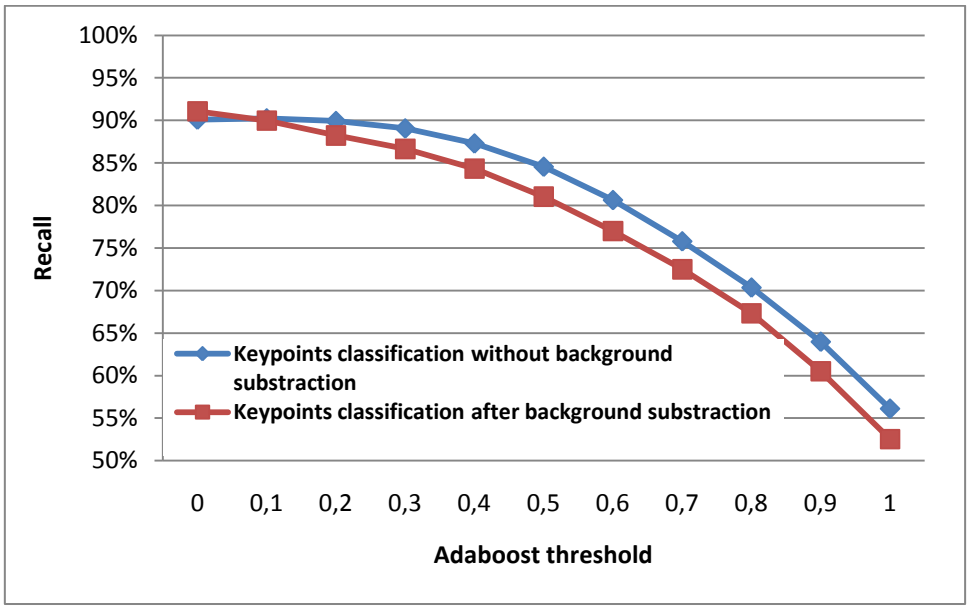


Figure 3-31: Algorithm recall when we use only Adaboost and when we apply background substraction before Adaboost

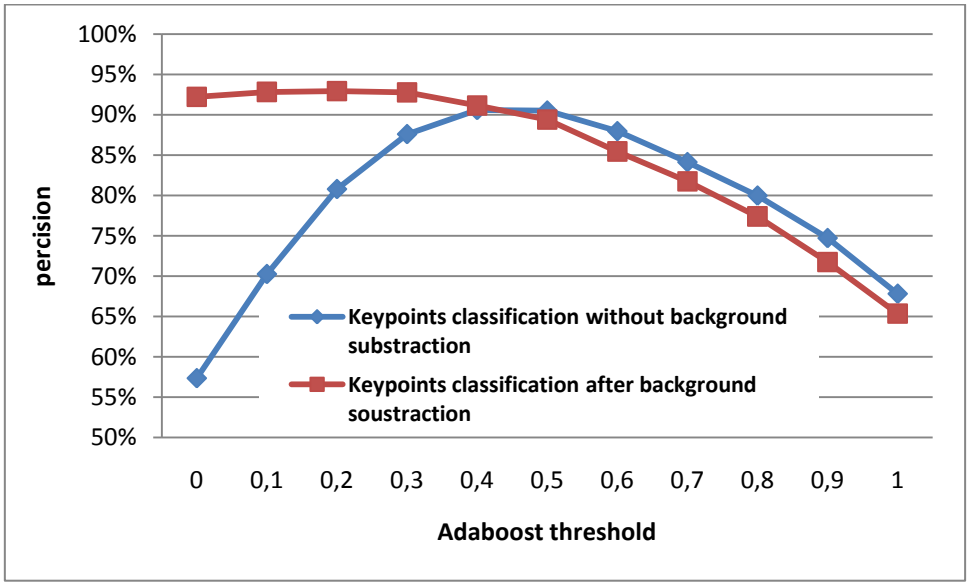


Figure 3-32: Algorithm precision when we use only Adaboost and when we apply background substraction before Adaboost.

3.4 Comparison with HoG detector

To further evaluate the performance of the proposed detection method, we compare our results with those of the HoG descriptor, which is commonly used in recognition and detection. For this, we begin by using the HoG detector (Dalal, et al., 2005) implemented in the open source library OpenCV. Table 3.6 shows the recall and precision rates for the HoG and the keypoints detector tested on 16 pedestrians, not included in the training dataset, where we can see that their performances are comparable. The best HoG detector performance lies at a precision=93.06% for recall=89.63% and threshold = 1.0. The best performance of our method lies at a precision=93% for recall=90% with an Adaboost threshold=0.2.

Table 3.6: the recall and precision rates for OpenCV HoG detector and the keypoints detector tested on 16 pedestrians not included in the training dataset

	HoG detector		Keypoints detector	
threshold	**Precision**	**recall**	**Precision**	**recall**
1	**93.06%**	**89.63%**	65.33%	52.53%
0.8	90.73%	91.96%	77.37%	67.35%
0.6	82%	93.85%	85.43%	77.63%
0.4	72.97%	94.90%	91.13%	84.31%
0.2	38.07%	95.73%	**93.20%**	**90.23%**
0.0	28.85%	95.97%	92.20%	91.05%

Moreover, the time needed to analyze an image (352 x 288) using HOG is about 52 ms (Intel Core i7 CPU, 2.93GHz, and RAM4GB). The detection computation time using our method is reduced to about 31 ms. And this time measurement includes the extraction of interest points also used in the re-identification step, so the actual extra time required in our case for pedestrian detection is much smaller than HoG detection.

A further advantage of our method is the ability to detect partial silhouettes. Since we detect the keypoints of a pedestrian, there is no need to retrain the detector on partial images of pedestrians. Our detector however cannot separate the interest points of two pedestrians occluding each other. In this case, the HoG detector solves this issue using a sliding window.

3.5 Conclusion

We have presented a method for modeling a scene and detecting a pedestrian based only on interest points. This algorithm consists of two stages: background modeling with keypoints, and filtering these keypoints using an Adaboost classifier. The background substraction is achieved by extracting the interest points of N frames. Then the modelling of the background is performed depending on their descriptors and the locations of the points. A point is considered as belonging to the foreground, if there is no match between its descriptor and the descriptor of any point in the background model. The drawback of this method is the influence of parasite interest points on the extraction of foreground. Therefore, we add a second stage which filters the foreground interest points by using Adaboost. The experiments on our dataset have shown a good detection performance (a precision of 90% for a recall of 87%) when we use only the Adaboost classification, which is further augmented to a precision of 93% and a recall of 90% when we use the cascade, which has the advantage of being much faster. According to our experiments this detection performance is comparable to state-of-the-art pedestrian detection such as HoG detector, which is more computer-intensive.

A more general conclusion is that, due to keypoints variability, a simple background model is not very effective, and some learning is needed to robustly discriminate the foreground pedestrian keypoints from the background keypoints.

We will now focus, in the next chapter, on the crucial task of pedestrian re-identification between non-overlapping cameras.

4 Re-identification

Motivation

We present in this chapter a system that addresses the problem of pedestrian re-identification. As mentioned earlier, the work presented in this chapter is expected to meet two objectives:

1. Propose a method for identifying a person using the matching Interest points found in several images.
2. Produce quantitative results on the performances of such a system to allow an objective comparison with other features (SIFT, Color, HOG).

Introduction

In many video-surveillance applications, it is desirable to determine whether a presently visible person, vehicle, or object, has already been observed somewhere else in the network of cameras. This kind of problem is commonly known as "re-identification". A general presentation of this field in the particular case of person tracking can be found for instance in §7 of (Tu, et al., 2007). Re-identification algorithms have to be robust even in challenging situations caused by differences in camera viewpoints and orientations, varying lighting conditions, pose variance, and also, for general appearance of persons, rapid change in clothes and with limited training data available.

A first category of person re-identification methods relies on biometric techniques (such as face or gait recognition). Face identification in "cooperative" context on high-resolution images with well-controlled pose and illumination can now be achieved with very good performance (see (Belhumeur, et al., 1997) and (Draper, et al., 2003)). A second group of methods performs person re-identification using only global appearance and without biometrics. There is a standard assumption: people do not change their clothes between successive appearing in the cameras. This assumption is usually quite reasonable in many applications such as surveillance of airports and metro. Among these, various approaches have been proposed: signature based on color histograms (such as in (Park, et al., 2006) and (Pham, et al., 2007)), texture characteristics (see (Lantagne, et al., 2003)). More recent works have proposed the use of matching interest points for establishing the correspondence between objects, like cars in (Arth, et al., 2007), and also for person re-identification as for instance in (Gheissari, et al., 2006).

This chapter is organized as follows: section 4.1 describes the state of the art methods of re-identification based on global appearance. Our re-identification method is described in section 4.2, and quantitative evaluation is presented in section 4.3, improvement of the method is presented in section 4.4. Section 4.5 provides an evaluation on a larger and public dataset (ETHZ). Finally the comparison with different signatures on benchmark datasets is presented in section 4.6.

4.1 Related works

In first chapter, we considered the problem of person tracking when the subject is constantly in the area of coverage of a network camera. When the targets are moving in and out of coverage area,

different labels would be created for one object. In this chapter, we consider the problem of person reacquisition which links these disconnected labels using a model of appearance.

There are two aspects of the person re-identification problem. First we must establish the correspondent, that is to say what parts of an object should be compared to parts in the second object. Then we must build the invariant signatures to compare the corresponding parts. Figure 4-2 represents the general structure of the re-identification method. Pedestrian are detected, tracked, and then a sequence is created for each pedestrian. The colors in the two cameras are calibrated. Then the feature vectors are built from each pedestrian. The model is constructed from the features of each object. Then the features are matched, using a distance metric which can cope with the variability of the objects. The score given to the matching between the two objects is based on the cardinality of the final set of correspondences. Table 4.1 gives an overview of research on surveillance and multi-cameras re-identification. We classify the re-identification methods depending on the features used to represent the parts of an object. We can distinguish three categories: template matching methods, color histogram method and local features methods.

Figure 4-1: Some examples from our pedestrian data set. Each column is one of 40 person example pairs. Note the wide range of viewpoint, pose, and illumination changes.

4.1.1 Template matching methods

Several approaches have been proposed where the signatures are invariant for each individual based on his global appearance. The first recognition approach is based on "Template Matching", which integrates the spatial information and the appearance (Lipton, et al., 1998). This approach is generally not robust to change of orientation, thus several methods propose to use the a priori of object shape to build a offline model of the object (Ning, et al., 2004), (Moeslund, et al., 1999), (Stauffer, et al., 2001). The method tries to predict the pose of object in the next camera, then, it finds the corresponding shape to project this model in the image plane for comparison with the image data. Another shortcoming of template matching is that it is sensitive to illumination. To avoid this disadvantage Huttenlocher et al (Huttenlocher, et al., 1993) use Edge templates and The

Hausdorff and Chamfer distances, which are mechanisms for efficiently comparing edge templates with some robustness to slight misalignments.

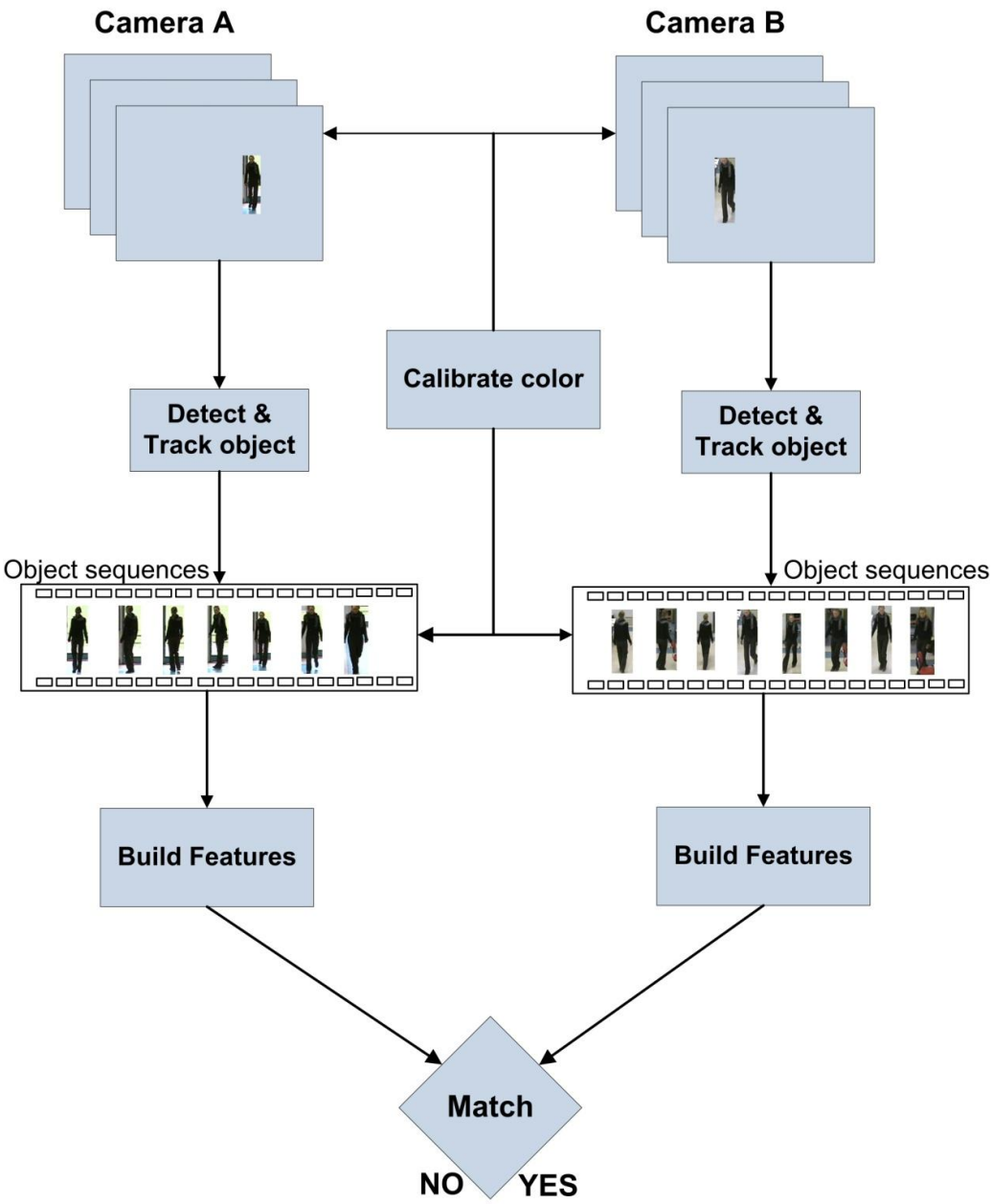


Figure 4-2: Diagram shows the general structure of the re-identification method. Pedestrians are detected, tracked, and constructed object sequence . The colors of the two cameras are calibrated. Then the feature vectors are built from each object. The features of each object construct a model. Then the features are matched, using a distance metric which can combine the variability of the objects. The score given to the match between the two objects is based on the cardinality of the final set of correspondences.

4.1.2 Color histogram based methods

The second approach and the most common appearance model is the histogram of colors. This model is invariant to shifts in position, to rotation, and to scale change. Jaffre and Joly (Jaffré, et al., 2004) and Pham et al (Pham, et al., 2007) proposed to use the histogram of RGB color model below the face (found by a face detector). The disadvantage of this representation is that it is not very distinctive, because all the information of spatial structure is lost. This can be improved by splitting the object into a number of subregions and computing a histogram for each subregion, as done by Park et al (Park, et al., 2006) who divide the detected person into three parts from top to bottom (at $1/5^{th}$ and $3/5^{th}$ of the height), they take only the color of middle and bottom parts as features. Another approach is proposed by Huang et al (Huang, et al., 1999) by adding the spatial information to the histogram. This new image signature is called the color correlogram (correlation of color with distance). Yu et al (Yu, et al., 2007) estimate the distribution of color and the length path which is the shortest distance of a pixel inside the silhouette of person to the head. In this case they integrate the colors information of pixels with its spatial information. This integration is done using kernel density estimation. Once the model of a person is built, the correspondence between silhouettes is found by using the Kullback-Leibler distance. Madden et al (Madden, et al., 2007) describes the pedestrian by its main color, they use the online K-means clustering algorithm to obtain the major color histogram which they called the Major Color Spectrum Histogram representation (MCSHR). This histogram is computed over a short window of successive frames.

Another disadvantage is the fact that color appears to be different in varying lighting conditions. Colors are sensitive to changes of lighting conditions, and change in cameras. Several approaches are proposed to calibrate the color between cameras. Porikli in (Porikli, 2003) propose to find the model function to calibrate the inter-camera colors. This method is based on the cross-correlation matrix between the histogram of a known object seen in the two cameras and dynamic programming. In (Finlayson, et al., 2005) Finlayson et al use a gray image enhancement technique: Histogram equalization on each channel of a color image, they claim that the rank ordering of sensor responses are invariant to change in illumination condition. Therefore, the histogram equalization of each channel makes the image invariant to these conditions. Javed et al (Javed, et al., 2005) estimate brightness transfer function (BTFs) for a pair of cameras during initial phase. They assume that BTF lies in a subspace of the space of all possible BTFs. They use probabilistic PCA to calculate this subspace using a set of common targets in each pair of cameras. The correspondent of an object is found by comparing its BTF against this subspace. Prosser et al. extended the BTF in (Prosser, et al., 2008) to a Cumulative Brightness Transfer Function (CBTF). Firstly they find CBTF using a sparse training set. Then they adapt this function using a combination of the original CBTF and the background illumination changes at each camera. Orwell et al in (Orwell, et al., 2001) assume the change of illumination between two cameras is given by an affine transform. They use Expectation Maximization (EM) training to find the parameter of this transform using set of common objects into the two cameras; they also studied the camera capture noise to find optimal histogram quantization intervals.

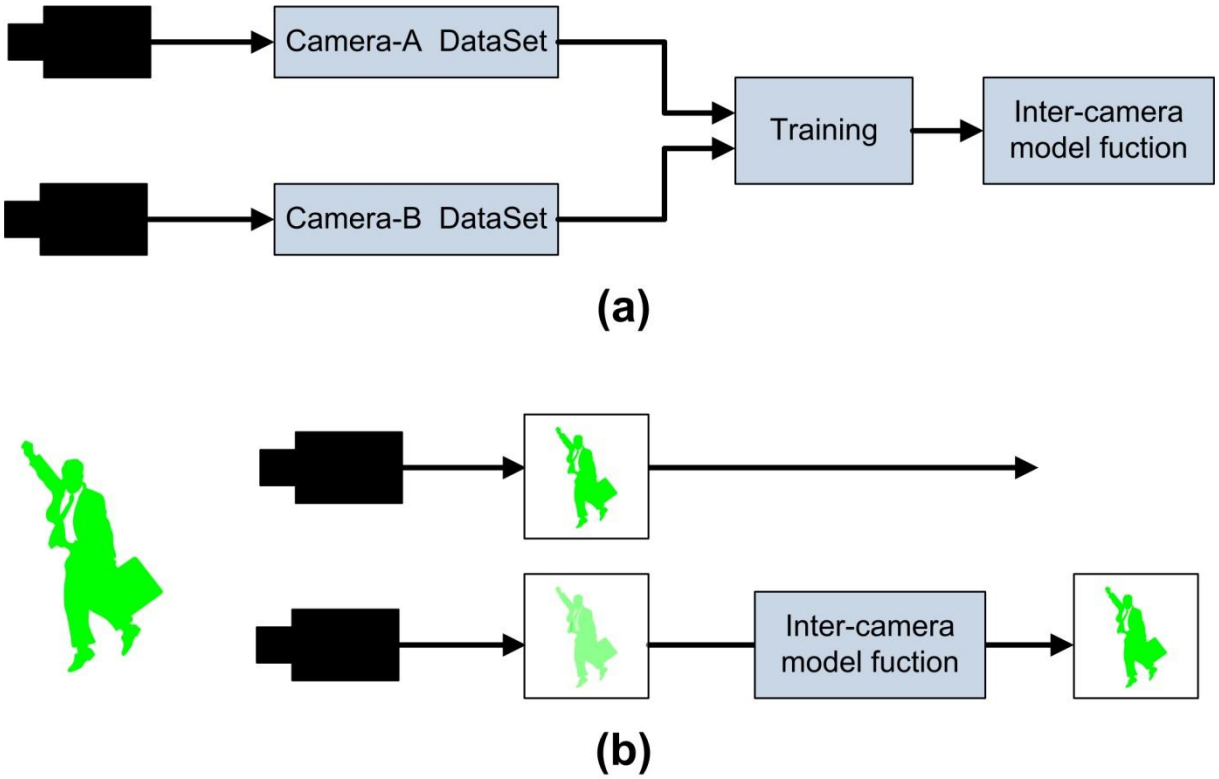


Figure 4-3: Diagram of the general structure of inter-camera color calibration. (a) A multi-camera setup, after generating camera data sets of videos. Inter-camera model function is found by training. (b) Using the model function obtained in the first stage. The output of the second camera is compensated to match the colors with the first camera.

Apart from the compensation of the change of illumination, several approaches use machine learning algorithms to discriminate features of different appearance. In addition to finding the discriminating feature, these methods try to reduce the dimension of the model which represents the global appearance of the pedestrian; Nakajima et al (Nakajima, et al., 2003) use the multiclass SVMs to perform the person recognition. The SVM classifiers are built for each pedestrian using a set of its global color features and local shape-based features. Bak et al (BAK, et al., 2010) propose to use Adaboost to find the discriminative features for each pedestrian. They use Haar features of each pedestrian to train the strong classifier of Adaboost. They use also Dominant color descriptor (DCD) which describes the main color of the upper and lower part of the person and the percentage of this color. Schwartz et al (Schwartz, et al., 2009) divide the pedestrian into a set of overlapping blocks. Then they construct the feature vector for each person, which consists of three types of feature co-occurrences, histogram of oriented gradient (HOG) and the histogram of each block using the HSV space. After that, they find the discriminative feature and at the same time, they reduce the dimension of the primary feature by using partial least squares (PLC). Cong et al (Cong, et al., 2009)characterize the silhouette using normalized color histogram or spatiograms. Then they use these features to construct a graph where its vertices are the histograms of all persons in two sequence and the edges are the distances between these histograms. Next they compute the Laplacian Eigenmap of the distance matrix. They use the first twenty eigenvectors to create the new coordinate system to project the n frame of each person. Then similarity is calculated between the barycentres of each person. The authors of all recent approaches determine the appearance model

depending on the features they use to re-identify the pedestrian. However, Gray et Tao (Gray, et al., 2008) do not determine the model. They define a space of features which they believe gives a good representation of object. Then they use Adaboost to construct the best model for the training data available. They use this model to identify the person where the model defines a set of simple color and texture feature (Gabor filter) and some spatial and intensity constrains.

4.1.3 Local features based methods

Several approaches use the segmentation to improve the representation of the model. In (Lantagne, et al., 2003) Lantagne et al begin by dividing the silhouette into three parts. These three regions (upper, middle and bottom) correspond to the head, trunk and arms, and legs. Each part of the silhouette is segmented into regions using the JSEG (Deng, et al., 2001) algorithm. Then for each region they calculate the color descriptors using HSV space. They take only the dominant color which has a percentage greater than 5%. Then they integrate this descriptor with the texture descriptor which is extracted by applying four edge detectors and calculating the sum of responses for each detector. Gheissari et al (Gheissari, et al., 2006) use the watershed algorithm to segment each silhouette. To avoid the folds and wrinkles in clothes they construct a spatio-temporal graph where each node represents a region and the edges represent the spatial and temporal relations between the adjacent regions. The salient regions are found by the partition of the graph which models each cluster as a minimum spanning tree. The histogram of each region is extracted, finally the correspondence between the signatures are established using decomposable triangular graph model. Farenzena et al (Farenzena, et al., 2010) propose to integrate maximally stable color regions (MSCR) with HSV Histogram and Recurrent High structure Patches (RHSP) which estimate the texture in the patches using the threshold of entropy. The three types of features are weighted by the distance with respect to the vertical axis of silhouette. Recently Alahi et al (Alahi, et al., 2010) propose to detect and track the objects simultaneously using a cascade of grids of region descriptors. They show that the cascades which consist of histograms of orientation outperform the descriptors based on the colors of objects.

Most recent works use global appearance. But since 2000 the works that use the local features have achieved good results in object recognition as in (Lowe 2001), (Sivic et Zisserman 2003), (Mikolajczyk et Schmid 2001). The main idea of these works is to generate a large number of points. The descriptors of these points are invariant to scale and rotations. These points are integrated into one or more models. The model consists of a 3D representation of the object in the approach of Lowe (Lowe 2001), which increases the probability to find true correspondents between two sets of features representing the two objects. In (Mikolajczyk et Schmid 2001) Mikolajczyk and Schmid use points of interest (Harris-Laplacian) to index images with significant rotation, translation and scaling. The images are characterized by a set of points. The disadvantage of interest points is their non-persistence over long periods of time due to the dynamic nature of the appearance of a person. Hence, in (Gheissari, et al., 2006) Gheissari et al use the interest points to re-identify the pedestrian. The Hessian affine invariance is used, which gives a large number of points, then they use the histogram of orientation (HOG) and the HSV histogram as descriptor, the region size is fixed. They use the dual matching to find the correspondence between the two sets of points representing two objects. Finally they expand the region of each primary pair correspondence. Then if the distance

between the two new signatures is below a threshold, they consider that there are true correspondences between the two interest points. Arth et al (Arth, et al., 2007) propose to use the PCA SIFT for object re-identification in large scale smart camera networks. First, they create a vocabulary tree using the training images. Then, for each object, the interest points are detected and their descriptors are calculated. The features of each object are built by finding the nearest neighbors for each interest point in the tree. The indices of the matched leaves represent the object signature. Hence the number of identical elements between the signatures of two objects determines the possibility of matching between these objects.

Table 4-1: Overview of research on monitoring and multi-camera re-identification.

System	Objective	Identification technique
VIP : Vision tool for comparing Images of People (Lantagne, et al., 2003)	Recognize, in real time, a person on different angles	Descriptor of color and texture descriptor
ViSE: Visual Search Engine Using Multiple Networked Cameras (Park, et al., 2006)	To aid the monitoring operation video and find people	The detected blob of a person is divided into three parts from top to bottom. Then they used the HSV histogram to construct a model
MONNET: Monitoring Pedestrians with a Network of Loosely-Coupled Cameras (Albu, et al., 2006)	To aid the operation of video surveillance, and find people	Build a model using the color appearance (Lantagne, et al., 2003)and face for every pedestrian observed and compare with the models received from other cameras
KNIGHT: A real time surveillance system for multiple overlapping and non-overlapping camera (Javed, et al., 2003)	They have a distributed system of smart cameras that detect, monitor, and classify moving objects	Estimate the path of object using Bayesian formulation and using the histogram of color, they use BTF between each pair of cameras.
A Multi-Camera Visual Surveillance System for Tracking of Reoccurrences of People (Pham, et al. 2007)	IDSS to track re-occurrence of items	Histogram of color
Person Re-identification Using Spatiotemporal Appearance (Gheissari, et al., 2006)	Re-identification of pedestrian	Points of Interest (Hessian operator) and a graphical model of triangular decomposition.
Object reacquisition and tracking in large-scale smart camera networks (Arth, et al., 2007)	Re-identification of vehicle	Interest points detector and PCA SIFT descriptor and 'Vocabulary Tree' integrated in DSP

4.1.4 Kd-Tree search

When using interest points matching, the search method used is critical for the performance of the algorithm. In this section we will present the Kd-tree approach which is very efficient and that we use extensively in our work. A Kd-tree is a data structure for storing a set of K-dimensional features. It was used in KNN search for the first time in (Friedman, et al., 1977). The idea of a kd tree is to partition the (multi-dimensional) sample space according to the underlying distribution of the data. The partitioning being finer in regions where the density of data points is higher. We can see the Kd-tree as a Hash table where the size of the bin is adaptive to local density of stored features.

For the feature set E in the space D, each node in the Kd-tree, which contains this set represents one feature, each node splits the set into the subset according to the feature element where the set of features are the most spread. All the features in the "left" subset are represented by the left subtree, and the points in the "right" subset by the right subtree. Let i be the index of the element used to split the set. Then a feature is to the left of the current node if and only if its $i-th$ element is less than the $i-th$ of element of the feature which represents the node. The complimentary definition holds for the right field. If a node has no children, then the splitting is stopped.

Figure 4.4 show a Kd-tree representation of a two dimensional dataset. Here seven different examples A-H are shown as small circles arranged on a two dimensional plane, the root node which represents the value D, splits the plane in the y-axis into subsets.

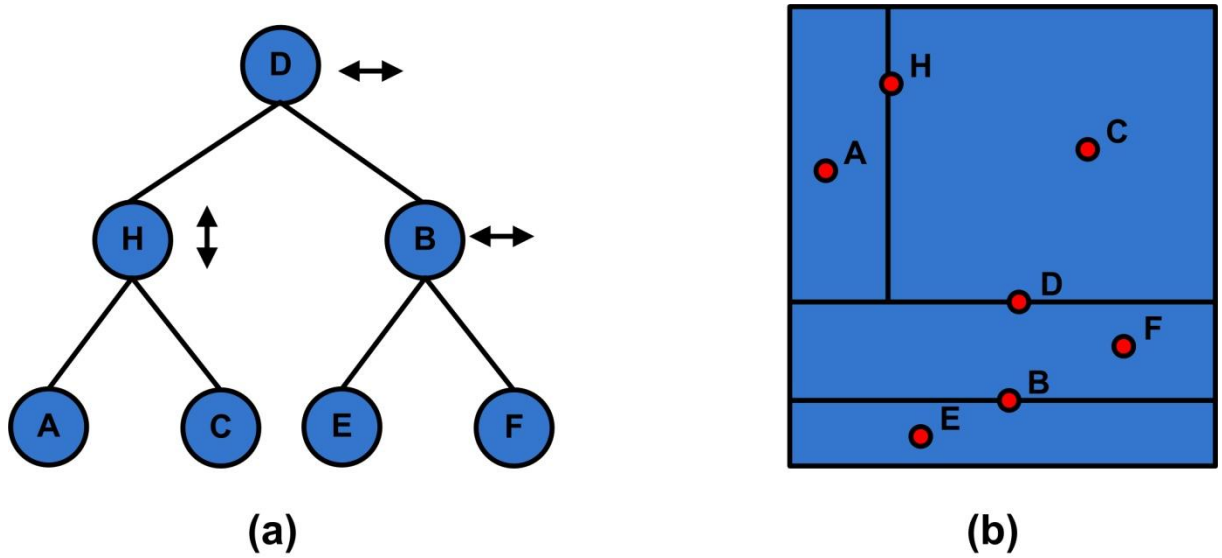


Figure 4-4: A Kd-tree construction: (a) 2d-tree of seven elements, the arrows besides the nodes indicate the direction of the splitting. (b) The tree in (a) split up the x,y plane.

At query time, a classic Kd-tree search first finds the leaf node which contains the target features. In figure 4.5 the target feature is marked x. It is not necessary that this leaf contains the nearest neighbor but at least we know any potential nearer neighbor must lie closer, and so it must lie within the circle centered on x and passing the leaf node. So it must be followed by a backtrack process which tries to find the best candidate by checking other leaves of tree. The terminating condition of

the search in any branch is if the distance between the parent leaf of this subtree and the query feature is farther than the distance from the query to the closest neighbor yet seen.

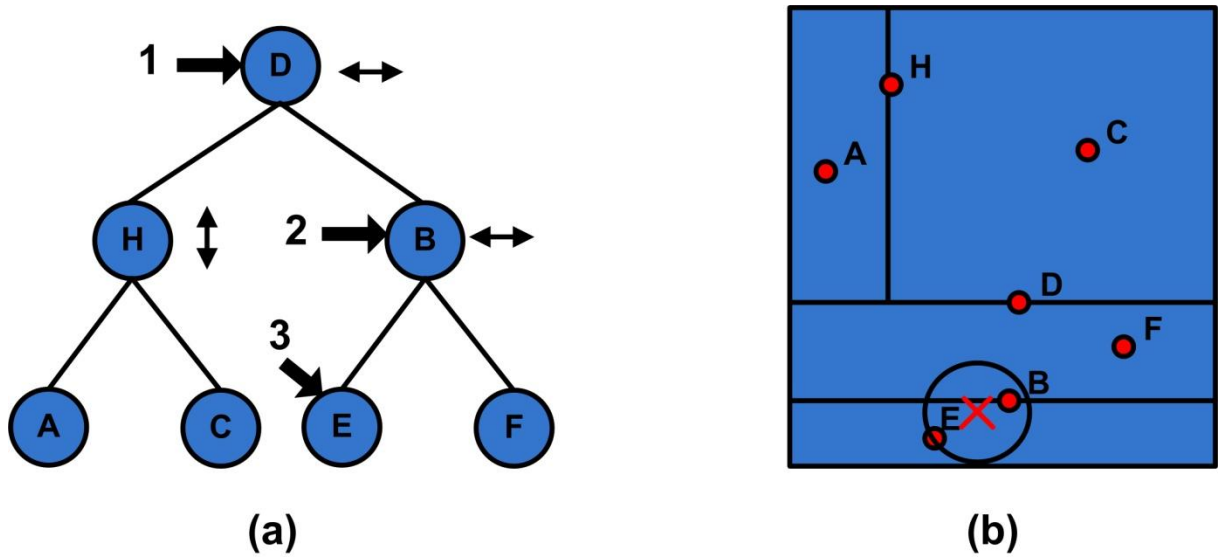


Figure 4-5: (search of a KD-tree) In this figure, a query point is represented by X and its closest neighbor lies in node B. A priority search first descends the tree and finds the node that contains the query point at the first candidate (label E). However, a point contained in this cell is often not the closest neighbor. The backtrack process finds the node B which is the nearest neighbor and changes the minimal distance between the query and model and terminates the search.

This process is very effective for low dimensional features, but this search strategy loses its efficiency in high dimensional kd-trees, because it requires searching a very large number of nodes to guarantee that the nearest neighbor has been found. This is very heavy in terms of computation time. Arya et al in (Arya, et al., 1993) find empirically that the classical algorithm used in kd-tree usually finds the nearest neighbor before the search terminates, and that all the additional search operations are to confirm this result. So they propose to interrupt the search before it terminates (visit a limited number of nodes). In this case the probability to find the nearest neighbor is reduced, to avoid this problem Bies et Lowe in (Beis, et al., 1997) and Arye et al (Arya, et al., 1995) propose to modify the process of backtrack search to take into account the position of the query. Hence the search priority is to look in the cells which are close to the query for that they propose to use a priority queue which contains the nodes ranked according to the distance between the query and the node. Then the search process adds the sibling nodes of the current node into the priority queue. These nodes will be considered at a later time in the search. The search is terminated when the algorithm scans the maximum number of nodes or sooner, if the distance from the query point to the node corresponding to the highest priority node is greater than the distance to the closest data point.

Many additional data structures have been developed over the years for solving nearest neighbor problems. (Nister, et al., 2006) defined the “vocabulary tree” which is built using a training set. This set is divided using K-mean clustering with a fixed k throughout the levels of the hierarchical tree. At each node, the subset of features clustered by k-means is split into k nodes and reproduced at the new level. The process of construction is terminated when no further splitting is possible or the

number of levels reaches the maximum value. Silpa-Anan and Hartley in (Silpa-Anan, et al., 2008) create multiple Kd-trees. They choose to split the data from the randomly chosen D dimensions which have the greatest variance. Muja and Lowe in (Muja, et al., 2009)compare these approaches and they introduce a new one of their own (priority search on hierarchical k-means trees). They prove that multiple randomized k-d trees often provide the best performance.

In our method we use the kd-tree to store the descriptors of interest points of all models. We use the algorithm proposed by Beis and Lowe (Beis, et al., 1997) which gives the fast matching and even if the exact nearest neighbor is not obtained, it is likely that some other close neighbors, corresponding to other nearby views of the same model, will contribute to the correct classification (i.e. match hypothesis).

4.2 The proposed algorithm

We propose a method for identifying a person using the matching of interest points found in several images. The main idea of our algorithm lies in the exploitation of image sequences, which can have additional information (3D pose, the dynamic of movement ...) compared with the use of a single image. The information extracted from sequences is subsequently incorporated into a model characterizing the object.

4.2.1 Schema of the algorithm

In this section, we describe the algorithmic choices for the construction of the recognition system. This algorithm follows the classical DRI (Detection, recognition, Identification) algorithms, and can be separated into two phases: a learning phase and a recognition step. The learning step is to detect and track the individual in the sequence to extract points of interest for the construction of the model, using a camera. The recognition stage uses the models from the learning phase to determine if it is the same person in another camera. However, given a query and a set of models, the problem is not to know what points are similar, but to find the most similar model (Ros, et al., 2006). The following sections will show how using a matching procedure to determine the identification of people through images. This is illustrated in figure 4.6 as follows. Being able to tell if two sequences of two individuals are very similar, regardless whether they are similar or different it is then possible to identify the person.

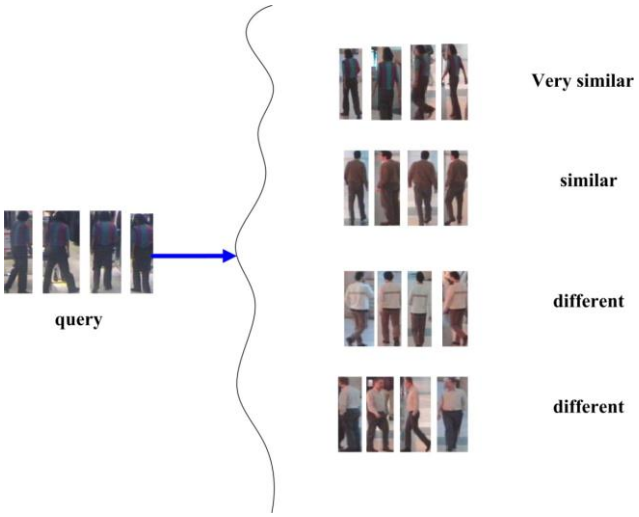


Figure 4-6: Principle of the re-identification of pedestrian

1. Model building

A model is constructed for each individual detected and tracked in the sequence. To increase the amount of information we do not process every successive image frames, but frames are spaced by a half-second. We accumulate in the model the interest points of these different images of the person.

Each model M is defined by a set of vector descriptors calculated at locations where the points of interest have been detected for the images of the model. During the registration phase in the database, each vector descriptor is added to the database with an explicit reference to the number k of the model for which it was calculated. This is illustrated in figure 4.2. After the accumulation of points of interest we use their descriptors to build the KD tree that will allow the matching between model and query descriptors.

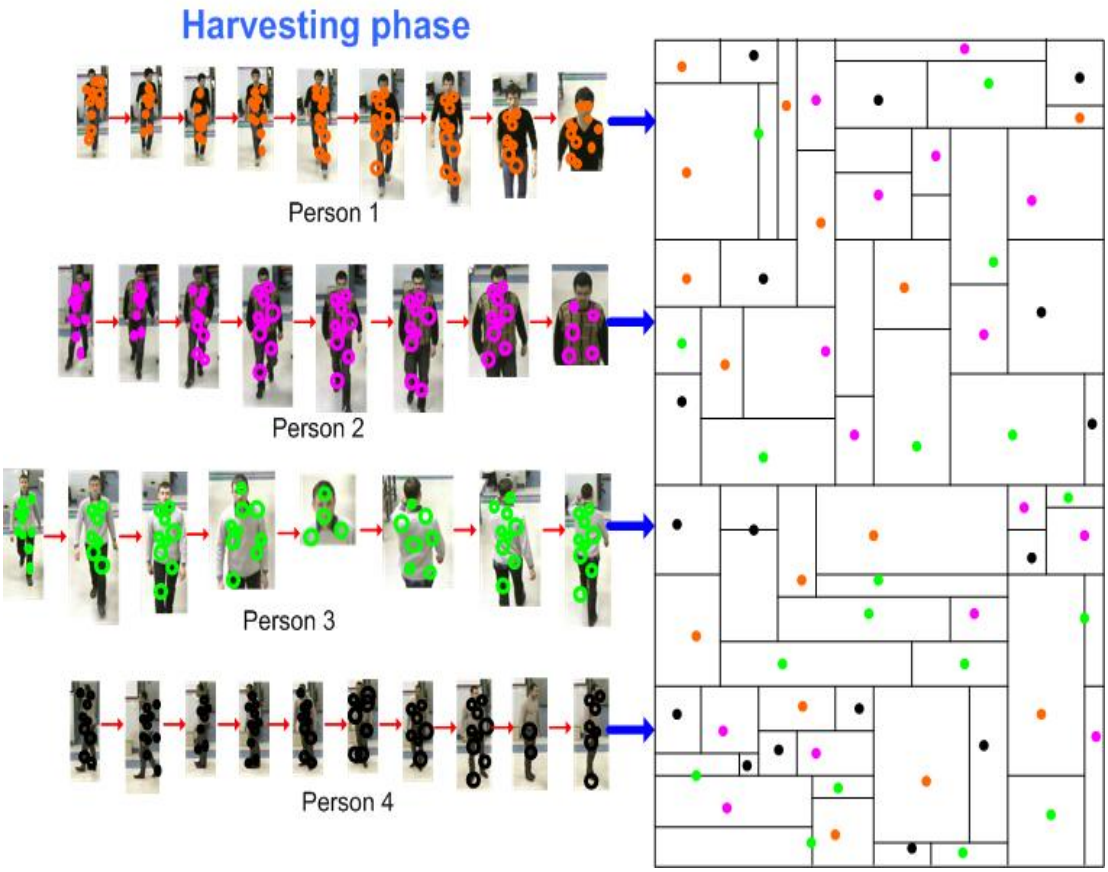


Figure 4-7: Schematic view of model building: for each tracked person. Interest points are collected every 4 frames, and the person's model is the union of all these keypoints stored in a global KD-tree.

2. Query Building:

The query for the target persons is built on several evenly time-spaced images, exactly in the same way as the models, but with a smaller number of images (therefore collected in a shorter time interval).

3. Descriptor comparison

The metric used for measuring the similarity of interest point descriptors is the Sum of Absolute Differences (SAD).

4. Robust fast matching

A robust and very fast matching between descriptors is done by the employed Camellia function, which implements a Best Bin First (BBF) search in a Kd-tree which we discussed above. This Kd-tree contains all the models of the pedestrians.

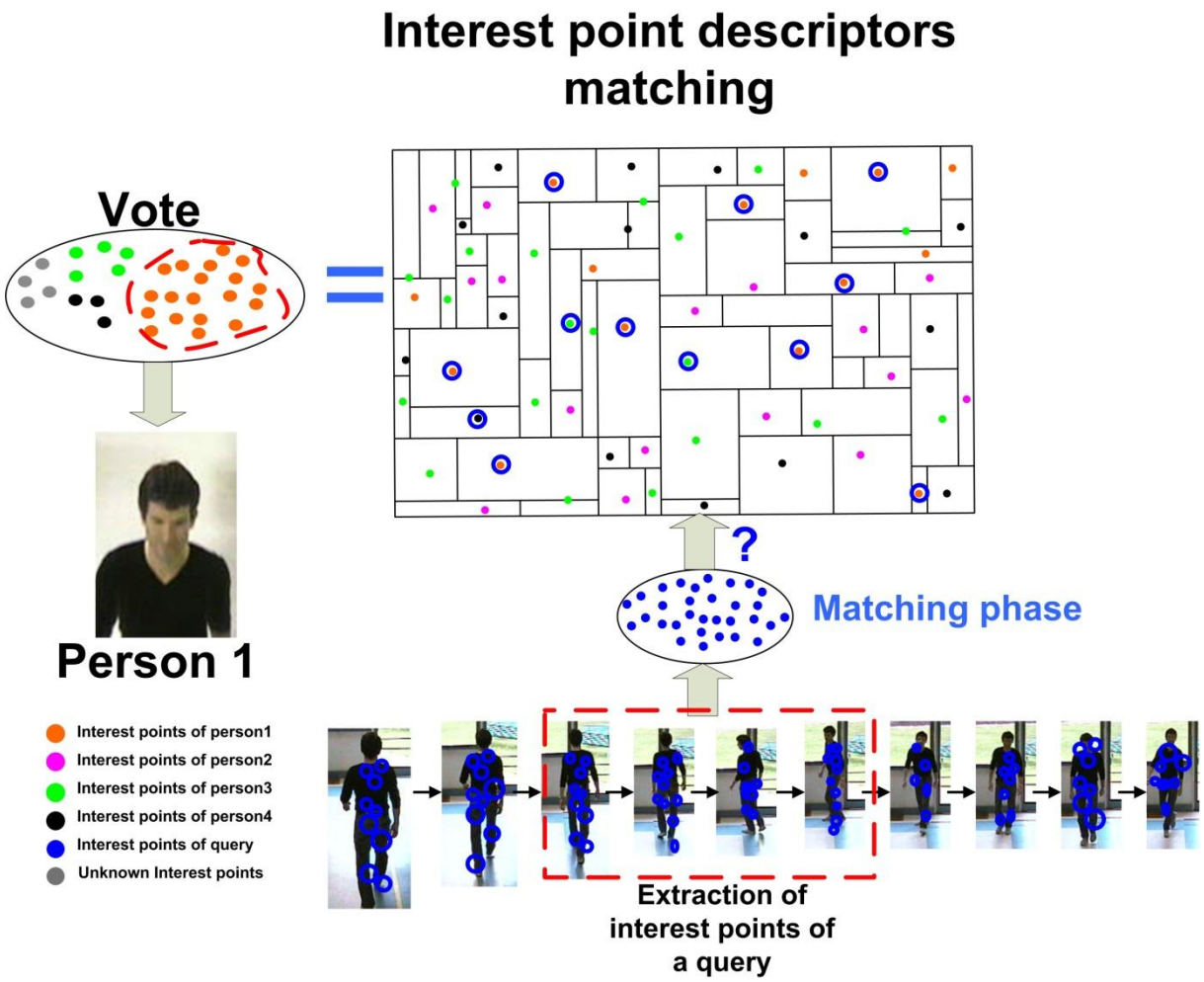


Figure 4-8: Re-identification of a query: for a person to be re-identified appearing in another camera, key-points are collected during a sliding window of 4 seconds (symbolized by red dashed-line red rectangle), and those are matched

5. Identification

The search query in all models is done using the voting technique. Each point extracted from the query is compared to all points of the models stored in a KD tree. The idea of a voting algorithm is to accumulate the number of times the interest points of the query corresponds to a model. Hence each time a model M_k is selected; a table of votes T is updated so that the value T (k) is incremented. Note that a point can select any model, but it can only select a single model. A vote is added to the input of a voting table if the absolute distance between the point of an individual and a point from the tree is below a given threshold (NNDR* second best distance). We use here the nearest neighbor distance ratio (NNDR) matching which we have explained in chapter three. This gives a table of votes where the highest scores correspond to the models most similar to the individual. The model $M_{\hat{k}}$ which has the highest score in the histogram of votes is considered as the best representative of the model:

$$\hat{K} = \max_{k} T(k)$$

Figure 4-8 shows the result of pairings between a sequence of a person to re-identify and a set of 4 models. The model number 1 has been correctly recognized, because of highest vote, even though some votes are also cast on erroneous models.

4.3 Experimental evaluation

4.3.1 Evaluation metrics

With several models and queries, we want a maximum of queries which are correctly matched. The evaluation metric chosen is the number of correct matches over the number of queries (The metric recall):

$$Recall = \frac{TP}{queries\ number}$$

with TP (True Positives) is the number of correct query-model re-identification matching, In addition, we use the precision metrics to evaluate the performance:

$$Precision = \frac{TP}{TP + FP}$$

With FP (False Positives) = number of erroneous query-model matching.

4.3.2 Caviar Dataset

A first experimental evaluation of the proposed method has been conducted on a publicly available series of videos (http://homepages.inf.ed.ac.uk/rbf/CAVIAR) showing persons recorded in corridors of a commercial mall. These videos (collected in the context of the European project CAVIAR [IST 2001 37540]) are of relatively low resolution, and include images of the same 10 persons seen by two cameras with very different viewpoints (see figure 4-9).

Figure 4-9: ***Typical low resolution views of the persons used as part of the models (top-line), and typical views of the same person in the other camera from which we try to re-identify.***

The model for each person was built with 21 evenly time-spaced images (separated by half-second interval), and each query was built with 6 images representing a 3 second video sequence (see figure 4-11). Camera color potential variability is avoided by working in grayscale. Illumination invariance is ensured by histogram equalization of each person's bounding-box.

Table 4-2: precision and recall, as a function of the score threshold for query-model matching

Score threshold for query-model matching (number of matched points)	Precision (%)	Recall (%)
40	99	49
35	97	56
30	95	64
25	90	71
20	85	75
15	82	78
10	80	79
5	80	80

The resulting performance, computed on a total of 760 query video sequences of 3 seconds, is presented in table 4-2, and illustrated on a precision-recall curve in figure 4-10. The main tuning parameter of our method is the "score threshold", which is the minimum number of matched points between query and model required to validate a re-identification. As expected, it can be seen that increasing the matching score threshold, increases the precision but at the same time lowers the recall. Taking into account the relatively low image resolution, our person re-identification performance is good, with for instance 82% precision and 78% recall when the score threshold is set to a minimum of 15 matching interest points between query and model.

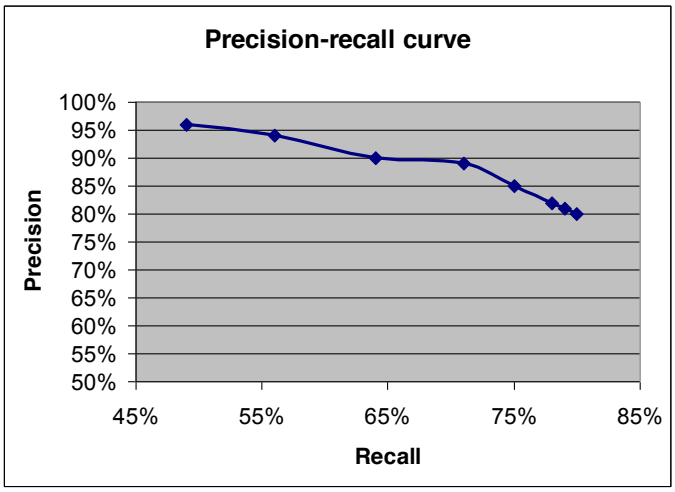


Figure 4-10: ***Precision-recall curve in our first person re-identification experiment.***

Figure 4.11 shows the result of matching a sequence of a person with 23 images (we show the first 12 images) from first camera and a query built with 6 images from the second camera.

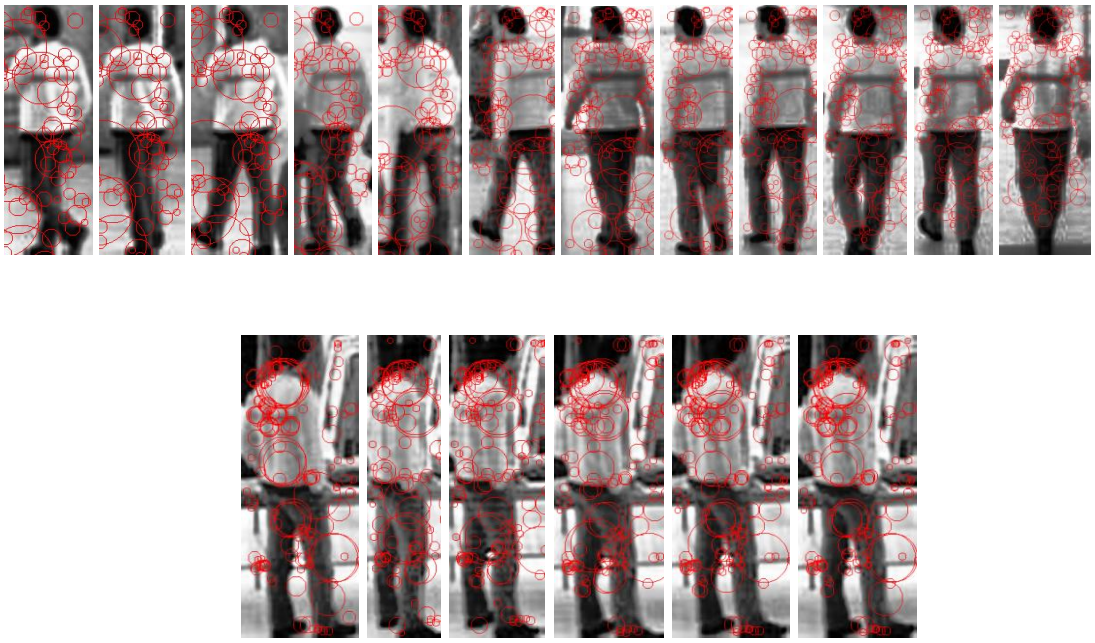


Figure 4-11: Illustration of matched points in the recognition process. The person below has been correctly re-identified in the sequence

Study of the influence of the method parameters

We studied the effect of increasing the number of images constituting a query. The recognition results, calculated on 1100 queries, are presented in Table 4-3 (we set the score threshold to 20 points with 21 image used for each model).

Table 4-3: precision, recall, depending on the number of images constituting a query

Number of images used in query sequences	Precision (%)	Recall (%)
8	83	79
7	83	77
6	85	75
5	87	71
4	92	63
3	96	48
2	99	27
1	99	07

As expected, when the number of images of the query is increased, the recall increases but at the same time, the precision degrades. We also tried to use various numbers of images per model of person, and studied the impact on obtained precision and recall. As can be seen on figure 4-12 and the table 4-3 there is a very clear and significant simultaneous improvement of both precision and recall when the number of images per model is increased. This validates the interest of the concept of harvesting interest points for the same person on various images. The recognition results, calculated on 1100 queries are presented in table 4-4. We set the threshold score to 20 points with 6 images with each sequence-query.

Table 4-4: precision, recall, depending on the number of images constituting a model

The number of images extracted from each sequence-model	The total number of points of interest	Precision (%)	Recall (%)	Computation time for matching (ms)
1	1117	55	52	123
2	2315	60	55	132
4	5154	65	57	141
8	11100	66	57	149
16	22419	74	65	157
24	32854	80	72	161

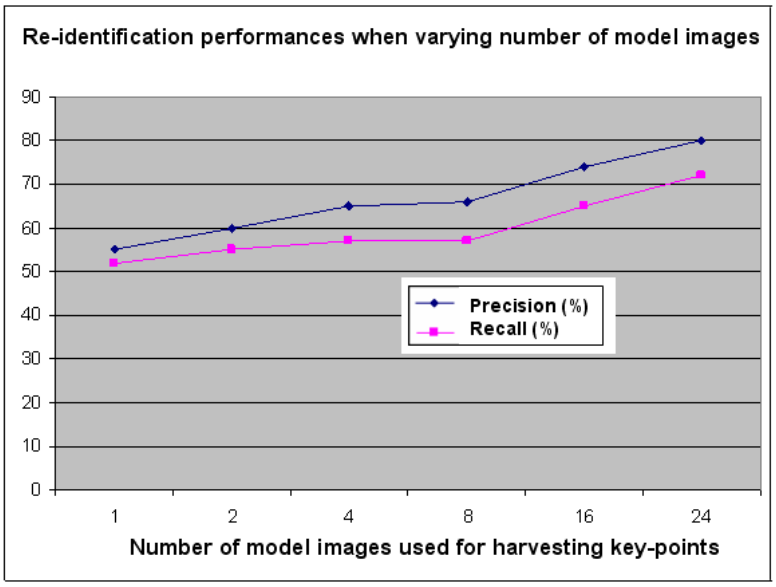


Figure 4-12: Influence of the number of images used per model on the re-identification precision and recall

The high execution speed of our re-identification method should also be emphasized. The computing time is less than 1/8 s per query, which is negligible compared to the 3 seconds necessary to collect the six images separated by ½ s. More importantly, due to the logarithmic complexity of the KD-tree search with respect to the number of stored descriptors, the query processing time should remain very low even when very large numbers of person models are stored. In order to verify this, we compared the re-identification computation time when varying the number of images used in model sequences, as reported in table 4-4. Indeed, figure 4-13 shows that the re-identification computation time scales logarithmically with the number of stored descriptors. Since the number of stored descriptors is roughly proportional to the number of images used, if 1000 to 10000 person models were stored instead of 10 (with ~ 20 images for each), the KD-tree would contain 100 to 1000 times more key-points, i.e. ~ 2.5 to 25 millions of descriptors. Extrapolating from figure 4-13, we therefore expect a query computation time ~ 1/5 to ¼ s for a re-identification query among thousands of registered person models.

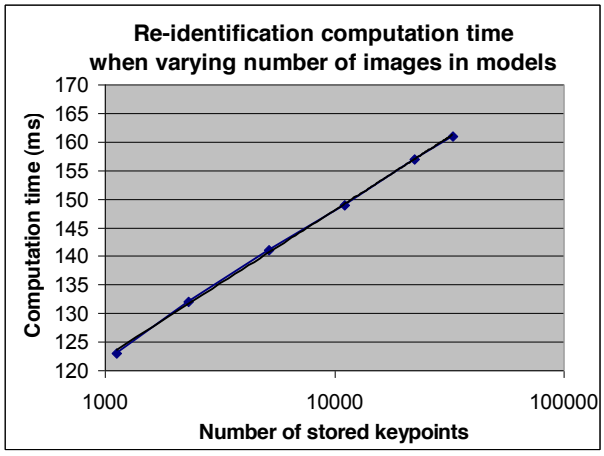


Figure 4-13: ***Re-identification computation time as a function of stored keypoint descriptors; the dependence is clearly logarithmic***

We have also studied the influence of the threshold used in the Camellia Keypoint detector. The recognition results, calculated on 1100 queries, are presented in Table 4-5, we set the score threshold to 20 points with 6 images with each query sequence, and 23 small images each constituent model.

Table 4-5: precision, recall, and computation time, depending on the threshold of the Hessian determinant

Threshold of detector	The total number of points of interest in model	The total number of points of interest in query	Precision (%)	Recall (%)
100	43632	224116	78	74
150	35204	182420	82	76
200	29385	153756	82	71
250	25084	132688	88	72
300	21832	116382	88	66
350	19199	103663	89	62
400	17135	93177	93	60
450	15414	84357	96	53
500	14016	77045	95	49

Increasing the threshold reduces the number of interest points (Table 4-5), which has the effect of reducing the recall (especially beyond 200-250) because we have a poorer description of the models. On the other hand the precision increases, probably by decreasing “noisy” interest points in the background (behind the people).

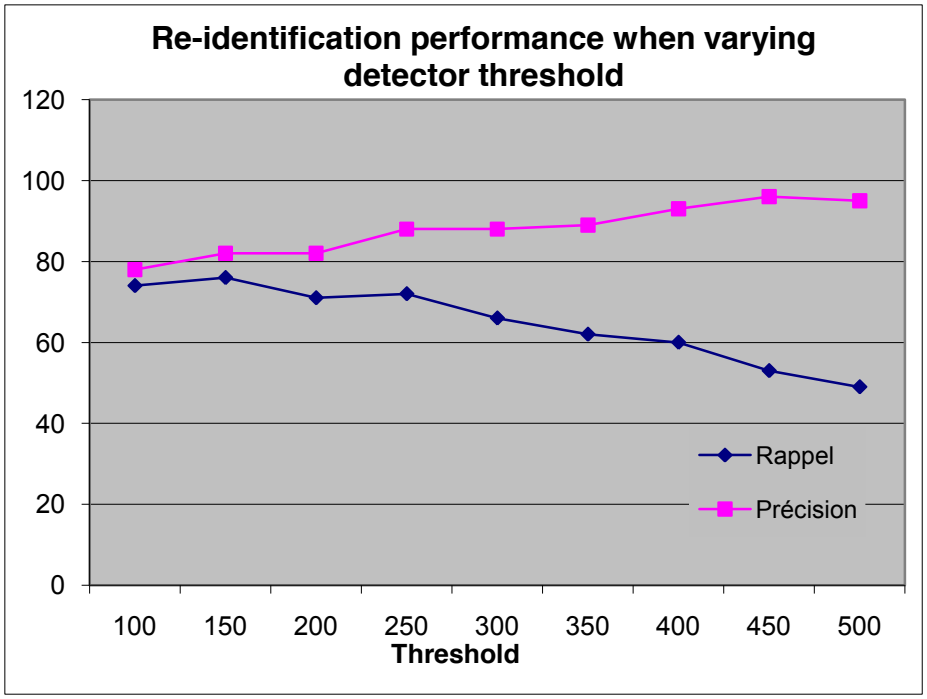
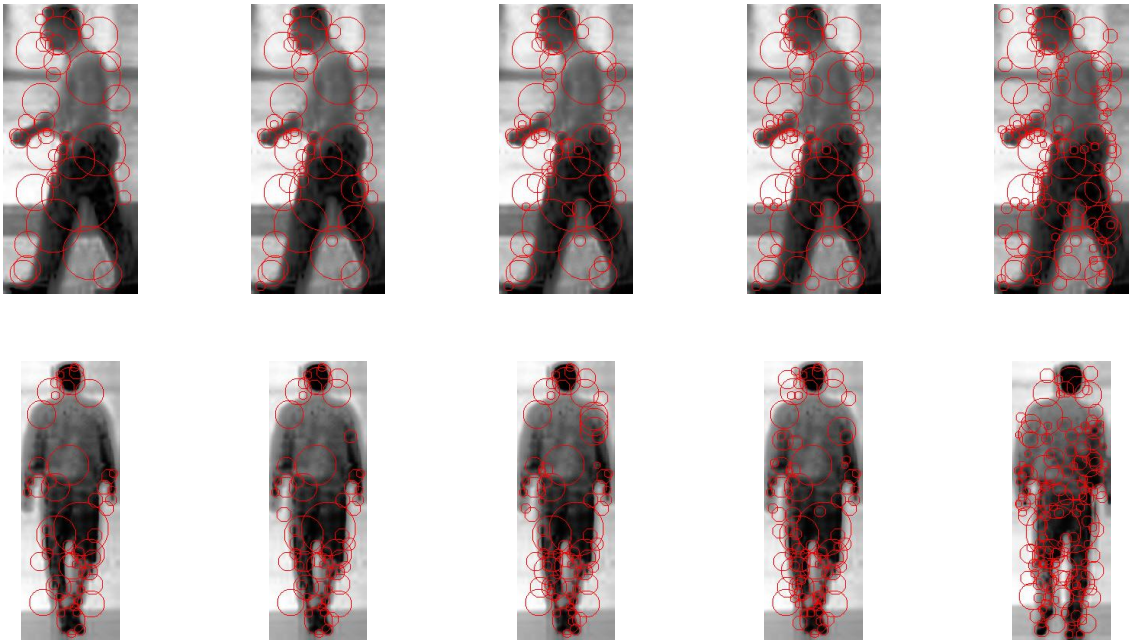


Figure 4-14: Influence of keypoint detector threshold on re-identification performance.

Figure 4-15 shows the effect of the change of threshold on the number of interest points that build a model. This figure shows that the number of points of interest generally increases when the threshold is reduced, particularly at the edge silhouettes.



Thresh=500 Thresh =400 Thresh =300 Thresh =200 Thresh =100

Figure 4-15: The interest points detected on two people for different threshold values

4.3.3 Our newly built corpus with 40 persons

Table 4-6 summarizes the datasets used in re-identification literature. However most of these involve low-resolution video or few frames per person, and only three are freely available (Viper, iLIDS MCTS, and (Schwartz, et al., 2009)).

The VIPER dataset (Gray, et al., 2007) contains 632 pedestrian image pairs from arbitrary viewpoints. Each image was cropped and scaled to be $128x48$pixels. The VIPER dataset has a relatively low resolution which rewards techniques that focus on global features like color histograms rather than local features.

The iLIDS MCTS dataset is a publicly available dataset captured at an airport arrival hall. Zheng et al extracted 479 images of 119 pedestrians. The disadvantage of this data set is the small number of images for each pedestrian. We must distinguish here between two re-identification approaches: the first approach is based on the use of one is single image to represent the query or the model which is more difficult than the other approach which use multiple images to define the query and model, our method works using multiple images.

The ETHZ dataset is a publicly available series of videos (http://www.vision.ee.ethz.ch/~aess/dataset/) showing persons recorded using moving cameras, intended for pedestrian detection. Schwartz and Davis in (Schwartz, et al., 2009) extract a set of samples for each person in the videos to test their PLS method. The most challenging aspects of ETHZ are the illumination changes and occlusions and the large number of pedestrians. There are 146 pedestrians amongst the three sequences: SEQ. #1 contains 83 pedestrians (4,857 images), SEQ. #2 contains 35 pedestrians (1,936 images) and SEQ. #3 contains 28 pedestrian (1,762 images).

Table 4-6: Overview of publicly available datasets for monitoring and multi-camera re-identification

system	dataset
VIP : Vision tool for comparing Images of People (Lantagne, et al., 2003)	16 people using 3 cameras
Full-body person recognition system (Nakajima, et al., 2003)	8 person using 1 camera
Person Re-identification Using Spatiotemporal Appearance (Gheissari, et al., 2006)	44 people using 3 cameras
MONNET: Monitoring Pedestrians with a Network of Loosely-Coupled Cameras (Albu, et al., 2006)	200 using 4 cameras
Evaluating Appearance Models for Recognition, Reacquisition, and Tracking (Gray, et al., 2007)	632 using two cameras
Video sequences association for people re-identification across multiple non-overlapping cameras (Cong, et al., 2009)	35 persons using two cameras
Learning discriminative appearance based models using partial least squares (Schwartz, et al., 2009) (ETHZ)	Using moving camera: three data sets: 1. 83 persons 2. 35 persons 3. 28 persons
iLIDS Dataset	479 images of 119 persons
Cascade of descriptors to detect and track objects across any network of cameras (Alahi, et al., 2010)	Use two cameras

Since none of the available datasets has the necessary characteristics for a deeper evaluation of our method, we contribute a high resolution video re-identification data set. The scenes were shot from two cameras as we described in Chapter 3. Videos were taken with a 352 x288 pixels resolution. The frame rate varied between 12 and 25 frames per second. The mean duration of each pedestrian sequence exceeds 200 frames. Video sequences lasted between 20 and 30 seconds, ending when the person passes in front of the camera twice. Our evaluation consists of using a pedestrian from one camera as the model and the same pedestrian from another camera as the query. We also evaluate the algorithm when the number of pedestrians increases using ETHZ dataset.

Most proposed methods in re-identification use the Cumulative Matching Characteristic curve (CMC) which represents the expectation of finding the correct matching in the best n matches. The CMC is shown in figure 4-16 (for number of images in query N=10). When the number of images of models increases the recognition percentage improves. So the simple solution would be to use all frames in the tracks and an all-to- all matching framework. This is because the increasing of images augments the distinctive information of each model. But this improvement is limited by the number of images which define the model and the capacity of storing the interest points. This increase also causes the increasing of the redundant information which prevents the matching using the ratio (NNDR). In figure 4-17 we switch between the query and model we note there is a small improvement when we use the image of camera 1 as model.

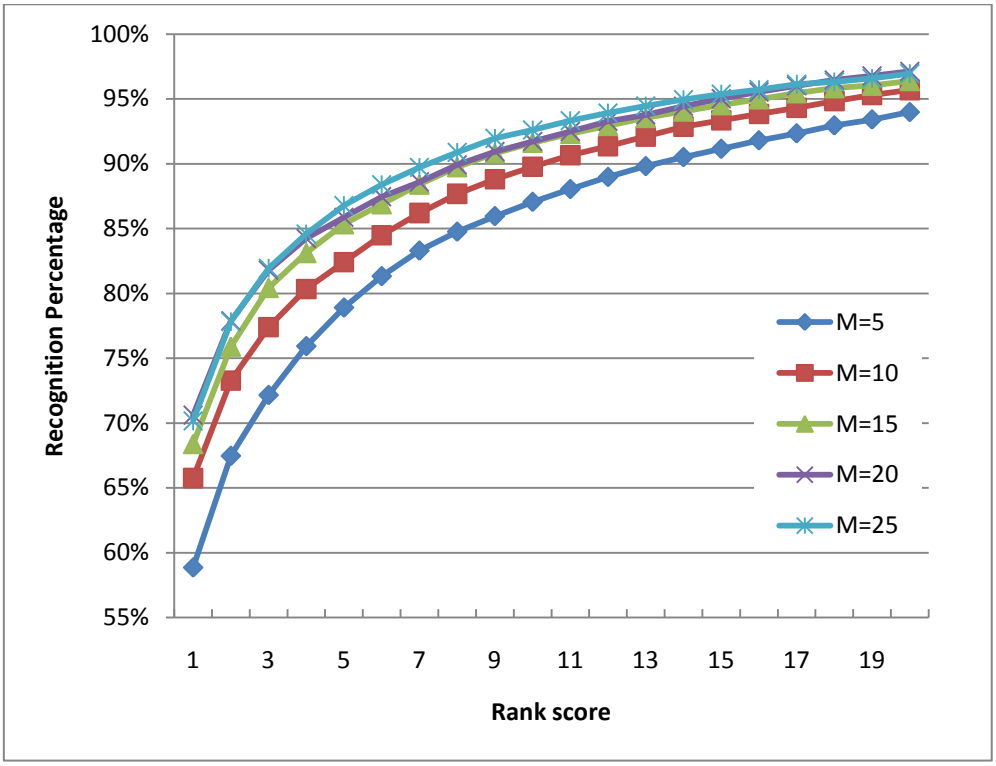


Figure 4-16: CMC for different number of images used per model
(query from camera 2 matched with model from camera 1, and number of images in query is N=10)

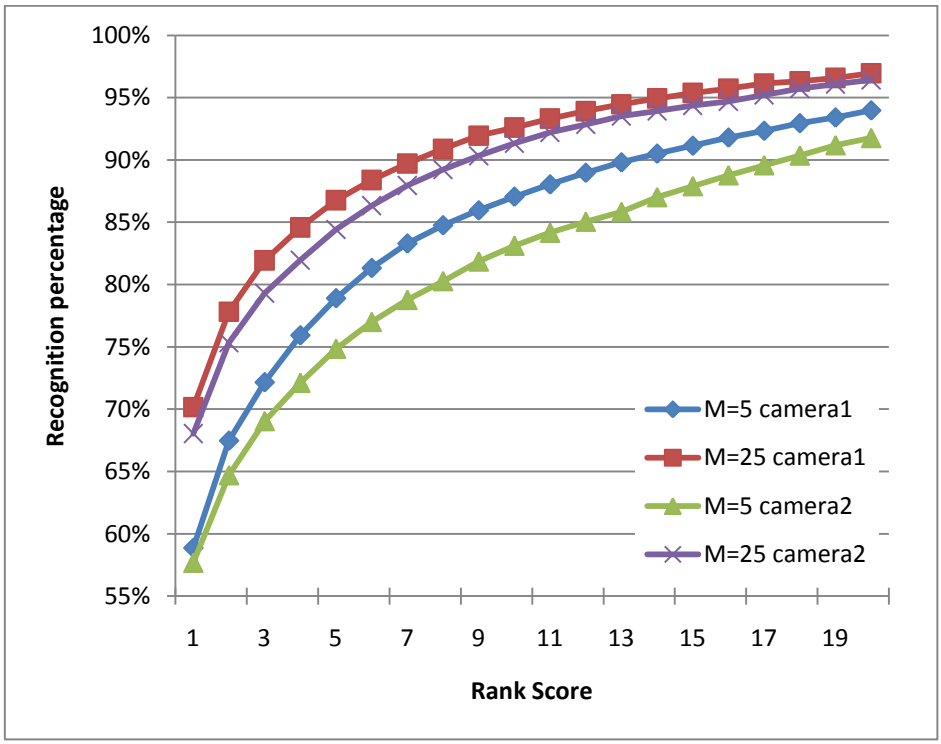


Figure 4-17: comparison when we switch between the query and model (number of images in query is N=10)

Increasing the number of query images also augments the performance, as we can see in figure 4-18. In our method we do not have any limit to increase this number because we use a sliding window. We can see in figure 4-19 the influence of switching between the model and query and we can see the performance in case we use the image of the first camera as a model outperforms slightly the second case.

A deeper analysis look at change in histogram of votes (see in figure 4-20): We can see that using large numbers of images to construct the query increases the margin between the bins of histograms. The red bar represents the bin of correct match; the histogram is normalized.

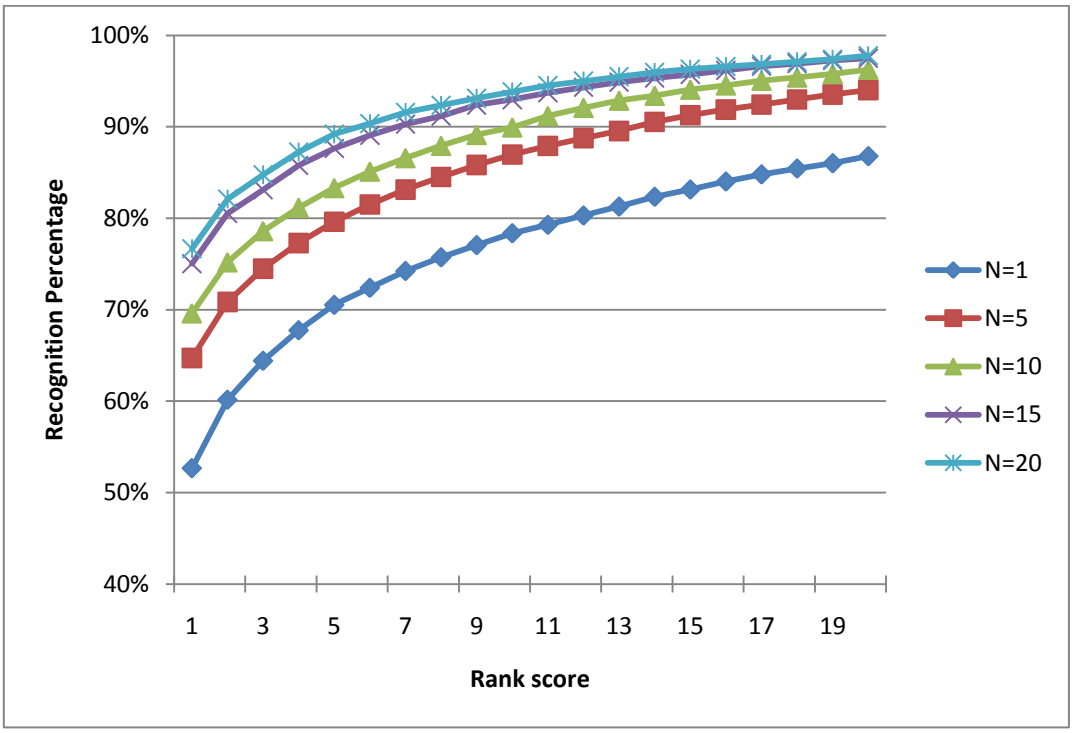


Figure 4-18: Influence on CMC of the number N of images constituting a query (query from camera 2 matched with model from camera 1, and number of images in model M=20)

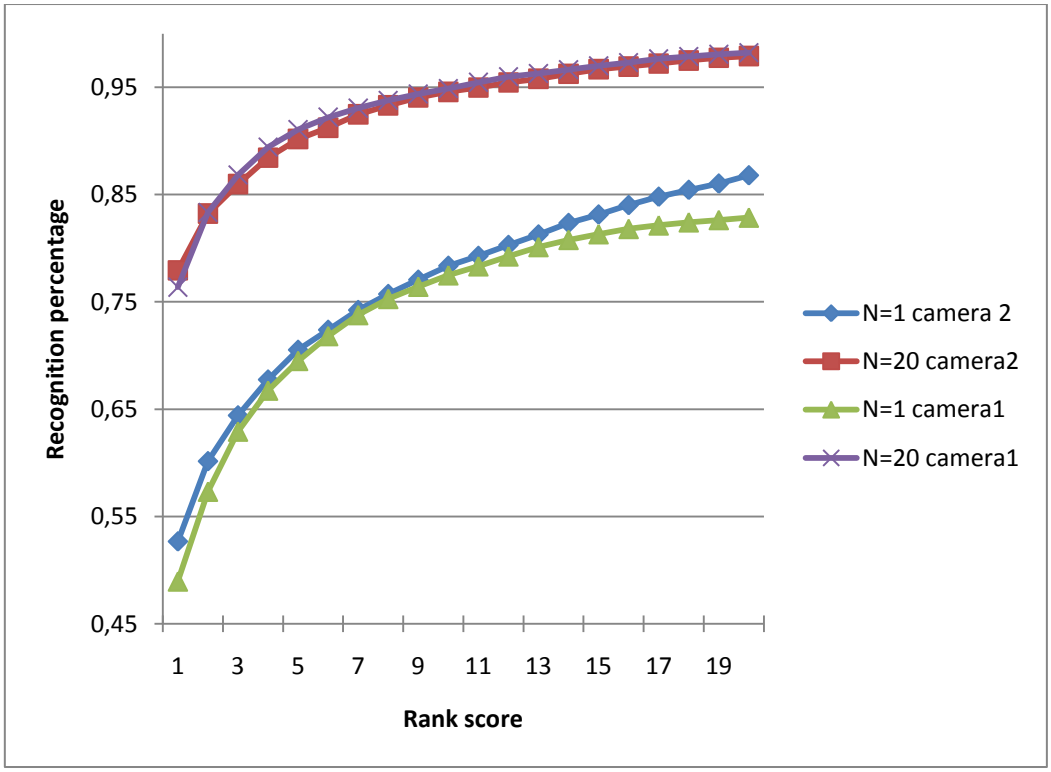
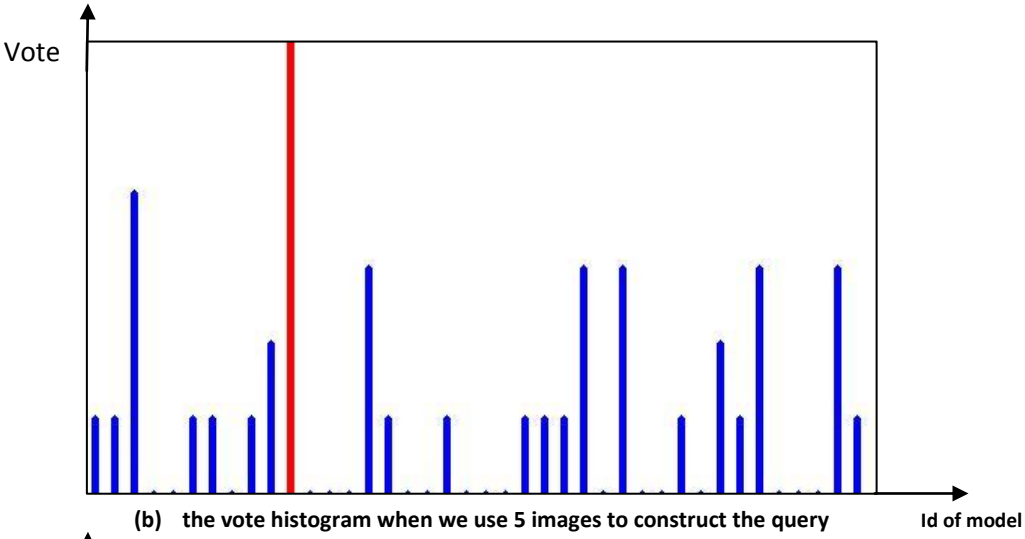
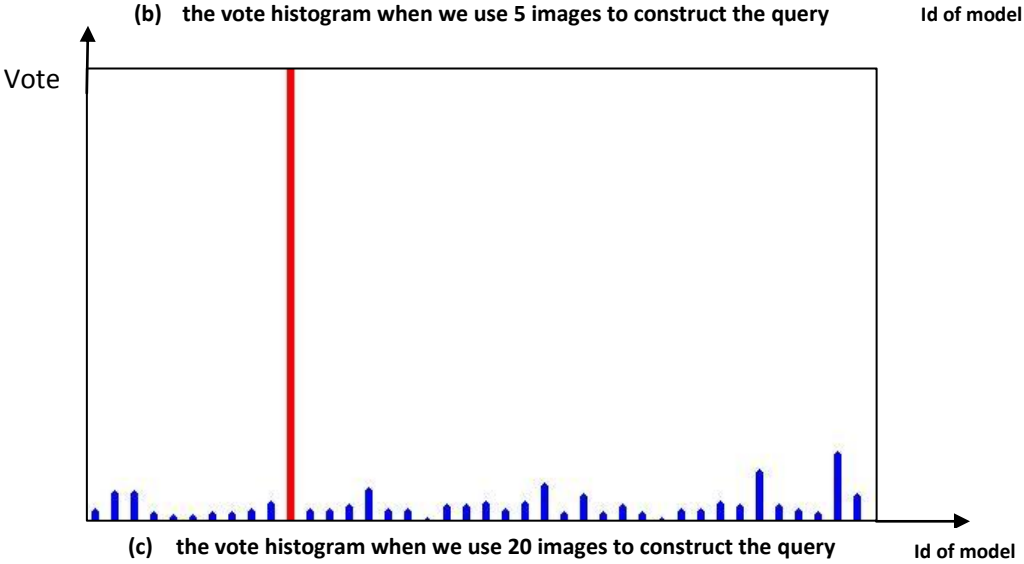


Figure 4-19: Comparing query from camera 2 on model from camera 1 with inverse roles (number of images in model is M=20)

(a) Original image



(b) the vote histogram when we use 5 images to construct the query



(c) the vote histogram when we use 20 images to construct the query

Figure 4-20: the vote histogram when we use different values of images to construct the query. We can see that using large numbers of images to construct the query increases the margin between the bins of histograms. The red bar represents the bin of correct match; the histogram is normalized.

To study the performance when the images of different size are matched, we scaled each dimension of the images of camera 1 to $\frac{1}{2}$,$\frac{1}{4}$ of the original size, and then matched them with images of camera 2. This comparison is done with other parameters giving best performance (number of images in model is M=20, and number of images in query is N=20). Table 4-7 and Table 4-8 show that only when the size is scaled to $\frac{1}{4}$, the recall rate drops significantly. On the other hand, the precision is slightly affected.

Table 4-7: the performance of re-identification when images of query are scaled

	Precision (%)	Recall (%)
Original scale	84.5	76.5
Image of query scaled by 1/2	78.1	70.8
Image of query scaled by 1/4	75.3	48

Table 4-8: Performance of re-identification when images of model are scaled

	Precision (%)	Recall (%)
Original scale	84.5	76.5
Image of model scaled by 1/2	70.4	67.2
Image of model scaled by 1/4	68.1	46.2

4.3.4 Individual recognition analysis

To further analyze the effect of varying the number of model images and query images we calculated the precision and recall separately for each pedestrian instead of averages. Figure 4-21 presents the precision for each pedestrian when we change the number of images. We note that the precision improves globally for each pedestrian. Similarly, when we increase the number of query images, we observe a clear improvement of the precision as shown in figure 4-22.The number of model images has a larger effect on the precision when selecting a higher number of query images. It can be noted that when the pedestrian's clothes are rich in texture, the precision and the recall increase. Conversely, the precision and the recall are lower when there is a lack of texture (see figure 4-23).

Id	1	2	3	4	5	6	7	8	9	10
M=5	79	65	43	85	81	44	58	83	24	44
M=10	98	65	63	86	78	46	86	94	52	51
M=15	95	97	66	94	88	56	75	89	56	54
M=20	91	89	76	95	88	63	83	98	50	71
M=25	95	95	80	100	88	75	89	96	50	51
Id	11	12	13	14	15	16	17	18	19	20
M=5	64	87	100	50	93	72	90	55	84	59
M=10	75	84	100	61	88	79	96	69	86	35
M=15	73	91	100	74	96	73	91	81	96	77
M=20	85	96	100	72	95	83	95	80	96	81
M=25	86	98	100	83	100	85	99	87	99	90
Id	21	22	23	24	25	26	27	28	29	30
M=5	78	74	82	28	35	31	2	31	83	31
M=10	76	85	82	29	30	36	23	32	73	48
M=15	78	87	86	49	69	42	42	61	95	11
M=20	90	88	87	24	48	47	31	70	97	44
M=25	91	89	91	33	60	66	40	80	100	30
Id	31	32	33	34	35	36	37	38	39	40
M=5	26	24	42	99	78	20	32	69	54	74
M=10	32	26	62	99	84	22	15	72	74	74
M=15	73	36	73	100	88	40	17	81	72	74
M=20	79	43	68	100	89	39	20	83	83	84
M=25	81	50	81	100	92	49	26	95	84	85

Figure 4-21: the precision for each pedestrian for different values of images in the model. We note that the precision improved globally when increasing the number of model images used as shown (number of images in query is N=10)

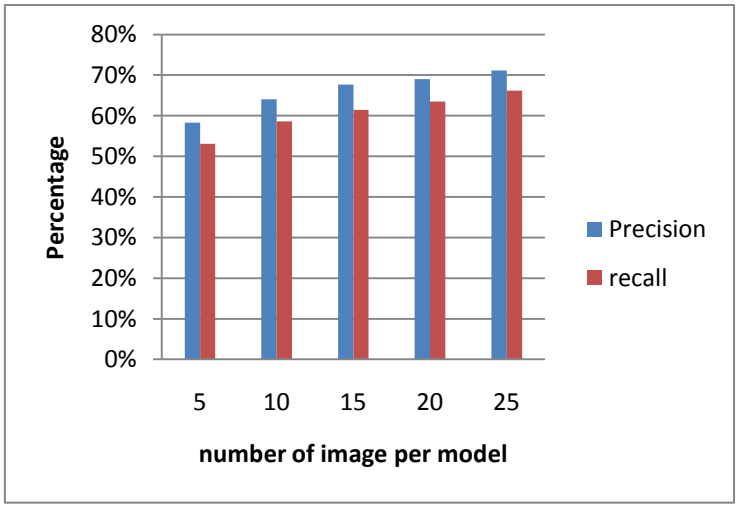


Figure 4-22: precision and recall with respect to the number of images per model (number of images in query is N=10)

Id	1	2	3	4	5	6	7	8	9	10
N=5	100	96	82	88	94	74	90	99	72	84
N=10	100	89	80	89	88	68	92	100	61	78
N=15	100	92	83	94	88	76	94	100	63	66
N=20	100	93	87	100	87	84	100	100	66	62
N=25	100	91	89	100	88	94	100	100	67	65
Id	11	12	13	14	15	16	17	18	19	20
N=5	93	90	99	67	98	82	99	67	87	56
N=10	90	93	100	74	100	78	97	71	96	41
N=15	93	95	100	79	100	83	97	78	99	49
N=20	93	99	100	84	100	86	99	85	100	60
N=25	93	100	100	87	100	88	99	92	100	71
Id	21	22	23	24	25	26	27	28	29	30
N=5	82	99	99	60	36	54	27	43	91	35
N=10	90	89	92	49	43	58	31	45	81	58
N=15	95	93	93	49	46	56	33	57	86	55
N=20	100	99	97	58	48	60	34	62	90	67
N=25	100	100	99	60	55	77	42	72	94	70
Id	31	32	33	34	35	36	37	38	39	40
N=5	43	51	86	100	99	30	22	76	92	77
N=10	36	49	74	100	94	23	24	75	73	68
N=15	36	52	72	99	91	24	25	81	73	70
N=20	40	60	77	99	91	25	30	85	73	74
N=25	45	62	82	99	94	30	33	88	74	74

Figure 4-23: the precision for each pedestrian for different values of images of query. We note that the precision improved globally when increasing the number of model images used as shown

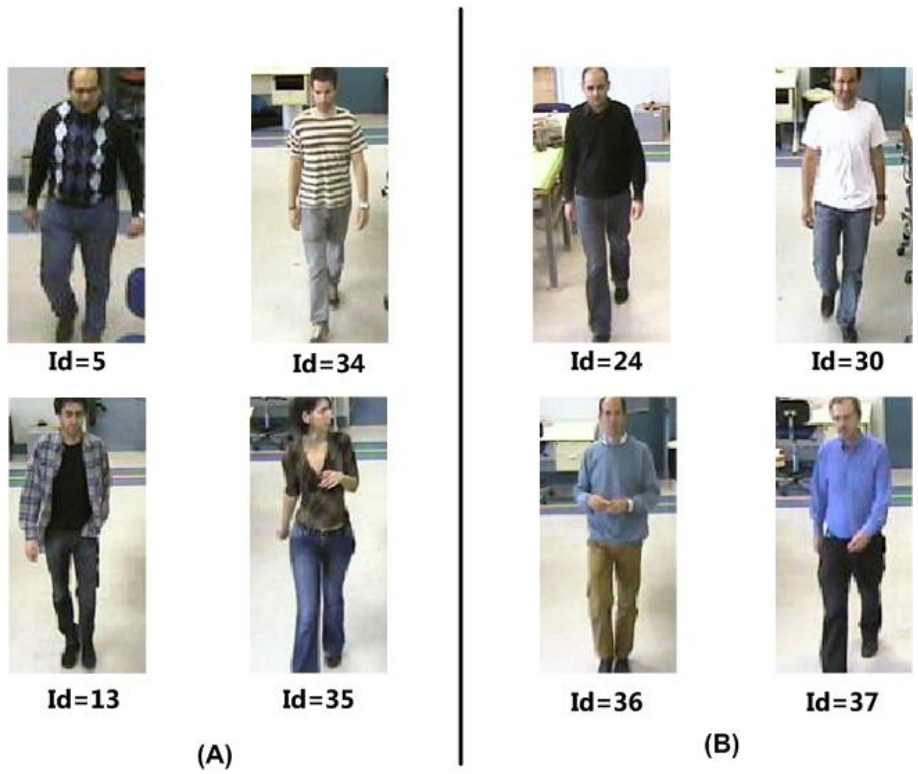


Figure 4-24: examples of cloth texture: richness of texture on (A), and poorness of texture on (B)

Figure 4-25 presents the matched interest points found across the cameras for a pedestrian with poor texture and that of one with rich texture. We note that the richness of textures increases the number of matched interest points. Some pedestrians have clothes with smooth regions which made very few interest points leading to an unfeasible matching process. Table 4-9 represent the number of interest points used to construct the model using 15 images for the pedestrian. In figure 4-24 we can note that the number of points increases when the clothes texture is more important.

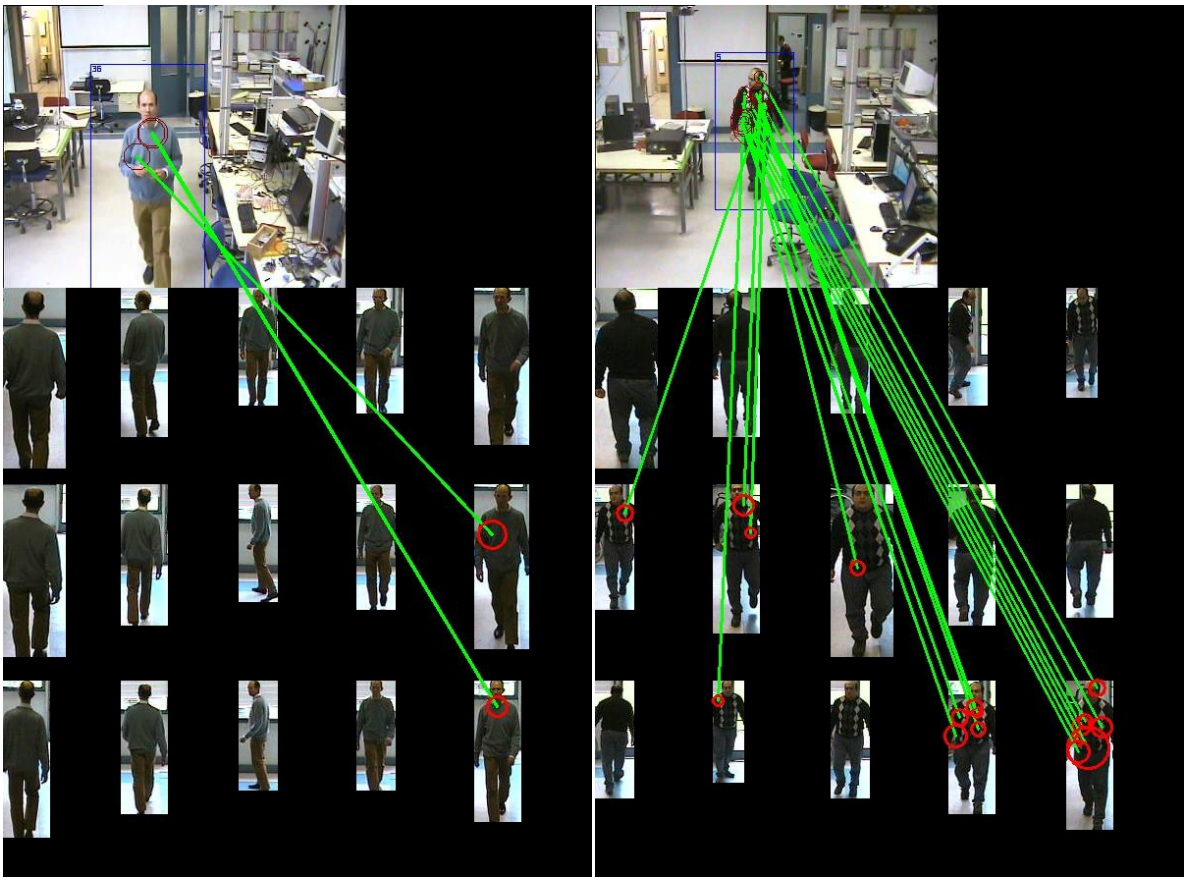


Figure 4-25: the matching between a query and the ensemble of models, we note that the richness of texture augments clearly the number of matched interest points.

Table 4-9: Number of interest points which construct the model for the pedestrians in figure 4-24

Richness of Textures		Poorness of textures	
ID=5	ID=34	ID=24	ID=30
1198	1027	812	972
ID=13	ID=35	ID=36	ID=37
1365	1169	835	845

Another evaluation of our algorithm was performed. We tested the algorithm using the first twenty pedestrians. We calculated the confusion matrix with respect to the number of images which constructs the query. We can note that the value in the diagonal of the matrix is higher when the

number of queries increases (i.e. the confusion is decreased). We note in the third matrix that there is confusion between the sixth individual and the eighteenth, seventh individual and the twentieth. This can be explained by the fact that some interest points are not distinctive as we can see in figure 4-29. So these points cause ambiguity. In the next section, we propose a method to decrease the influence of these points by giving the points the vote value depending on the information present in these points.

id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	86,02	2,15	1,08	2,15	1,08	1,08	0	0	0	0	2,15	0	2,15	0	0	0	2,15	0	0	0
2	2,54	81,36	2,97	1,27	0	0,85	0,85	0	0	1,27	0	2,12	0,85	0,85	0,85	0	2,54	1,27	0	0,42
3	1,49	0,75	80,22	2,99	0	1,87	0,75	0,37	0,75	1,12	0,75	0	1,49	0	0,75	0	3,73	2,99	0	0
4	1,03	0	3,08	85,64	0,51	0,51	0	0	0,51	1,03	1,03	0	0	0	0	0,51	1,03	5,13	0	0
5	0,48	0,96	0,48	1,91	91,87	0	0	0	0	1,91	0	0,48	0	0	1,91	0	0	0	0	0
6	0	1,1	2,76	4,42	1,1	39,23	0,55	3,31	0	1,66	0,55	0,55	3,31	2,21	1,1	4,97	2,21	30,94	0	0
7	0	0	0,47	0,47	0	0	81,69	1,41	2,82	0	0	1,88	2,82	1,88	0	0,94	0,94	0	0,94	3,76
8	1,79	0	0,89	3,57	0	4,02	0	69,64	0	0	0	1,79	2,23	2,23	0,45	2,23	1,34	8,48	1,34	0
9	3,23	0	1,94	1,29	0	0	12,9	4,52	48,39	1,29	0	7,74	7,1	1,94	1,29	2,58	1,29	0,65	0	3,87
10	1,07	1,6	4,81	8,02	4,81	0,53	0	0,53	0,53	62,03	2,14	0,53	2,14	1,07	3,74	0	1,6	3,21	1,07	0,53
11	7,6	0	0	0,58	3,51	0	0	0	0	1,75	83,04	0,58	1,17	0	0,58	0	0	0,58	0,58	0
12	4,97	1,24	0	0,62	0,62	1,24	0	0	1,24	0	0	83,85	0,62	0,62	0,62	2,48	0	1,86	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
14	3,83	2,87	0,48	4,78	0,48	0,48	0,96	2,87	0,48	1,91	1,44	8,13	0,96	58,85	2,39	1,91	4,31	2,87	0	0
15	1,7	3,41	3,41	0,57	1,7	2,84	0	0,57	0	0,57	1,7	3,98	0	1,14	75,57	0	0	2,84	0	0
16	0	0,67	0	0	0,33	0,67	1,33	1	0	0,33	0,33	5,33	4	0,33	0,33	84	0	1,33	0	0
17	1,4	0	1,4	0,7	1,4	1,4	0,7	0	0	1,4	0	0	0,7	0,7	0,7	0	89,51	0	0	0
18	0,65	0	1,31	0	0	2,61	0	0,65	0	3,92	0	1,31	3,27	0	0	0,65	5,88	79,74	0	0
19	0	0	3,09	1,23	0	2,47	0,62	1,85	2,47	1,23	0	0	0,62	0	0	1,85	1,23	24,69	58,64	0
20	2,08	0	1,39	3,47	0	1,39	21,53	3,47	11,11	1,39	0	4,17	6,94	5,56	0	2,78	1,39	2,78	1,39	29,17

Figure 4-26: the confusion matrix of the first twenty pedestrians with 1 image to construct the query

id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	94,38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4,49	1,12	0	0
2	2,59	91,38	0	0	0	0	0	0	0	0	0	2,16	0	1,72	1,72	0	0	0	0,43	0
3	0	0	91,67	1,52	0	0,38	1,14	0	0	0	0,38	0	0,38	0	0	0	4,55	0	0	0
4	1,05	0	2,62	93,72	0	0	0	0	0	0,52	0	0	0	0	0	0	0	2,09	0	0
5	0,49	0	0	0	99,02	0	0	0	0	0	0	0,49	0	0	0	0	0	0	0	0
6	0	0,56	1,69	1,13	0	44,07	0	0	0	0	0	0	3,39	0,56	0	0	0,56	48,02	0	0
7	0	0	0	0	0	0	91,39	0	1,91	0	0	0	1,91	1,44	0	0	0	0	0	3,35
8	0,45	0,45	0	2,27	0	1,82	0	78,18	0	0	0	0	1,36	1,82	0	0	0,45	12,73	0,45	0
9	0	0	0,66	0	0	0	9,93	0	64,9	0,66	0	3,97	7,28	0,66	0	0	3,31	0	1,99	6,62
10	0	0	0	7,1	0	0	0	0	0	83,06	3,83	0	0	0	2,73	0	1,64	0,55	1,09	0
11	0,6	0	0	0	0	0	0	0	0	0	99,4	0	0	0	0	0	0	0	0	0
12	4,46	1,27	0	0	0	0	0	0	0	0	0	90,45	1,27	0	0,64	1,91	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
14	0,98	1,46	0	1,95	0	0	0	0,49	0	0,49	5,85	3,41	0	76,59	2,44	0	2,93	3,41	0	0
15	0	3,49	1,16	0	0	0	0	0	0	0	1,74	0	0	0	93,02	0	0	0,58	0	0
16	0	0	0	0	0	1,01	1,69	0	0	0	0,68	0,68	0	0	0	95,95	0	0	0	0
17	0	0	0	0	0	0	1,44	0	0	0	0	0	0	0	0	0	98,56	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	2,68	0	0	0	7,38	89,93	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24,05	75,95	0
20	0	0	0	0	0	0	29,29	0	3,57	0	0	0	0,71	2,86	0	0	0	0	2,86	60,71

Figure 4-27: the confusion matrix of the first twenty pedestrians with 5 images to construct the query

id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	70,95	0	0	0	0	0	0	0	0	0	0	0	29,05	0	0
7	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	89,5	0	0	0	0	0	0	0	0	0	10,5	0	0
9	0	0	0	0	0	0	18,32	0	80,15	0	0	1,53	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21,74	78,26	0
20	0	0	0	0	0	0	17,5	0	0	0	0	0	0	0	0	0	0	0	0	82,5

Figure 4-28: the confusion matrix of the first twenty pedestrians with 20 images to construct the query

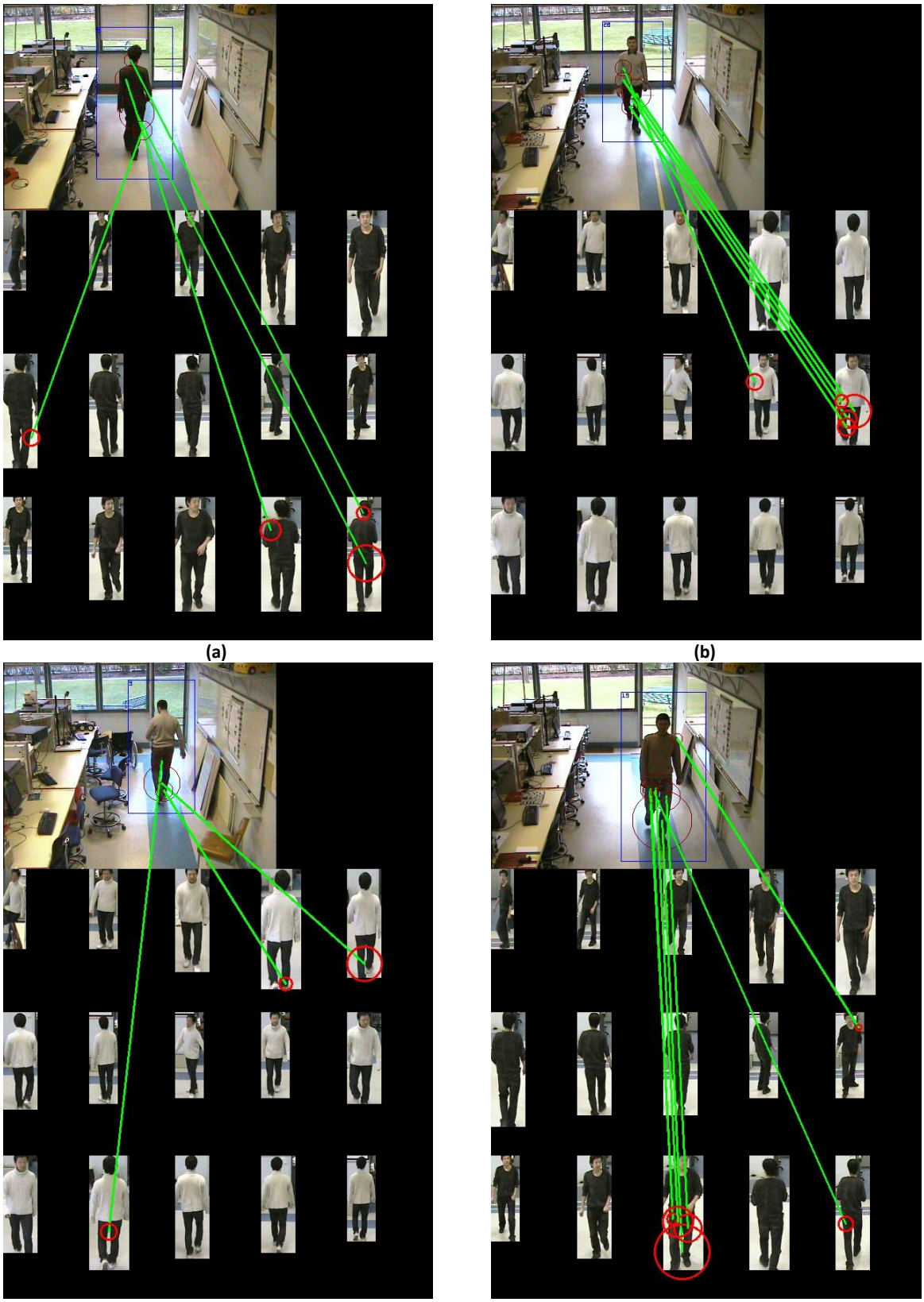


Figure 4-29: example of ambiguity when matching, (a) ambiguity between pedestrian6 &pedstrian18, (b) pedestrian7 &pedstrian20, (c) pedestrian8 & pedestrian7, (d) pedestrian19 & pedestrian18. We note that the most confusion in where there is a lack texture (information)

4.3.5 NNDR ratio influance

As seen in the previous section, a classical approach is able to achieve a satisfactory level of precision in a lot of situations. However, this level of performances is still limited by the tuning of parameters. There are many parameters which have an effect on the precision. The main one is arguably the fact that re-identification depends on the quantity of information and the quality of this information. Using several frames to construct the model augments the quantity of information; unfortunately, in some situation this augmentation does not provide the improvement of the quality. On the contrary the increasing of number of model images can cause the redundancy of information, and decreases the performance. In this case the influence of ratio NNDR amplifies.

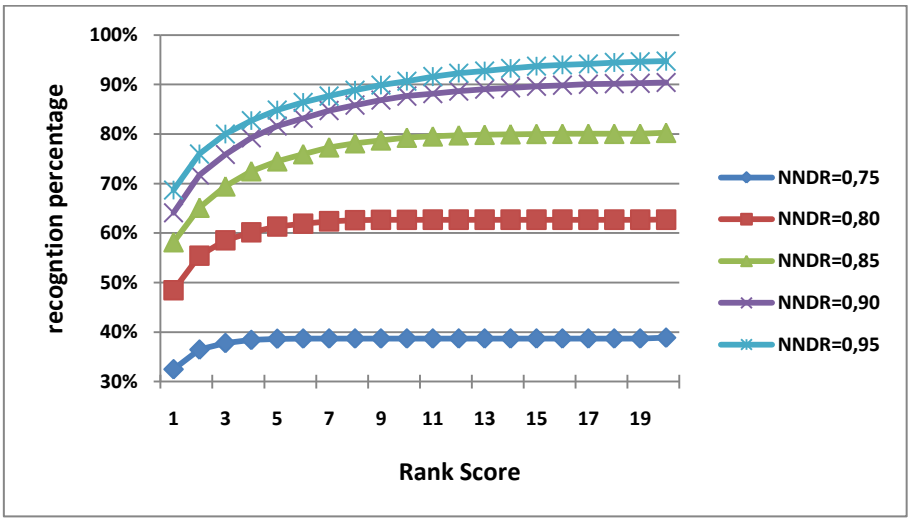


Figure 4-30: chart showing the influence of the NNDR on the performance when M=25 images are used for models (and N=10 images for the query)

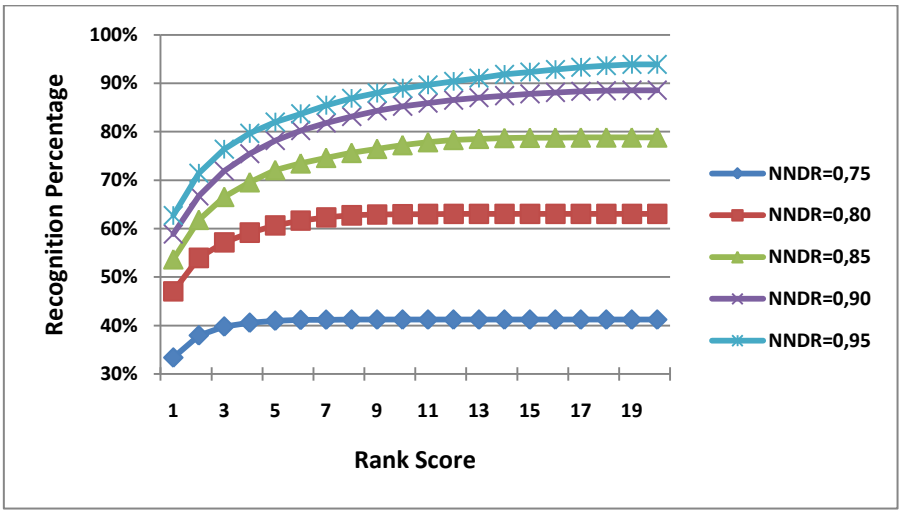


Figure 4-31: chart showing the influence of the NNDR on the performance when M=10 images are used for models (and N=10 images for the query)

We can see on figure 4-30 & 4-31 that NNDR dramatically affects the performance. Note that this value depends on the number of images which construct the model (i.e the number of keypoints). We can note that the value of NNDR=0.95 gives the best performance. However in the work of Lowe (Lowe, 2004), he chose the value=0.85. We can also see that the impact of this parameter is more significant when the number of points augments. The difference in first rank recognition is about 37% M=25 images and about 29% for M=10 because there more redundancy of information which decreases the performance.

The performance with fewer images in model is limited by this method because all the points have the same weight in this situation, while we saw in figure 4-29 that some points have more information than other points. So these points should have more weights, and we should weaken the points which are not distinctive. In addition we didn't use the temporal frequency of interest points, as we compare two sequences so that some interest points have more frequency in the query. We weight these points by their temporal frequency and we assume that the interest points are found in the highest part of the silhouette are more important. For the descriptors of an image of the query, there are several correspondences. Each of the point-wise correspondences is weighted by its variability and its saliency. We then search for a set of correspondence which reinforces each other.

4.4 Proposed improvement for the re-identification

In this section, we propose to use a principle derived from video Google which was proposed by Sivic and Zisserman (Sivic, et al., 2008) and consists in weighting the descriptors according to their frequencies in each model compared with their global frequency.

The method used is a probabilistic model, taking into account the variability of interest points as well as the probability of temporal spatial configuration. It is structured to use the matching results, the frequency of the descriptor in the model and in all models, and the temporal spatial relations between the matched interest points. Using this model has several advantages: first, it makes the recognition more robust as the uncertainty is integrated into the model, secondly it is more adaptive in the sense that there is no need to arbitrarily fix a threshold on votes.

We can distinguish two steps of this method: the first constructs the model and the second constructs the query. To construct the model, we propose two variants. The first act at the level of model frame, and the second act at the set of interest points level. However in the two variants the interest points are weighted proportionally to their frequency in the model, and inversely proportional to their global frequency in all models.

4.4.1 Construction of the model

A model is constructed for each tracked pedestrian in the sequence. We accumulate in this model the discriminating interest points of the pedestrian. The goal of the construction of the probabilistic model is to identify the interest points of a model which are more significant than other points and to avoid the redundancy of information to save matching time and storage space. Each model is defined by a set of interest points which are selected depending on the quantity of

information which are represented in the models or by adding information from the frame. We will discuss now these two methods of selecting interest points.

4.4.1.1 Frame level

Let's assume that for model M we have one track T^Mcontaining N consecutive images$T^{(M)} = \{I_0^M, I_1^M, \dots, I_N^M\}$. It is important to select from these images the key frames that contain all the information in the sequences. The selection process is performed as follows. The first frame is selected as the first key frame. It becomes the current key frame F_i for the following steps. Then, the keypoints in next image I_j^m are compared to keypoints in current key frame F_i. If the percentage of matched points is less than a threshold, this current frame becomes the next current key frame F_{i+1}. Inversely, if this ratio is greater, we update the frequencies of the corresponding keypoints in frame key frame F_{i+1} , and the following frame I_{j+1}^m will be analyzed. The points of frames $I^M{}_t$ and $\mathrm{I_j^M}$ are matched by a cross-validation method which rejects most of outliers between the two frames. The cross-validation consists of finding the couples of points that are mutually selected: for a point of interest $\mathrm{p_t^i}$, the point $\mathrm{p_{t+1}^j}$is the nearest point to $\mathrm{p_t^i}$. Then, the closest point to $\mathrm{p_{t+1}^j}$ in $\mathrm{I_t^M}$is $\mathrm{p_t^k}$. If the points p_t^i and p_t^k correspond to the same point in the frame I_t^M, it means that the points p_t^i and p_{t+1}^jhave chosen each other, and the matching was successful. Otherwise, the match is rejected. Figure 4-32 illustrates the principle of cross validation. The cross-validation method allows obtaining the most correct matches.

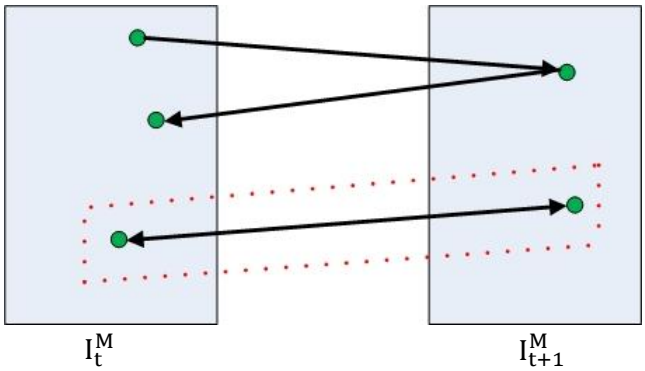


Figure 4-32: Illustration of the cross-validation method. The matching surrounded by a red box is validated because points of interest have chosen each other. In contrast, the other matching is rejected because there is no mutual selection.

In this way, the frames with large information gain or containing new information are selected, and those not selected can be represented by the key frames. In figure 4-33 illustrates the key frames of some pedestrians.

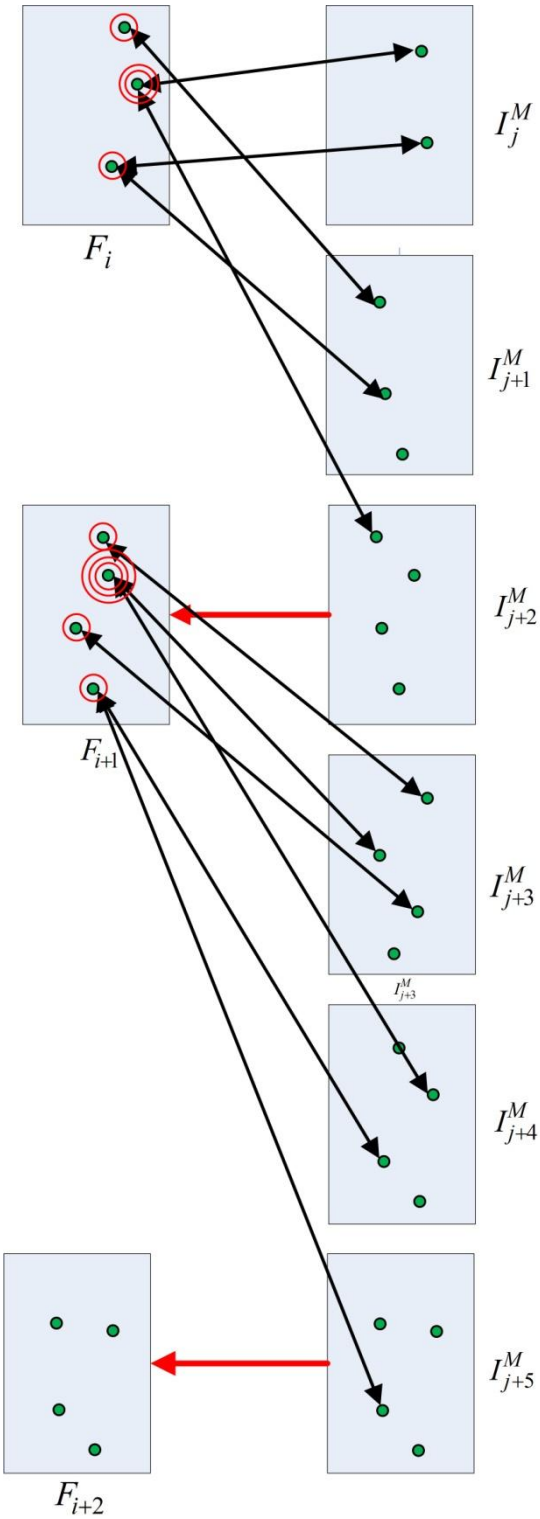


Figure 4-4-33: Key frame selection: the keypoints in current image I_j^m are compared with keypoints in current key frame F_i. If the percentage of matched points is less than a threshold, this current frame becomes the next current key frame F_{i+1}. In all cases, the numbers of matching for keypoints in the current keyframe (illustrated by circles on figure) are updated.

Once the key frames are selected, we accumulate the interest points of all key frames to construct one model. Then we employ the "*term frequency-inverse document frequency*" (tf-idf) weighting, which is used in text retrieval. Sivic and Zisserman (Sivic, et al., 2008) propose to use it in visual search of objects in video. The weight for keypoint i in the model m is given as follows:

$$w_i = \frac{n_{im}}{n_m} log \frac{N}{N_i} \quad \text{Eq.4-1}$$

Where n_{id} is the number of occurrences of interest point i in all frames of model m, n_m is the total number of points in the model m , N_i is the number of models containing interest point i, and N is the number of models in the whole database. The weighting is a product of two terms: the interest points frequency,$\frac{n_{im}}{n_m}$ and the inverse document frequency $log \frac{N}{N_i}$. The idea is that the interest point with frequency weights occurring more often in a particular model is higher (compared to interest points present/absent), and thus describes it well, while the inverse interest point frequency down-weights the interest points that appear often in the database, and therefore do not help to discriminate between different pedestrians.

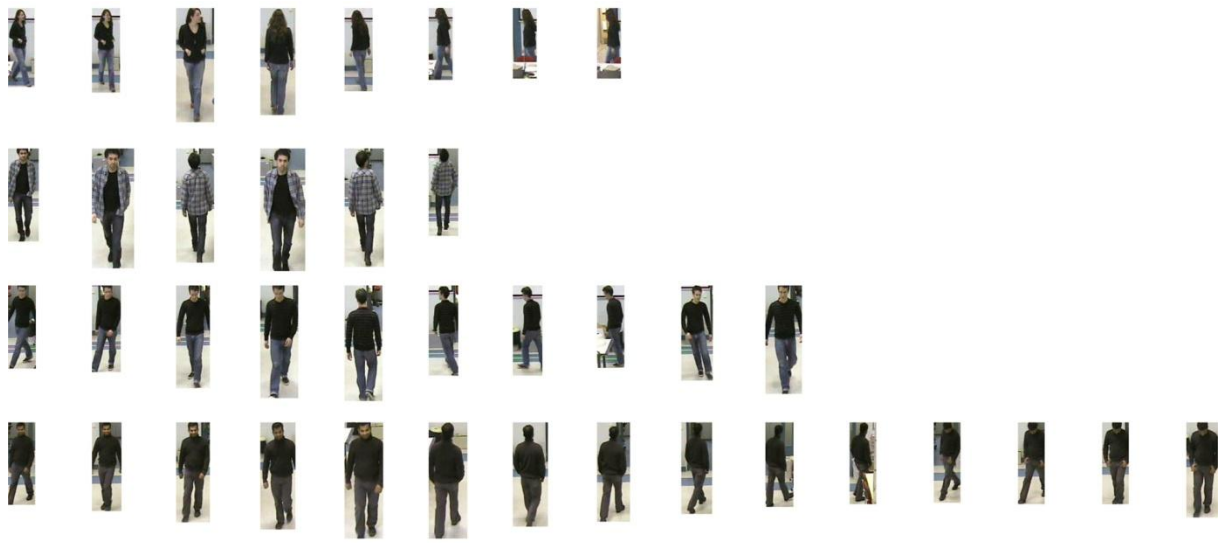


Figure 4-34: example result of key frame selection (in each case the frames are selected among 200 frames of a sequence)

4.4.1.2 Interest points level

The goal of our method is to identify the discriminative interest points of a model, so instead of comparing two successive frames to find which frame has more information, we compare the interest points of the current frame with the global interest points. If the interest points are not found in the model, we add it to model. But to avoid the erroneous interest points we filter those interest points by comparing the interest points to current frame with its neighbours in the previous frame, and we employ the clustering of the interest points which are found into two frames. The principle of clustering is similar to the background construction in chapter three, but there is a difference in the matching method. We use the cross-validation algorithm here.

4.4.1.3 Comparison of the two variants of constructing models

The two variants have similar performances (see figure 4-35). But using the interest points level allows to reduce the quantity of these points, on other hand the frame level variant increase the number of interest points but the calculation time is smaller than using the level of interest points.

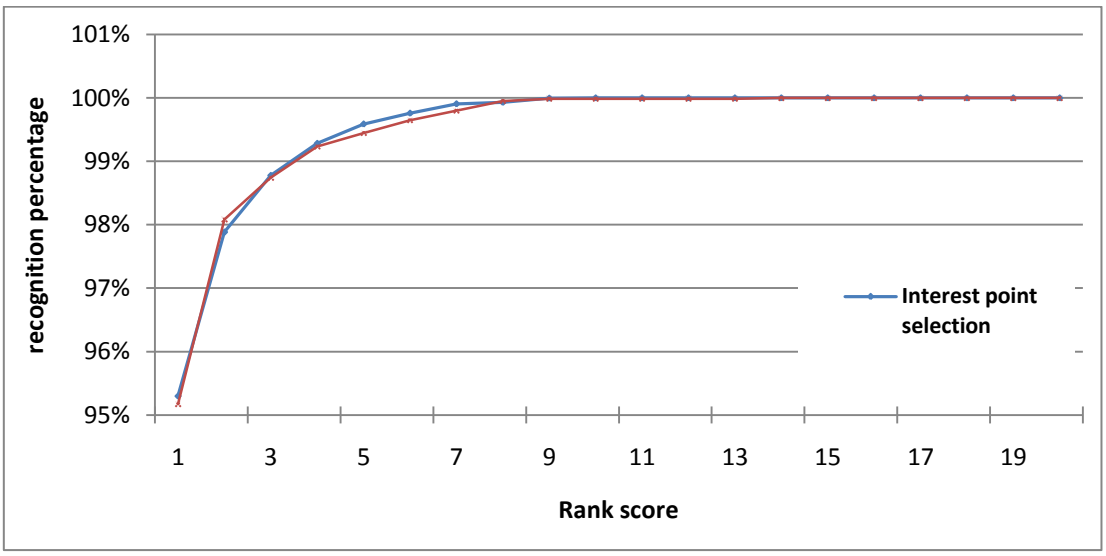


Figure 4-35: chart represents the performance using interest points selection and frame selection, the query is constructed using 20 frames in both cases.

Indeed the performance is similar due to the two variants constructing the model using the most significant features. The difference is the number of interest points. In the previous example the number of interest points using the frames is 70000 points. On the other hand, the number of interest points is 20000 points, when we select the interest points.

4.4.2 Construct the query

When we use the vote technique to find the most similar model for a given query, the idea is to accumulate the number of times the interest points of the query corresponds to a model. Hence each time a model M_k is selected. Table of votes T is updated so that the value T (k) is incremented as we can see in Eq.4-2:

$$T(k) = \sum_{t-n}^{t} \sum_{j=1}^{|p_i \in M_k|} \begin{cases} 1 & if\ the\ corresponding\ of\ the\ interest\ point \in M_k \\ 0 & otherwise \end{cases} \quad \text{Eq.4-2}$$

Where n is the number of the images of the query, and $|p_i \in M_k|$ is the number of correspondences for an image of query in the model k. The model with the highest score $T(k)$ is considered to be the most similar model. But in the probabilistic approach $T(k)$ is given by:

$$T_t(k) = \sum_{t-n}^{t} \sum_{j=1}^{|p_i \in M_k|} \begin{cases} p_{jkt} & if\ the\ corresponding\ of\ the\ interest\ point \in M_k \\ 0 & otherwise \end{cases} \quad \text{Eq. 4-3}$$

Where forP_{jkt}, we use the same formulas proposed by Schmid (Schmid, 1999) except we also add the temporal coherence:

1. The quality of the correspondenceP_c.
2. The spatial coherence of the corresponding interest points P_s
3. The temporal coherence of the corresponding interest points between successive frames of queryP_t.

The probability P_{jkt} is then the product of three independent probabilities.

$$P_{jkt} = P_c * P_s * P_t \qquad \text{Eq. 4-4}$$

The way these three probabilities are computed in detailed in the following section:

4.4.2.1 The probability of correspondence

There are two independent factors affected to the probability of correspondence: the similarity of the matched descriptors of interest points and the frequency of these interest points in the models. So the probability of a correspondence is then the product of two independent probabilities similarity P_{cs} and frequencyP_{cf}, where P_{cs} is defined as the inverse SAD distance from its best match keypoint. A query point that closely matches a model keypoint, therefore, would have a very high probability; conversely, a poorly matched keypoint would have low probability. Therefore, the probability of similarity is given by:

$$P_{cs} = \frac{1}{dist\left(d_{p_i}, d_{p_j}\right)} \qquad \text{Eq.4-5}$$

The probability of frequency, we had computed when we constructed the mode. So the correspondence probability is given by the product of the two probabilities:

$$P_c = \begin{cases} P_{cs} * P_f & if\ the\ corresponding\ of\ the\ interest\ point \in M_k \\ 0 & otherwise \end{cases} \qquad \text{Eq.4-6}$$

4.4.2.2 The probability of spatial configuration

The rate of incorrect matches can be reduced by using a neighborhood and geometric constraints. These constraints are based on the assumption of constancy of the structure of the neighborhood. Firstly a correspondence is only kept if a percentage of the positions of matched interest points respects the geometric constrains. For our experiments, the relative positions of these points are geometrically consistent if they do not vary by more than 15% vertically and horizontally. In figure 4-36 we can see that the effect of the use of geometric constrains is low because of the good match.

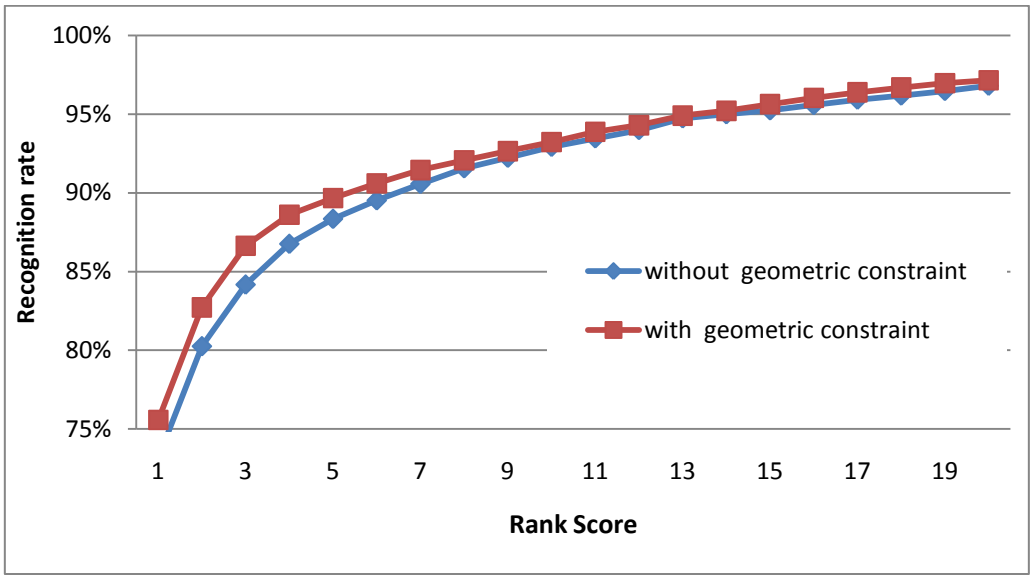


Figure 4-36: chart represents the influence of geometric constraints

The weight of the spatial configuration in our case depends on the number of interest points in an image of the query which corresponds to a model in the database. We do not assume that the interest points of the image of the query is the product of an affine transform of the interest points of the model like Lowe in (Lowe, 2001) because the pedestrian is a non-rigid object. Another solution using semi-local configuration like Sivic et al (Sivic, et al., 2003) is more feasible. However our problem in this case where the number of corresponding points between an image of the query and an image of the model is low (depending on the richness of textures). So we assume a simple spatial configuration. The spatial probability is given by:

$$P_s = \begin{cases} \frac{N_i}{M} & if\ the\ corresponding\ of\ the\ interest\ point \in M_k \\ 0 & otherwise \end{cases} \qquad \text{Eq.4-7}$$

Where N_i is the number of interest points from model i corresponding to the interest points of the query, and M_t is the number of matching points in the image of query in time t.

4.4.2.3 The weight of temporal configuration

The weight of the temporal configuration depends on the saliency of the matched interest points in the query. In this case the saliency of the interest point is based on the frequency of the model in previous images of the query. This weight allows to profit of the fact that the query is a sequence of images. The temporal weight is given by:

$$P_t = \begin{cases} \frac{1}{d}\ if\ t < n_w \\ \alpha * P_{jk(t-1)} + (1-\alpha) * P_{t-1}\ otherwise \end{cases} \qquad \text{Eq.4-8}$$

Where d is the number of models in the database, n_w is the number of images used to construct the query, where α is a time constant that specifies how fast new information supplants the old query. The advantage of using the temporal weight is that the importance of correspondence is increased due to the accumulated vote from multiple query frames, whereas false positives tend not

to be consistent. This step is equivalent to finding the frequency of the interest points of the model but in real time.

Correspondences are determined by using the SAD distance with a distance below a certain threshold. We use the kd-tree to store the interest points of all models to avoid an exhaustive comparison.

4.4.3 Experimental Results

This section displays the gain obtained by adding a probabilistic model. Experimental results for our dataset clearly show the improvements: the recognition rate increases and the number of interest points decreases.

4.4.3.1 Comparison of the probabilistic approach with the first approach

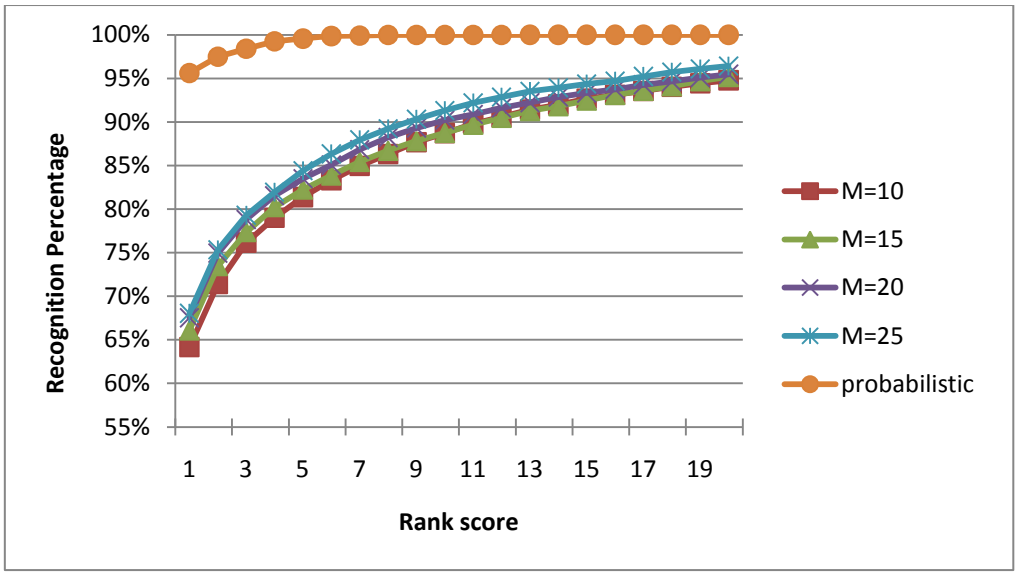


Figure 4-37: comparison between the probabilistic model and the first models. We note that the performance of the probabilistic model outperforms other initial models.

In the next experiment, the re-identification performance between cameras 1 and 2 is tested. The effect of changing the number of images of the model is shown in figure 4.37 and the number of points which are used to construct all models in figure 4.38. The first chart shows that a probabilistic model outperforms the first approach in spite of the increase of the number of images. A very interesting observation can be seen when looking at the lower ranks, where the performance is very high when we use the probabilistic model. In figure 4.38 we note that the number of points is not the principal factor of our method but the discriminative information of these points is more important.

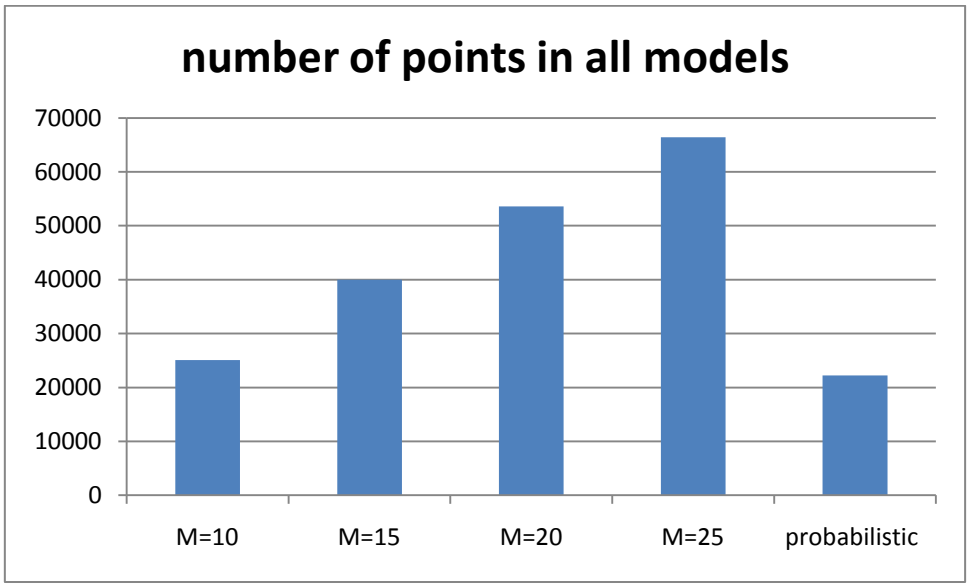


Figure 4-38: the number of points used used to construct the models when we change the number of points. We note that in spite of the small number of points in the probabilistic model its performance is the best.

Figure 4-39 shows the influence of the number of images constituting a query on the performance. We can see that this influence is small compared to the first approach. This is because the interest points which construct the model are more important than the interest points which construct the query.

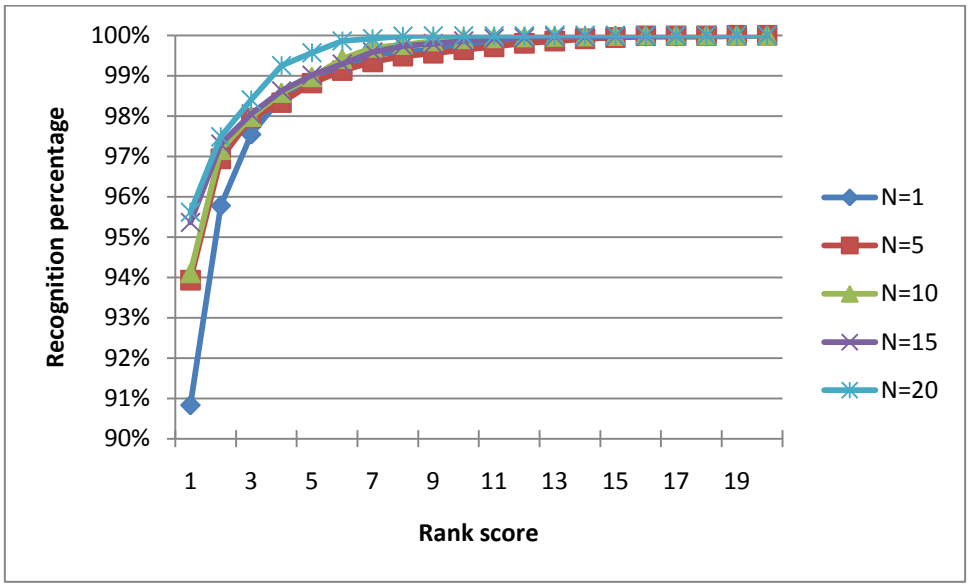


Figure 4-39: Influence on CMC of the number N of images in query, for the probabilistic variant with SURF

Another evaluation of our algorithm is done. We tested the algorithm using the first twenty pedestrians. We calculated the confusion matrix with respect to the number of images which construct the query. We can notice that the value of the diagonal of the matrix is increased (See figures 4-40, 4-41, 4-42) when the number of images of query augments. Compared to the first

approach we can notice that the confusion is almost solved and that we do not need the same number of images used in the old method.

id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	99,42	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	8,9	0	91,1	0	0	0	0	0	0	0	0	0	0	0
10	0	0	24,72	0	0	0	0	0	0	75,28	0	0	0	0	0	0	0	0	0	0
11	4,32	0	0	0	0	0	0	0	0	0	94,44	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	99,5	0	0	0	0	0	0
15	0	0	1,2	0	0	0	0	0	0	0	0	0	20,35	0	78,45	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	3,09	0	0	96,91	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	12,5	0	0	0	0	87,5	0	0
19	0	0	0	0	0	16,34	0	0	0	0	0	0	0	0	0	0	0	3,92	79,74	0
20	0	0	0	0	0	0	0,74	0	1,48	0	0	0	0	8,89	0	0	0	0	0	88,15

Figure 4-40: the confusion matrix of the first twenty pedestrians with 1 image to construct query

id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	99,43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	87,38	0	0	7,86	0	0	0	0	4,76	0	0
11	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	4,21	0	95,79	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	16,2	0	83,8	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	3,29	0	0	0	0	0	96,71	0
20	0	0	0	0	0	0	0,8	0	0	0	0	0	0	0	0	0	0	0	0	99,2

Figure 4-41: the confusion matrix of the first twenty pedestrians with 5 images to construct query

id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	99,43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	97,62	0	0	0,6	0	0	0	0	1,79	0	0
11	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	4,21	0	95,79	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	12,84	0	87,16	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	1,99	0	0	0	0	0	98,01	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure 4-42: the confusion matrix of the first twenty pedestrians with 10 images to construct query

In the next experiment, we split each sequence of pedestrians into two sequences depending on the direction of their movements. The first sequence presents the pedestrian when he approaches the

camera. In this case there is more texture than in the other sequence where most images are present on the back of the pedestrian which is more interesting due to the poorness of discriminate information. The results for the two cases are presented in figure 4.43 and figure 4-44. The probabilistic model dramatically improves the matching performance in both cases especially at low ranks. Even if the back provides only little information, they still generate a good performance. It is not surprising that the recognition performance decreases when only using the back of the pedestrian.

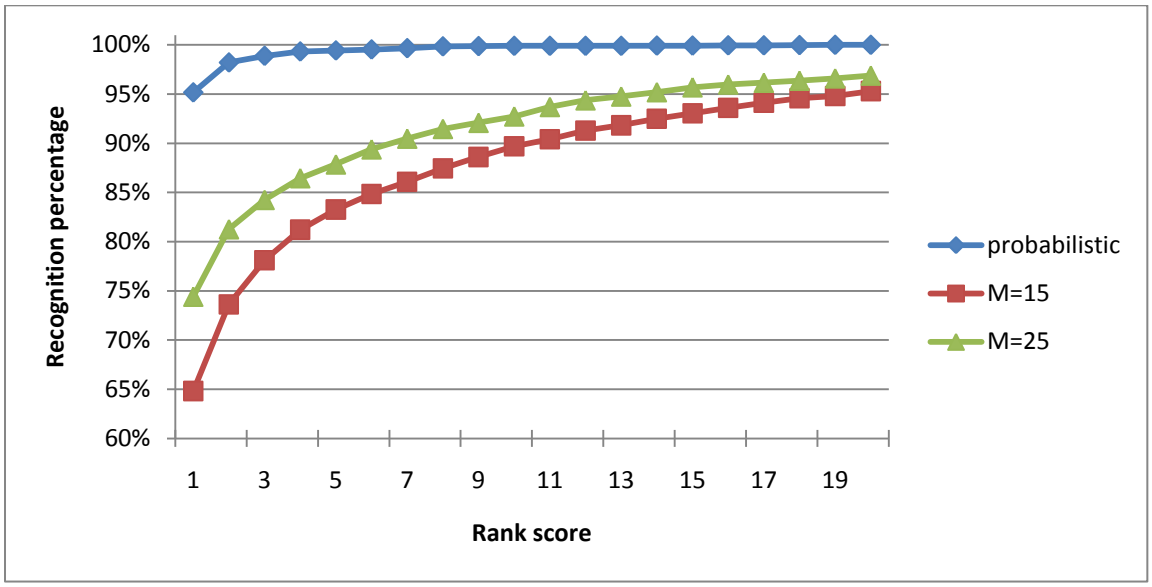


Figure 4-43: the performance of the matching of the sequence of images of the front.

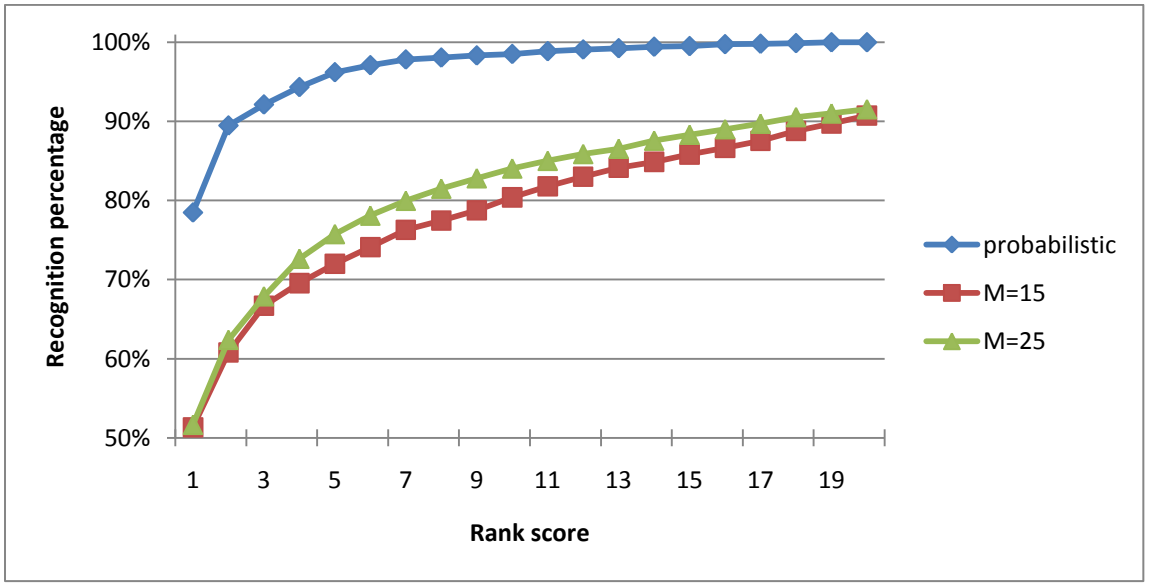


Figure 4-44: the performance of the matching of the sequence of images of the back.

Summary

Experiments showed that the model provides the most discriminative power, while the overall accuracy is still best when we increase the number of query images. The back showed a low discriminative power. This may be due to the poorness of the texture in the back of the pedestrian. Using probabilistic models increase the influence of the discriminate information. On other hand, decreasing the quantity of interest points makes this method more realistic, as we can see in the next section when we evaluate the algorithm using ETHZ.

4.5 Experimental Results using ETHZ

We studied the effect of increasing the number of images constituting a model as well as the query. In the experiments, the elements for the model and the query are chosen randomly. The experiment was repeated 10 times to provide reliable statistics. We varied the number of images of model N = {1, 2, 5, 10}, figure 4-45 shows the pedestrians of first sequence; we can see the occlusion between them. We compare our method with the results of PLS method in (Schwartz, et al., 2009)and SDALF method (Farenzena, et al., 2010).

Figure 4-4-45: Samples of different people in sequence #1 used to learn the models (figure extracted from (Schwartz, et al., 2009).

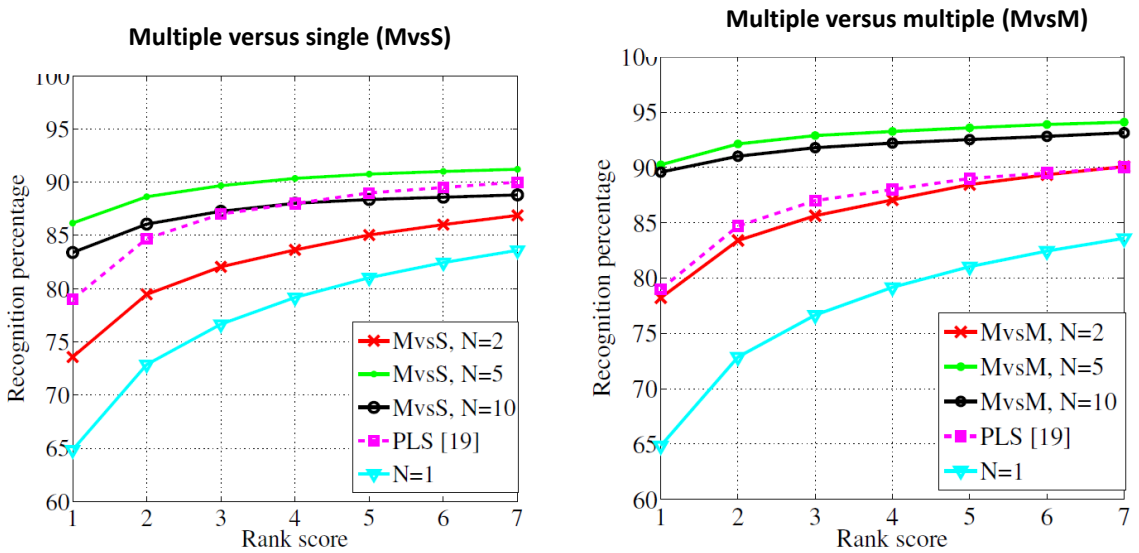


Figure 4-46: The results of SEQ. #1 using PLS method in (Schwartz, et al., 2009) and the results of SDALF in (Farenzena, et al., 2010). On the left are the results for single-shot SDALF (one image per model and one image per query) and MvsS SDALF (several images to construct the model and one image for the query). On the right, the results for MvsM SDALF (multiple images to construct the model and query). In accordance with what is reported in (Schwartz, et al., 2009), only the first 7 ranking positions are displayed (figure extracted from (Farenzena, et al., 2010).

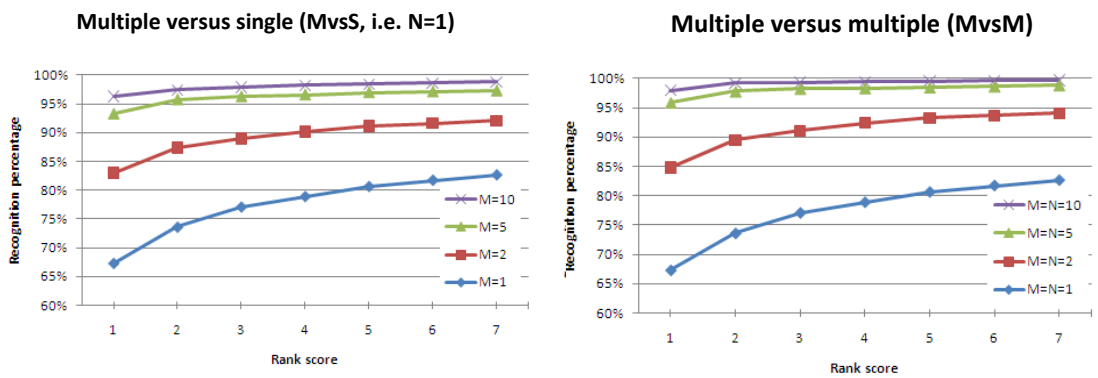


Figure 4-47: The results of SEQ. #1 using our method.

For the first sequence which is the largest data set (83 pedestrians), we do not obtain the best results in the single-shot case (compare lowest curves of figure 4-47 with PLS Curve of figure 4-46), the performance of this case varied depending on which image was used to construct the model. However using multiple images to construct the model increases the recognition percentage up to 96% rank 1 correct matches for MvsS (multiple images to construct the model and one image for query) and up to 98% for MvsM (multiple images to construct the model and query), which is clearly above results of other methods (compare upper curves of figure 4-47 to upper curves of figure 4-46).

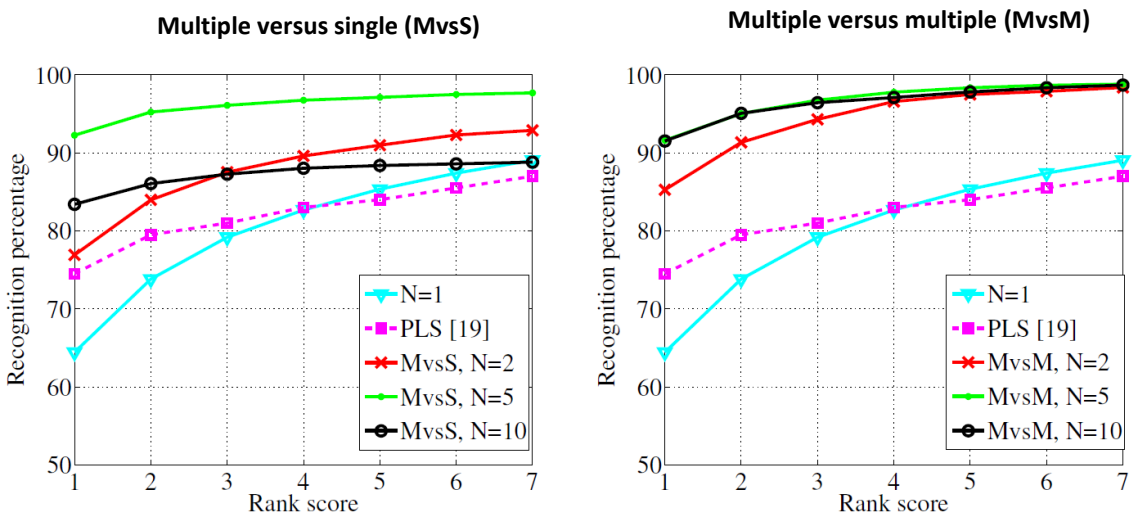


Figure 4-48: The results of SEQ. #2 using PLS method in (Schwartz, et al., 2009) and the results of SDALF in (Farenzena, et al., 2010).

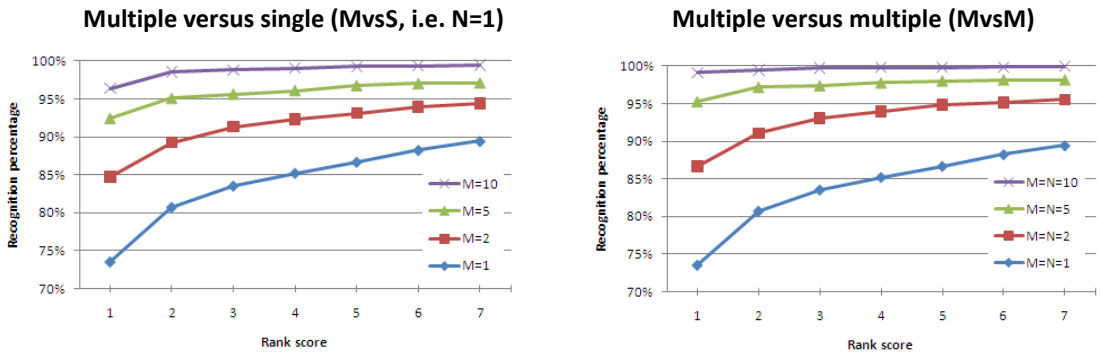


Figure 4-49: The results of SEQ. #2 using our method.

Figure 4-49 represents the evaluation result of the second sequence which contains 38 pedestrians (1,936 images) as we can see the performance of the single shot case is similar, and the recognition percentage increases up to 96% of the cases for MvsS, and up to 99% of the cases for MvsM. Figure 4-51 illustrates the performance of the third sequence which contains 28 pedestrians (1,762 images), our method outperforms PLS and SDALF in most cases except (N=5 & M=5). The best performance of rank 1 correct match is 99% for MvsS and 99% for MvsM.

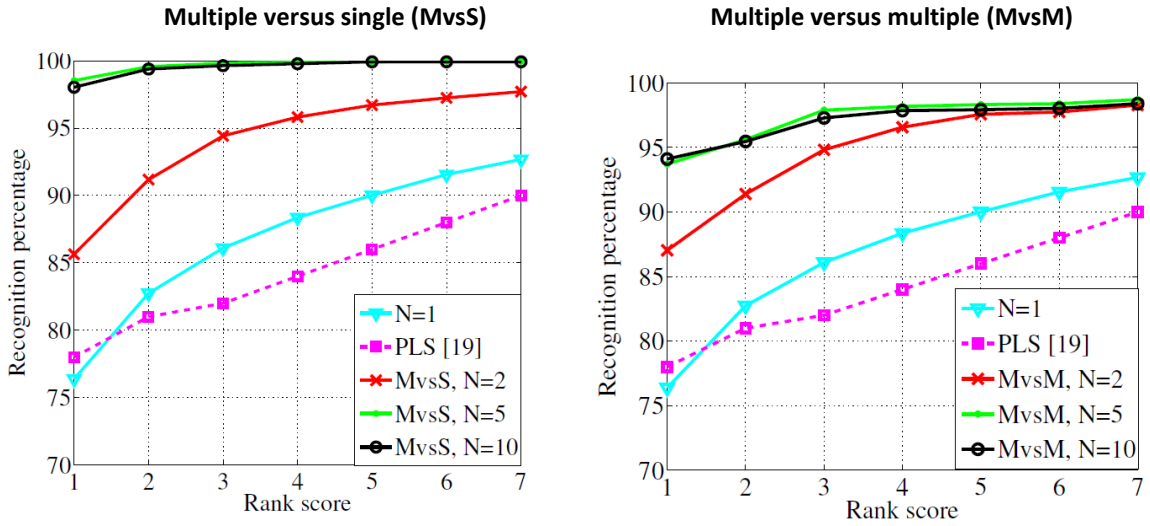


Figure 4-50: The results of SEQ. #2 using PLS method in (Schwartz, et al., 2009) and the results of SDALF in (Farenzena, et al., 2010).

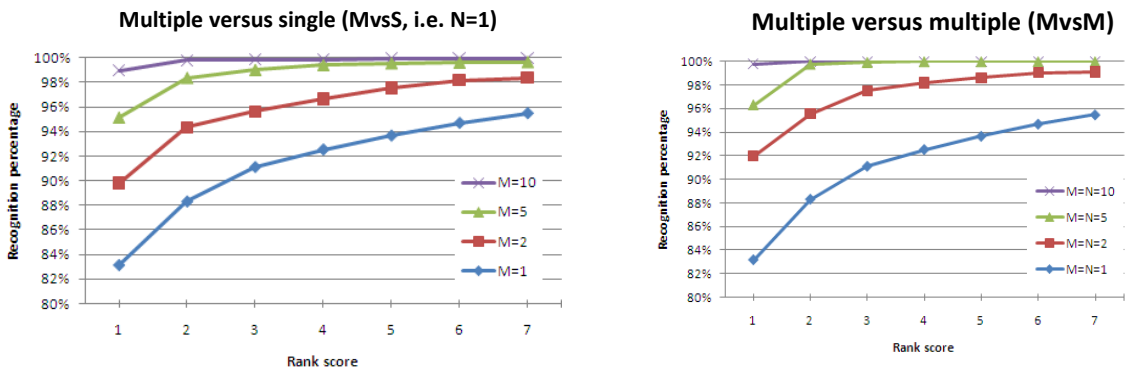


Figure 4-51: The results of SEQ. #3 using our method.

We grouped the three sequences in one sequence containing 146 pedestrians to evaluate our algorithm. Fig 4-52 illustrates the results. We can see the high performance of our algorithm. The objective of this evaluation is to study the performance of the algorithm when the number of pedestrians increases, although the data set is less difficult in comparison with the other two datasets (The model and the query are taken from same camera sequence). However, it gives a general idea of how our approach works even for hundreds of people.

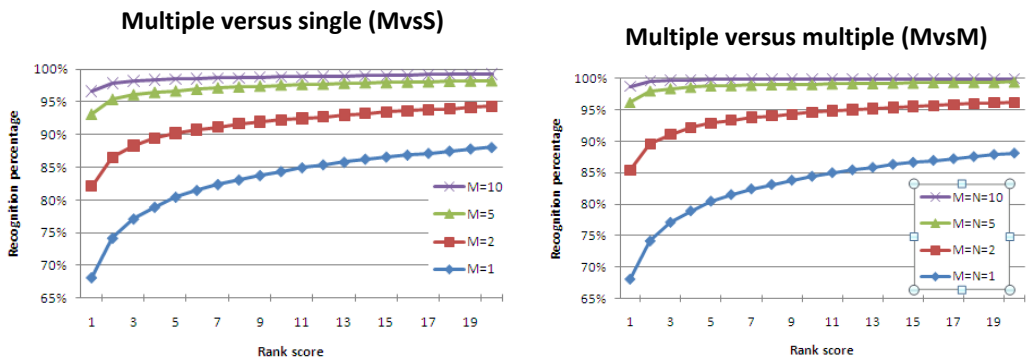


Figure 4-52: The results for 146 pedestrians.

4.6 Comparison with other region descriptors

To further evaluate the performance of the proposed method to re-identification, we study several object descriptors. We start with SIFT descriptor which outperforms other local descriptors. Then we evaluate two global descriptors: Color Histogram and HOG descriptors.

4.6.1 Comparison SURF vs. SIFT

To further evaluate the strength of the proposed method, we evaluate its performance by using SIFT features which are commonly used in recognition and detection. We have described this descriptor in chapter 2. We substitute SIFT to SURF. We do not need to tune neither the training. First we evaluate the different numbers of images per model. The cumulative match curves(CMC) are shown in figure 4-53 where we can see that the performance of SURF features is better than the SIFT features. This is because the SIFT detector produces a larger number of keypoints compared to the SURF detector. For example the number of points to construct all 40 models using SIFT features is 88000 points, while the number of SURF features is 40000 points. This number of descriptors increases the chance to mismatch. The idea of Lowe (Lowe, 2004) is to generate many features then filter the correspondence using geometric constrains and affine transforms which are used in our method due to the fact that the pedestrian is a non rigid object. We can see in figure 4-54 the matching between the query and the models using the same parameters for SIFT and SURF. The number of correct matching using SURF is larger than those using SIFT. And we must also mention that it takes about 18.1minutes (Intel Core i7 CPU, 2.93GHz, RAM4GB) to evaluate the whole sequence of 40 pedestrians (10000 images). When we use the Surf features, the computation time for matching is reduced to about 5.30 minutes.

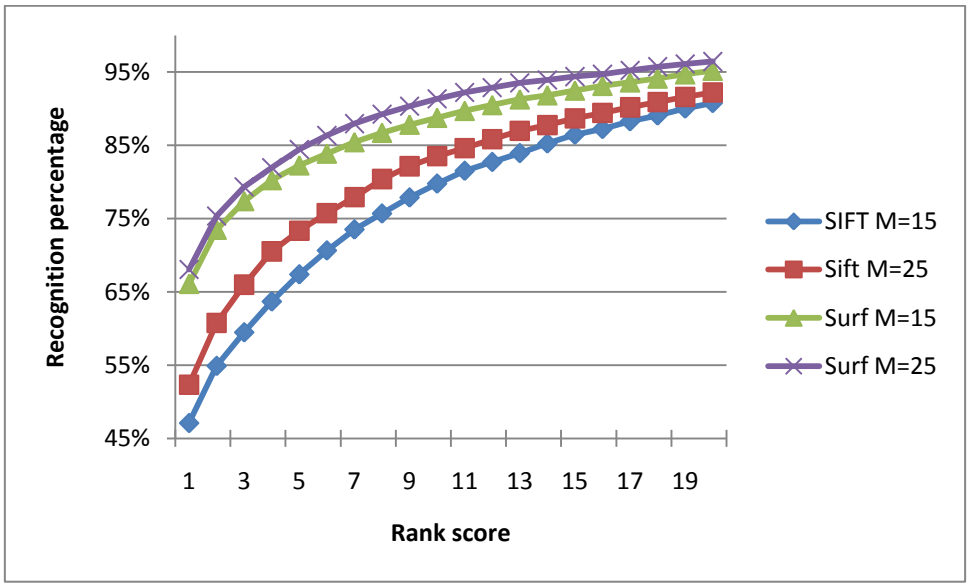


Figure 4-53: Comparison between the SIFT and SURF. We note using SURF gives much better results than SIFT in our application.

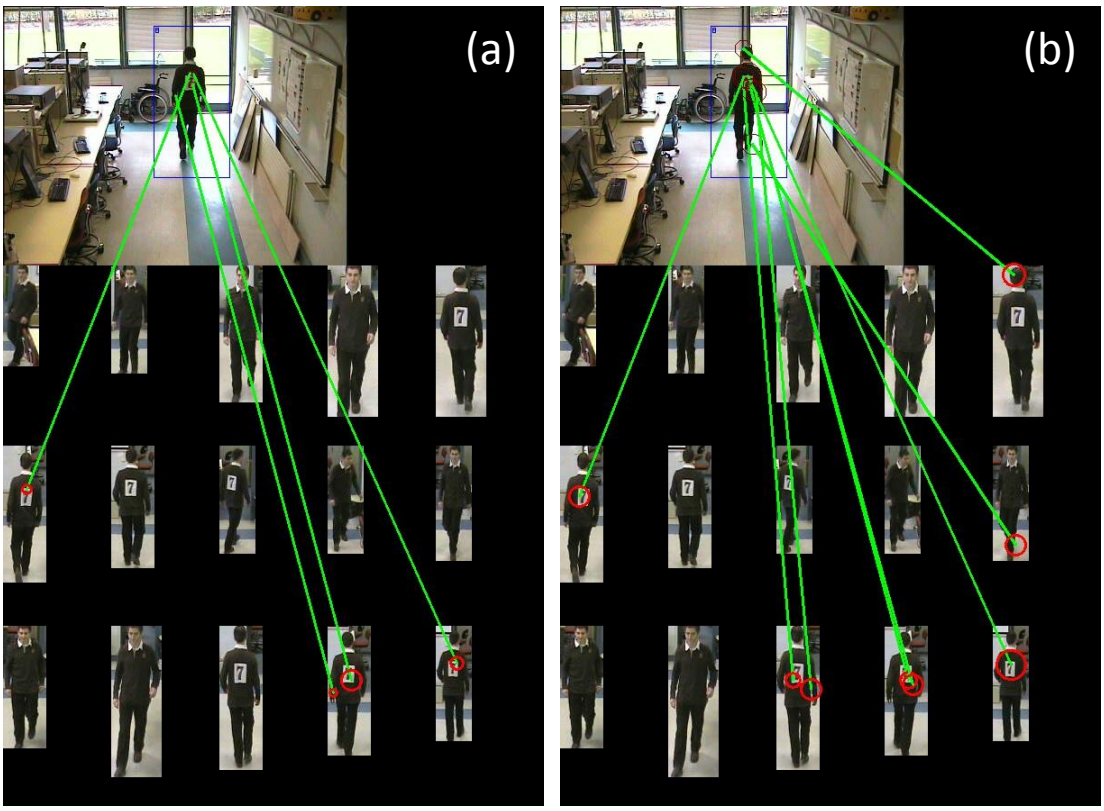


Figure 4-54: Correct keypoints matching (a) using SIFT features, (b) using SURF features. We can see that the number of correct matches using SURF is larger than using SIFT features.

To generalize the idea of the probabilistic model, we perform this method using SIFT features. Figure 4-55 shows the result of comparing the probabilistic model with the initial method. We note that, as with SURF, the performance of re-identification is increased using the probabilistic model. At the same time, the number of points used to construct the model is decreased. The number of points in this case is about 70000 points, but the calculation time changes slightly to 17.2 minutes (instead of 18.1 minute). As we know the calculation time depends on the Kd-tree which constructs the model but the main part of the computation time depends on the detection of interest points and the calculation of descriptors to construct the query which is done in a fixed time in both cases.

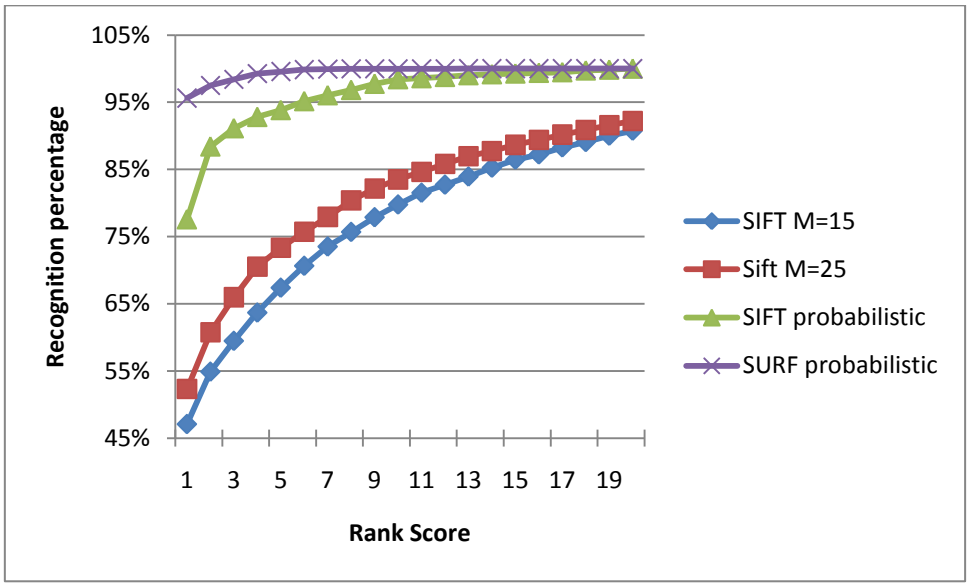


Figure 4-55: comparison between the probabilistic model and the initial models, in the case of SIFT features we note that the performance of probabilistic model using SIFT gives much better result than initial method, but that probabilistic SIFT remains below probabilistic SURF

4.6.2 Comparison SURF vs. COLOR

We also tested the global approaches to compare with our algorithm. The most used feature in re-identification is the color as we described in the state of the art. The color features of a pedestrian have been represented in a number of ways and used to perform a variety of image understanding tasks. These methods differ according to the color space used to define the observed color. The color calibration used, the distance measure used to compare the features and the representation of the distribution. The most used representation is the histogram which is invariant to spatial positioning. This property is advantageous for the representation of non rigid objects of variable pose. Nevertheless the drawback of a global histogram representation is that the information about object location, shape, and texture is discarded. To avoid this disadvantage, a large variety of techniques exists for incorporating the spatial information e.g. spatiogram (Cong, et al., 2009) .In this approach, a very simple and intuitive technique is employed. Instead of computing a global histogram for the whole pedestrian, the box of pedestrians is divided into several slices. Moreover we take advantage of our framework by considering each slice as a keypoint. We perform the same schema used for local features. We evaluate different color spaces (RGB, HSV, XYZ, Luv), different numbers of bins in the histogram and the number of slices

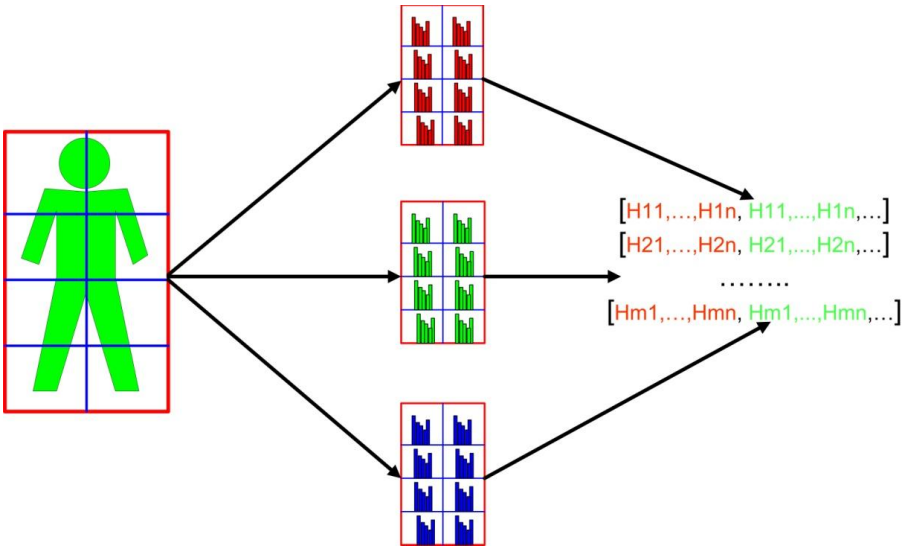


Figure 4-56: Schema of the construction of color features of pedestrians

To compare two histograms H_1, H_2, there are many measures which we can use. We used the Bhattacharyya distance as shown in equation $Eq.$ 4-9. N denotes the total number of bins and $H_i$denotes the histogram value of bins i.

$$d(H_1, H_2) = \sqrt{1 - \sum_{i=1}^{N} \frac{H_1(i).H_2(i)}{\sqrt{\sum H_1(i).\sum H_2(i)}}} \quad Eq. 4\text{-}9$$

When calculating a single histogram for each color channel (in case of an RGB image), a distance comparing all three histograms can be realized for example by taking the sum of all distances .To make the histogram invariant to scale, we normalize the histogram. Hence, the comparison of two histograms is not influenced by the size of the slice.

$$h(i) = \frac{H(i)}{\sum_{i=1}^{N} H(i)} \quad Eq. 4\text{-}10$$

To avoid the effect of the background, codebook-based background substraction (Kim, et al., 2005) was used to segment the moving people. The small noise is filtered and the morphological operations of closing and connected component analysis are used to obtain the silhouettes of people. As we have said in the state of art of re-identification, the disadvantage is the fact that colors appear very differently in varying lighting conditions. The responses of cameras are dependent on several factors, such as surface reflectance, the cameras parameters, lighting geometry…. So the color normalization procedure has been carried out in order to obtain an invariant signature. We use a technique called histogram equalization (Finlayson, et al., 2005) which consists of equalizing independently the histogram of each color channel of an image. Figure 4-57 illustrates the effect of applying the histogram equalization technique to images of the same individual captured by different cameras. These images highlight the fact that a change in illumination leads to a significant change in the colors captured by the camera. The right column shows result images after applying histogram

equalization procedure. It is clear that the resulting images are much more similar than the two original images. Figure 4-58 illustrates the improvement of using the histogram equalization.

Figure 4-57: The left column shows original images of the same person captured from different cameras in different environments. The right column shows result images after applying histogram equalization procedure.

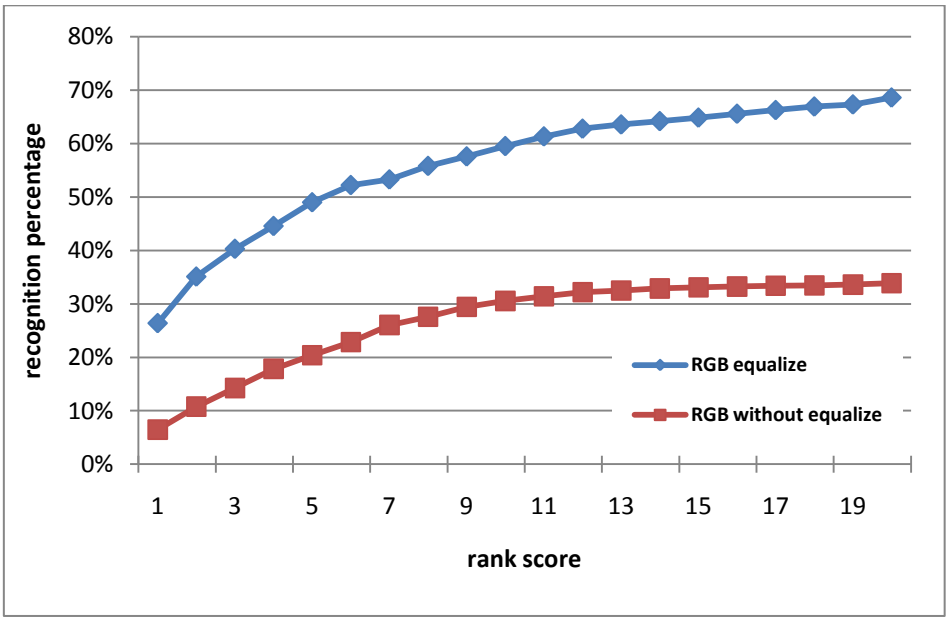


Figure 4-58: Influence of the normalization procedure using histogram equalization. We note that the equalization improves the recognition rate very significantly.

Six color spaces have been evaluated. We divide the person into eight vertical slices, and then use a color histogram with 32 bins for each color. The performance of these descriptors is presented in figure 4-59. We can note that the color features perform poorly with the histogram compared to the local features (SURF-SIFT) as we can see in figure 4-59. This is because the sequences are registered as jpg and so the histogram equalization increases the distortion of the compression as we can see in figure 4-60. We do not have the same effect in the case of local features because we filter around, the patch, as we have described in the chapter three, using a Gaussian filter which decreases the effect of the distortion.

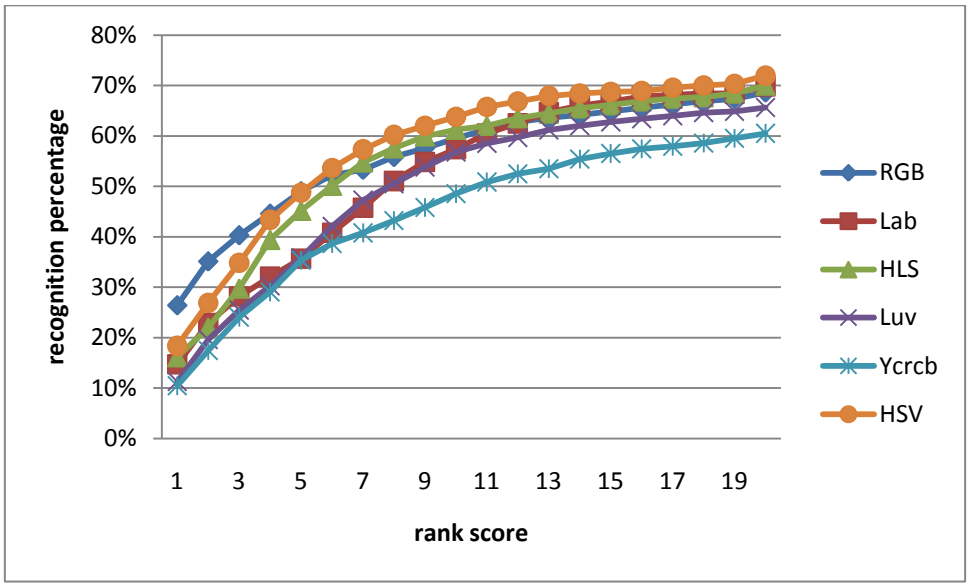


Figure 4-59: Performance of recognition using various color spaces.

Figure 4-60: the influence of compression distortion

It is worth noting the use of multiple images to construct the model and query (25 images, 20 images respectively) because although the sequences are taken indoors, the illumination varies significantly as the pedestrian moves towards an extended and proximal light source. In figure 4-61, we can see the effect of illumination on colors, even when we use the histogram equalization.

Images of model. We note that the color varies

Figure 4-61: Images of query. We note that the color varies

We study also the influence of increasing the number of slides vertically and horizontally (the illustration of used slices in figure 4-62); figure 4-63 illustrates the performance of this increase. We note that increasing the number of slide increases the performance of the histogram of color. But we must mention that increasing the number of vertical slice has more influence as we can see in figure 4-63 (look for instance at improvement from curve 1-2 to curve 1-4). Because in this case we isolate the lower part of the silhouette where there is not a lot of information, as people tend to wear less colorful pants. Common colors for pants are blue, black and there is information in the shadow. On the contrary, the variety of colors of the upper body clothing is much larger. In the used data set a broad spectrum of colors with white, black, yellow, blue, green, red, purple, pink, etc. exists.

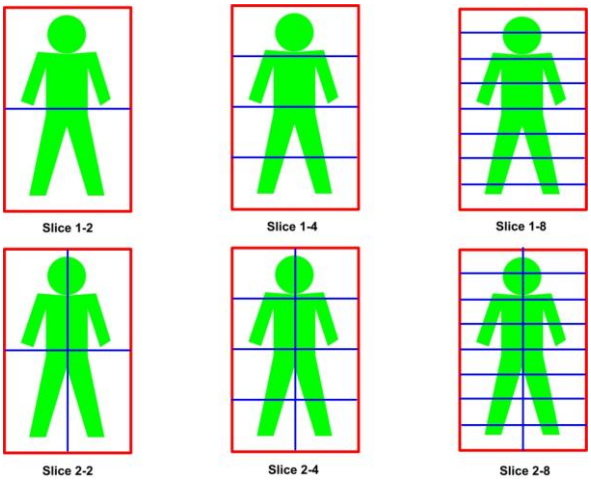


Figure 4-62: The slices used to calculate color histograms.

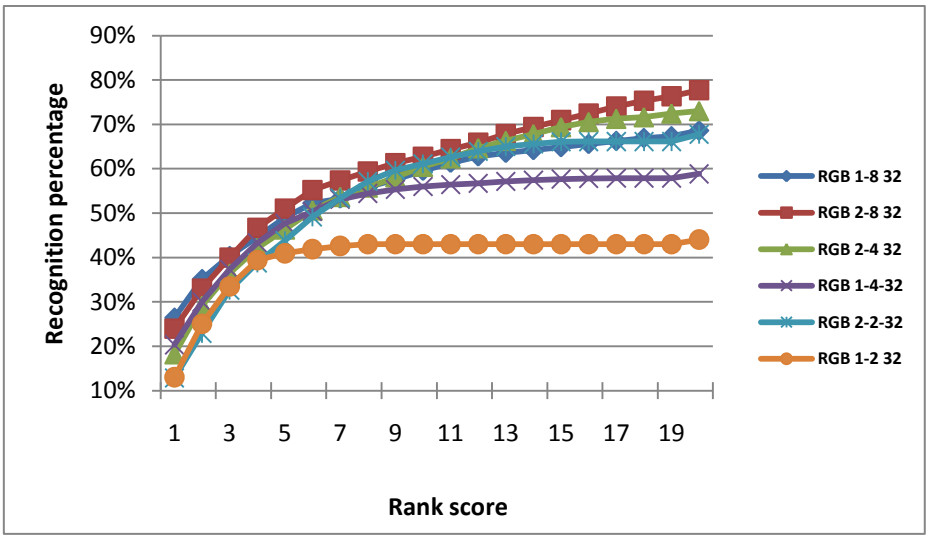


Figure 4-63: Recognition performance with respect to the number of slices

We study the influence of the resolution of histogram i.e. the number of histogram bins, figure 4-64 shows the performance for two color spaces. We note that the increase of the resolution increases the performance for the two color spaces. With more bins, the histogram can distinguish smaller variants of color.

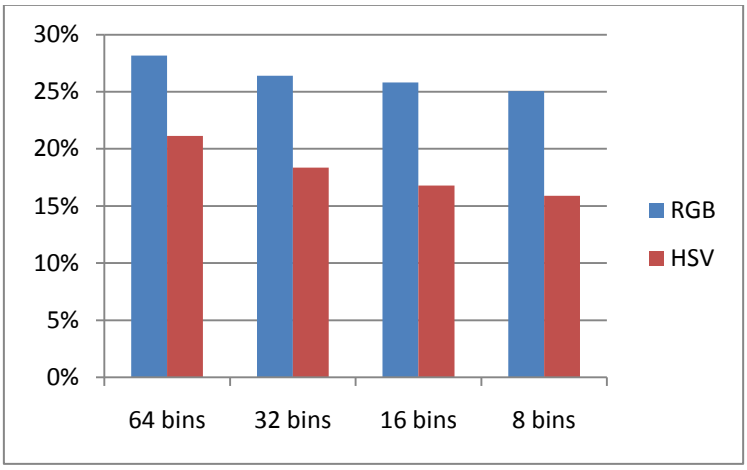


Figure 4-64: Recall as a function of the number of bins of color histograms

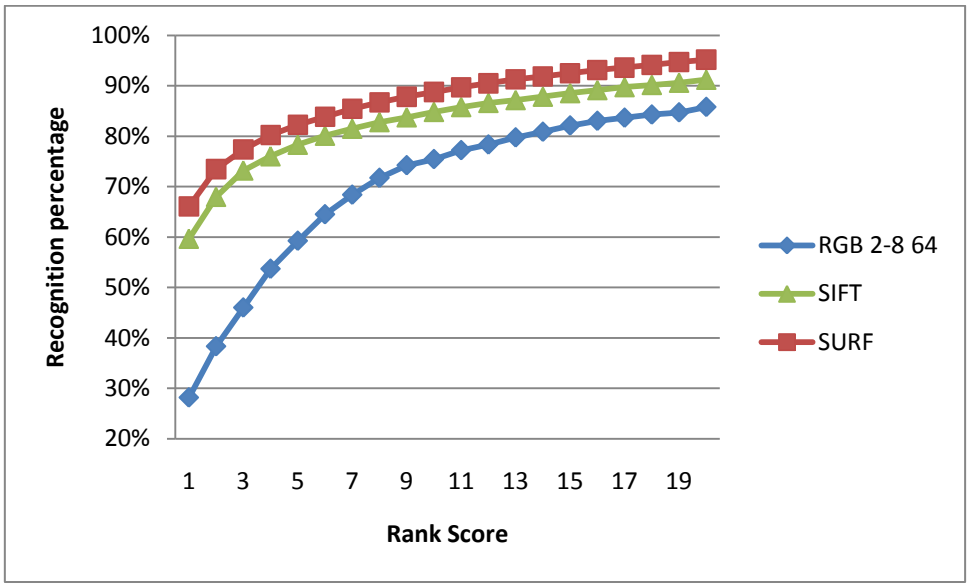


Figure 4-65: Comparison between the performance of local features (SURF, SIFT) and the best performance of color histogram using the same number of image to construct the model and the query.

Experiments showed that the performance of using color features in our case is not good or not enough (figure 4-65). This is coherent with evaluation results presented in (Bak, et al., 2010) or (Alahi, et al., 2010), where they use the color histograms directly but using different data sets. And also get poor performance. In some cases where the resolution of the pedestrian is not high enough to get sufficient number of interest points, the use of color histogram is recommended if the illumination does not change significantly. A perspective of our work could be to fuse multiple features to increase the performance.

4.6.3 Comparison HOG vs. SURF

Another efficient descriptor used commonly in detection is the histogram of oriented gradients (HOG). We divide the pedestrian into small slices then we calculate the HOG for each slice to augment the discernment information of each pedestrian. The calculation of this descriptor is composed of the angle of the gradient for each pixel. We use a filtering of the image in two dimensions:

Horizontal: $G_x = (-1, 0, 1)$.

Vertical: $G_y = (-1, 0, 1)^T$.

The orientation of the gradient is given for each pixel by:

$$\mathrm{O_G} = \arctan\left(\frac{\mathrm{I_x(x,y)}}{\mathrm{I_y(x,y)}}\right) \quad \mathrm{Eq.}\,4-11$$

We also calculate the magnitude of the gradient at each point:

$$\mathrm{mg}(x,y) = \sqrt{I_x^2(x,y) + I_y^2(x,y)}\ \mathrm{Eq.}\,4-12$$

Where $I_x(x,y)$, $I_y(x,y)$ are the results of filtering the Image by G_x, G_y.

We divide each slice into four cells. We then calculate a histogram of oriented gradients for each cell. The angle value $O_G \in [0,360]$ is quantized to N discrete levels. A histogram is formed where each bin is the sum of all magnitudes with the same orientation in a given cell. Each pixel cell participates in the vote. This vote is weighted by the magnitude of the gradient at the location of the pixel. This weighting allows giving more importance to the vote of a pixel belonging to an edge, which then generates an important magnitude, compared to the vote of a pixel belonging to a homogeneous region. The vote allows taking into account the shape of the object in the image. The descriptor is composed of the histograms of four cells. To compare two histograms H_1, H_2, there are many measures which we can use. We used the Bhattacharyya distance as shown in equation $Eq.$ 4-9.

We also study the influence of increasing the number of slides vertically and horizontally. Figure 4-66 illustrates the performance of this increase. We note that increasing the number of slices increases the performance with HOG. But we must also mention that the performance of this descriptor is better than the result of the color histogram. The number of image used to construct the model is 25, and the number of images used to construct the query is 15 images. The number of histogram bins in this experiment is 128.

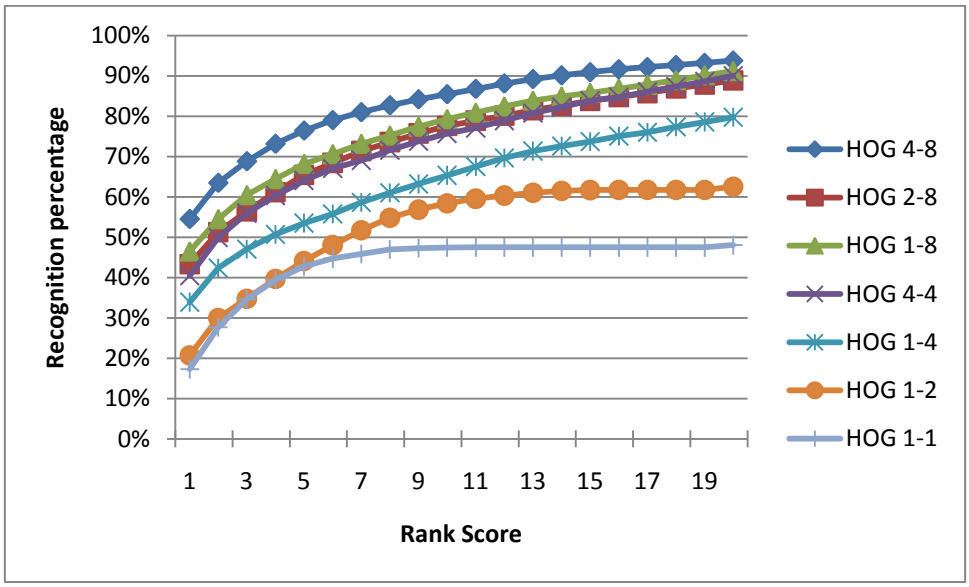


Figure 4-66: Recognition performance with HOG applied on varing number of slices

We evaluate the effect of increasing the number of histogram bins. Figure 4-67 illustrates the influence of this increase. We can note that the performance increases slightly, once we increase the number of bins.

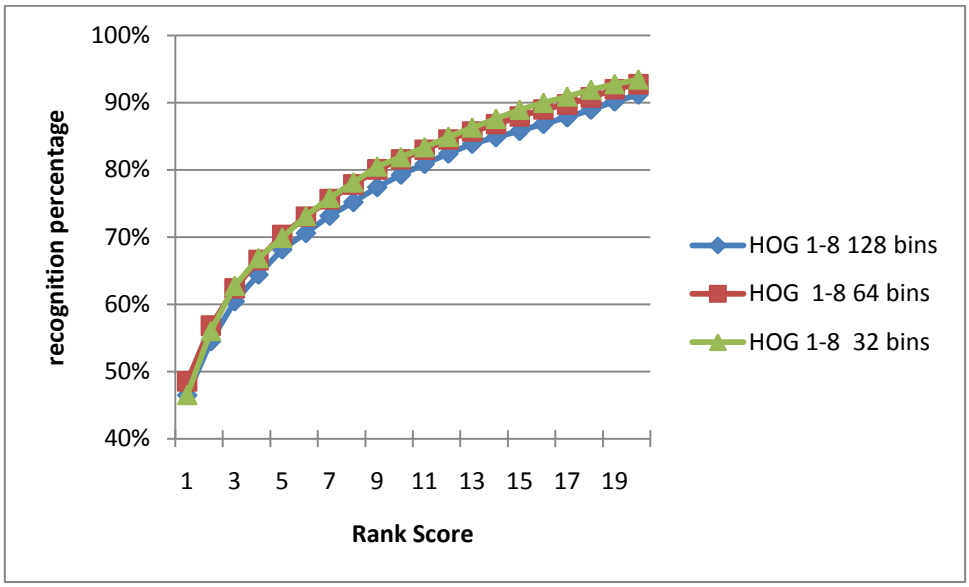


Figure 4-67: Performance with respect to the number of histogram bins.

Finally, we compare the performance of matching using HOG with using either SIFT, SURF, color. We can see in figure 4-68 that the performance of HOG is better than color histogram and comparable with SIFT but the drawback of a global histogram representation is that the histogram depends on the location of the bounding box which contains the pedestrian. Therefore any drifting in location causes a considerable error, as we can see in figure 4-69. It is worth noting that the use of the HOG descriptor in detection does not have the same problem because we scan all images at different scales. For example, Alahi et al in (Alahi, et al., 2010) mix the detection process with the re-identification to avoid this disadvantage.

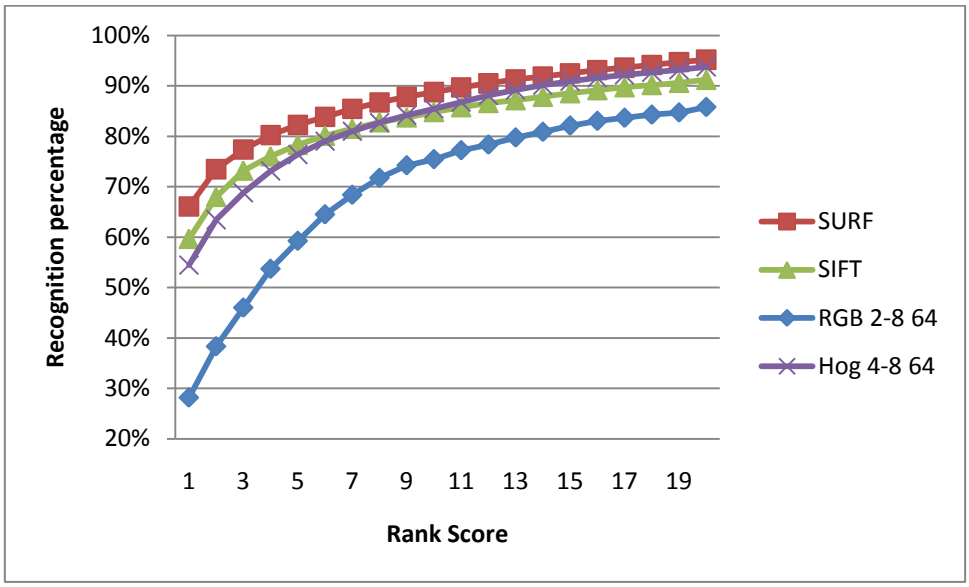


Figure 4-68: Comparison between the performance of local features (SURF, SIFT) and the best performance of HOG using the same number of image to construct the model (M=25) and the query (N=15).

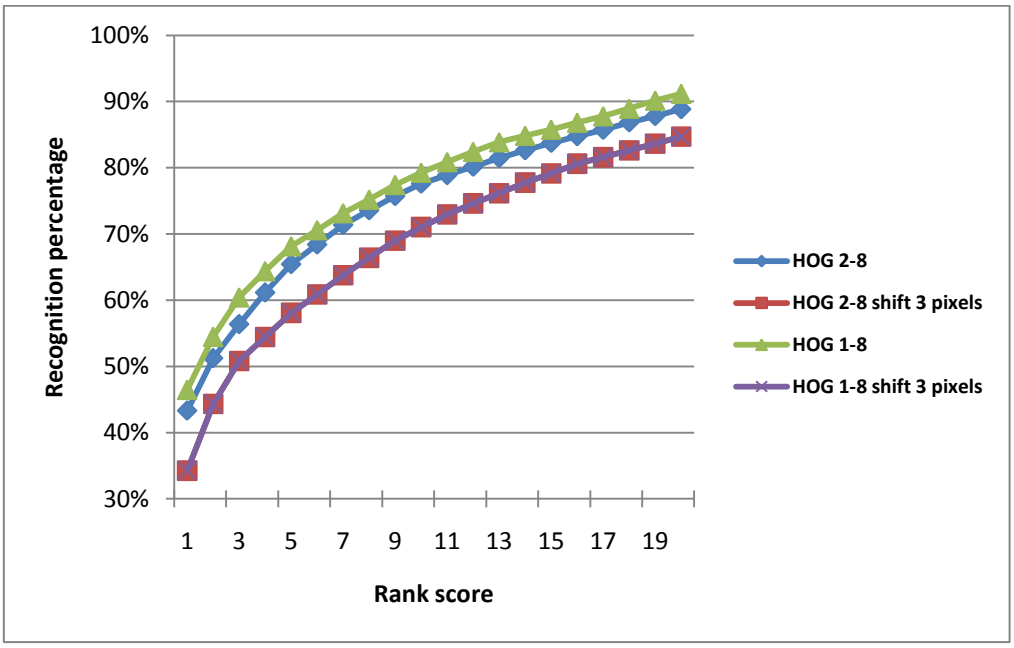


Figure 4-69: the influence of drifting of bounding box by only 3 pixels

To generalize the idea of the probabilistic model, we perform this method using HOG and RGB Global features. Figure 4-70 shows the improvement of result when we use the probabilistic model. We note that, as with SURF and SIFT, the performance of re-identification is increased using the probabilistic model (15% for RGB and 11% for Hog)

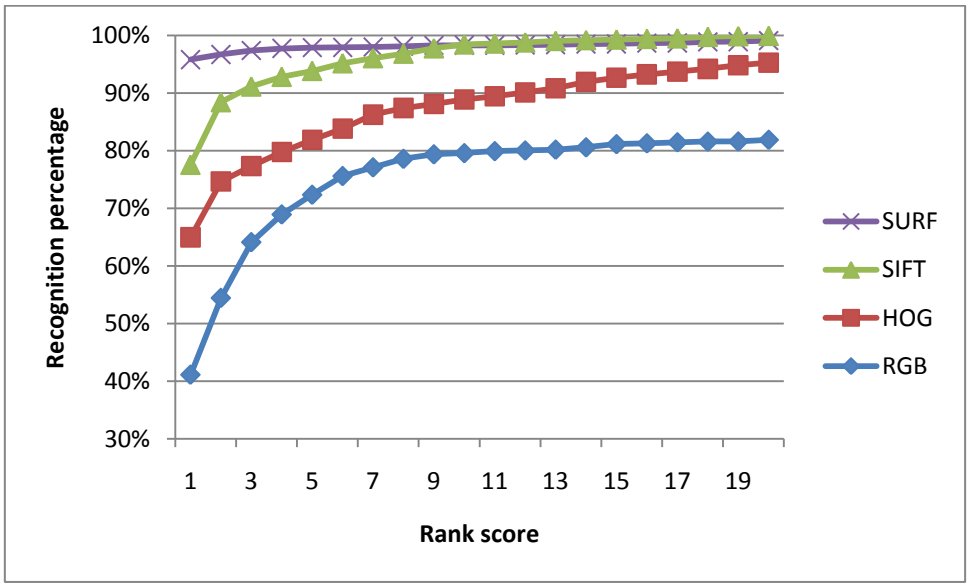


Figure 4-70: Performance of recognition using probabilistic model

4.7 Comparison KD-tree vs. hierarchical k-means tree

Another evaluation of our algorithm is done. We compared KD-tree with hierarchical k-means tree which are commonly used in nearest-neighbors algorithms (see §4.1.4). We evaluate its performance by using the ETHZ dataset (First sequence which contains 83 pedestrians). We adopt the Flann library which was provided by Muja et al (Muja, et al., 2009).We substitute hierarchical K-means tree to KD-tree, we use SURF features. The cumulative match curves (CMC) are shown in figure 4-71 where we can see that the difference between the two is negligible.

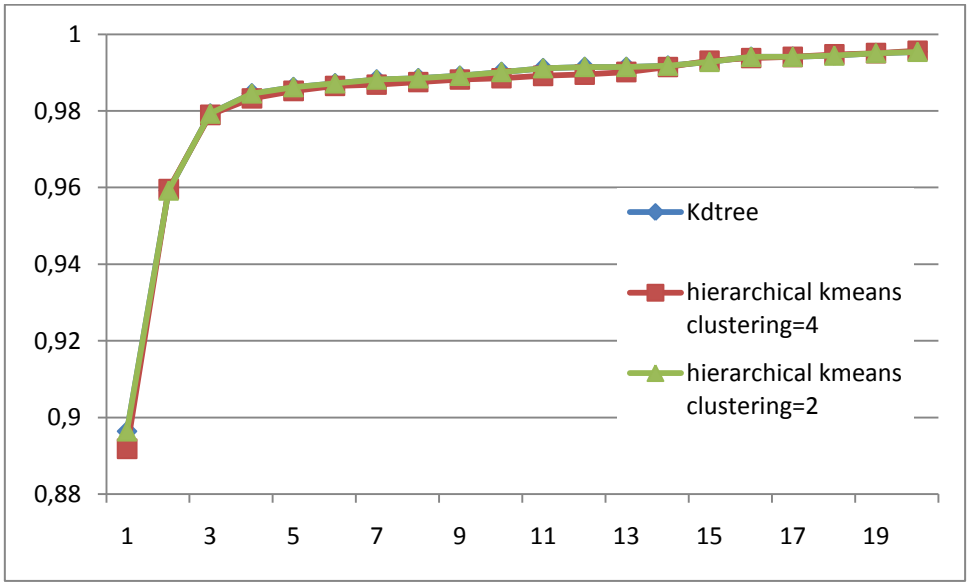


Figure 4-71: Comparison between the performance of KD-tree and the performance of hierarchical k-means.

On one hand the Kd-tree is fast to build and non parametric. In the other hand the number of keypoints correct match using hierarchical K-means is slightly higher than when using KD-tree (the difference is 4%). But the location of these points could be in the background, so this augmentation does not impact the performance at pedestrian level. The performance of hierarchical K-means depends on the *number of clusters used in the K-means algorithm, as well as on the number of iterations chosen. However* the performance of the nearest-neighbors algorithms depends on the dataset used (Muja, et al., 2009), so it may be better use the structure proposed by (Junejo, et al., 2008) which adapted the number of branches of tree using bags of words.

4.8 Conclusions

We have presented a new person re-identification approach based on matching of interest-points collected from a short query video sequence with those collected from longer model videos used for each previously seen person. Our experiments on the CAVIAR data set have shown a good inter-camera person re-identification performance (a precision of 82% for a recall of 78%).

We have also conducted a deeper evaluation on a larger corpus specially built by us for the inter-camera re-identification performance evaluation. This new evaluation of our method confirms its good inter-camera person re-identification performance: a precision of 85% and a recall of 76% even on a corpus of 40 persons.

However a drawback that we see in this approach is that there are many parameters we must adjust. The influence of these parameters is large on model. In addition, this approach does not take into account the spatial and temporal information. Therefore we propose to solve these problems using a probabilistic model which integrates the construction of the model in the tracking framework. This integration allows to choose the discriminative interest points of the model. Then we give these points a weight depending on their frequency in their model and other models. This improved variant of our method has much higher performance: 95% recall and precision on our 40 persons dataset. This generic approach was also tested using SIFT interest points (instead of SURF), with same large performance improvement.

We also evaluate our approach on 146 pedestrians (ETHZ data set). The performance is 96% first rank recognition using 5 images to construct the model and the query. This clearly outperforms published results of other methods on same dataset.

It should be noted that our matching method is very fast, with a typical computation time of 1/30s for the re-identification of one target person among 40 stored signatures of previously seen persons in other cameras (22220 interest points) using (Intel Core i7 CPU, 2.93GHz, RAM4GB). Moreover, this re-identification time scales logarithmically with the number of stored person models.

We compare also the performance of using the simple global features approach with the local approach. We find in our case that the local approach (SURF, and also SIFT), outperforms the global approach because the data set containing the pedestrians has a sufficient resolution, which permits to distinguish the pedestrians. There is a significant difference in illumination between the two sequences which makes it difficult to use the color features.

A general conclusion drawn from all the evaluations is that the re-identification result depends mostly how well the model represent the pedestrian. The use of machine learning algorithms to find the discriminative information is a possible solution. Yet a potential problem is that most statistical classifiers cannot be incrementally adapted when adding new classes, and retraining with the whole database may take significant time. On the contrary, with our approach, adding a new pedestrian is done by comparing his keypoints to the interest points of all the pedestrians in kd-tree, and directly changing the weights which match. Our method is therefore more suitable for real-time applications of re-identification.

5 Conclusion & Perspective

5.1 Summary

The objective of this thesis was to build a reliable and efficient system of re-identification and tracking of pedestrians for intelligent multi-sensor video surveillance. We have proposed and evaluated a person re-identification scheme using the matching of interest points collected in several images during short video sequences. The central point of our algorithm lies in the exploitation of image sequences. This allows to get a more "dynamic" and multi-view descriptor than when using a single image, and is slightly similar to the "averaging of interest-point descriptors throughout the time sequence" used in the work by Sivic and Zisserman in (Sivic, et al., 2003). However, contrary to them, we do not use the SIFT detector and descriptor, but a locally-developed and particularly efficient variant of SURF, Camellia Keypoints. This is also differs from with Gheissari et al. who use a color-histogram of the region around interest points for their matching in (Gheissari, et al., 2006).

All the experiments were carried out on real world datasets. The first experiment was done using the CAVIAR dataset. We chose the sequences of ten persons taken by two cameras. The performance of re-identification of our algorithm is comparable to the state of the art (BAK, et al., 2010) and (Oliveira, et al., 2009) , with approximately 80% of correct identification with a high confidence level.

We constructed a new corpus for the re-identification evaluation, which presently contains 40 different pedestrians recorded by 2 non-overlapping cameras to perform a deeper evaluation. This evaluation confirms our method's good inter-camera person re-identification performances: 76% of correct identification with our first method. We have proposed an improvement, which involves the automatic selection of moments or interest points, to obtain a set of points for each individual, which are both the most varying, and more discriminating against those of other people. In addition this improvement reduces the redundant information, and use the temporal vote of the query to improve the performance. With these modifications we obtain a 20% improvement of performance, which climbs to 95% first rank recognition using SURF on our 40 persons dataset.

We also evaluate our approach on 146 pedestrians (ETHZ data set). The performance is 96% first rank recognition using 5 images to construct the model and the query. This clearly outperforms published results of other methods on same dataset. Although this data set is less difficult in comparison with the other two datasets (CAVIAR and our dataset), this evaluation also indicates that our approach works even for more than one hundred of people.

Our matching method is also very fast: ~ 1/30s for the re-identification of one target person among 40 previously seen persons. The computation time is logarithmic to the number of stored person models, making the re-identification among hundreds of persons computationally feasible in less than ~ 1/5s second.

The adaptive background substraction algorithm presented in Chapter 3 is not perfect. It has the advantage of being generic. This means that it does not require any special preprocessing for a given installation. However, this algorithm needs to be combined with other algorithms (visual object

detection) to achieve a good accuracy. We use the Adaboost for selecting the interest points of a pedestrian. Our experiments have shown a good detection performance (a precision of 90% for a recall of 87%) when we use only the adaboost classification, further augmented when we use the cascade, to a precision of 93% for a recall of 90%. Another advantage of using the cascade is that the number of the points which need to be classified by Adaboost decreases to 10 fold, when we use background substraction. The performance of our keypoints-based method is comparable to state-of-Art method such as Hog, but is attained at lower computing cost, and also has the advantage of using the same primitive (keypoints) as we use for re-identification.

5.2 Future work.

First, we shall put online, for use by the research community, our re-identification benchmark (after blurring faces for respecting privacy laws).

Also, a variety of possible enhancements exists, which could further improve results. In particular, the following ideas can be mentioned:

- We shall try to integrate our re-identification scheme with our background modelling and foreground extraction using interest points, which would allow to restrict the search area to the silhouette of the person, excluding most background key-points. This should significantly improve the performance of our system.
- The shortcoming of local features is that when the resolution of the pedestrian in not high enough to get sufficient number of interest points, the use of global features is recommended. A possible perspective of our work is to fuse multiple features to increase the performance, in particular by incorporating global signature that could relay keypoints when image resolution gets too low.
- The biometrics features should be taken in consideration to re-identify people wearing similar color clothes, especially when there are several hundreds of people.
- The usage of smart cameras as distributed agents imposes new architecture and new problematic related to the cooperation and the computing distribution as well as the fusion of information from multiple cameras. The implementation of keypoints extracting on these cameras is a special task; the onboard embedded processors have generally special architectures. The mapping of the algorithms to smart cameras could be guided by a co-design activity which aims to develop new cores for accelerating computer vision tasks using special hardware. Hence the camera behaves like an application sensor that just transmits the results to ensure privacy.

6 Publications

Conférences internationales avec comité de relecture

- "*Keypoints-based background model and foreground pedestrians extraction for future smart cameras*", Omar Hamdoun and Fabien Moutarde, proceedings of 3rd ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2009), Como, Italy, August 30 - September 3, 2009.

- "*Interest points harvesting in video sequences for efficient person identification", Omar Hamdoun*, Fabien Moutarde, Bogdan Stanciulescu and Bruno Steux, proceedings of '8th international workshop on Visual Surveillance (VS2008)' of "10th European Conference on Computer Vision (ECCV'2008)", Marseille, France, October 17th, 2008.

- "*Person Re-identification in Multi-camera System by Signature based on Interest Point Descriptors Collected on Short Video Sequences*" Omar Hamdoun, Fabien Moutarde, Bogdan Stanciulescu and Bruno Steux, proceedings of ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC-08), Stanford University, California, USA, September 7-11, 2008.

Conférences nationales, et autres communications

- "*Vidéosurveillance intelligente : ré-identification de personnes par signature utilisant des descripteurs de points d'intérêt collectés sur des séquences*" Omar Hamdoun, Fabien Moutarde, Bogdan Stanciulescu and Bruno Steux, proceedings of Workshop on «Surveillance, Sûreté, Sécurité des Grands Systèmes » (3SGS'08), Troyes, 4-5 juin 2008.

- « *Ré-identification de personnes entre caméras par comparaison de descripteurs de points d'intérêts collectés sur des séquences* », Omar Hamdoun et Fabien Moutarde, présentation à la journée « vidéosurveillance intelligente » du Gdr ISIS (Information, Signal, Images et ViSion), Paris, 17 décembre 2008.

7 Bibliography

Abbasi, S., Mokhtarian, F. and Kittler, J. "Curvature scale space image in shape similarity retrieval," *Multimedia Systems* (7), 1999, pp. 467-476.

Abdel-Hakim, A. and Farag, A. "CSIFT: A SIFT Descriptor with Color Invariant Characteristics", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006* 2, 2006, pp. 1978 - 1983.

Adam, S., Ogier, J., Cariou, C., Mullot, R., Gardes, J. and Lecourtier, Y. "Using the Fourier Mellin transform for multi-oriented and multi-scaled patterns recognition: application to automatic analysis of technical documents," *TS. Traitement du signal* (18), 2001, pp. 17--33.

Alahi, A., Vandergheynst, P., Bierlaire, M. and Kunt, M. "Cascade of descriptors to detect and track objects across any network of cameras," *Computer Vision and Image Understanding* (114), 2010, pp. 624--640.

Albu, A., Laurendeau, D., Comtois, S., Ouellet, D., Hebert, P., Zaccarin, A., Parizeau, M., Bergevin, R., Maldague, X., Drouin, R., Drouin, S., Martel-Brisson, N., Jean, F., Torresan, H., Gagnon, L. and Laliberte, F. "MONNET: Monitoring Pedestrians with a Network of Loosely-Coupled Cameras", *18th International Conference on Pattern Recognition, 2006. ICPR 2006.* 4, 2006, pp. 924 -928.

Arth, C., Leistner, C. and Bischof, H. "Object Reacquisition and Tracking in Large-Scale Smart Camera Networks", *First ACM/IEEE International Conference on Distributed Smart Cameras, 2007. ICDSC '07.* , 2007, pp. 156 -163.

Arya, S. and Mount, D. "Algorithms for fast vector quantization", *Data Compression Conference, 1993. DCC '93.* , 1993, pp. 381 -390.

Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R. and Wu, A. Y. "An optimal algorithm for approximate nearest neighbor searching in fixed dimensions", (Techreport) University of Maryland at College Park, 1995.

Bak, S., Corvee, E., Bremond, F. and Thonnat, M. "Person Re-identification Using Spatial Covariance Regions of Human Body Parts", *Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2010* , 2010, pp. 435 -440.

Bak, S., Corvee, E., Bremond, F. and Thonnat, M. "Person Re-identification Using Haar-based and DCD-based Signature", *Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2010* , 2010, pp. 1 -8.

Barron, J., Fleet, D., Beauchemin, S. and Burkitt, T. "Performance of optical flow techniques", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992* , 1992, pp. 236 -242.

Bauer, J., Sunderhauf, N. and Protzel, P. "Comparing several implementations of two recently published feature detectors" International Conference on Intelligent and Autonomous

Systems, IAV', 2007.

Bay, H., Ess, A., Tuytelaars, T. and Gool, L. V. "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding* (110:3), 2008, pp. 346 - 359.

Bdiri, T., Moutarde, F. and Steux, B. "Visual object categorization with new keypoint-based adaBoost features", *IEEE Intelligent Vehicles Symposium, 2009* , 2009, pp. 393 -398.

Beis, J. and Lowe, D. "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997. Proceedings.*, 1997, pp. 1000 -1006.

Belhumeur, P., Hespanha, J. and Kriegman, D. "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* (19), 1997, pp. 711--720.

Bramberger, M.; Doblander, A.; Maier, A.; Rinner, B.; Schwabach, H.*;* , "Distributed embedded smart cameras for surveillance applications," *Computer , vol.39, no.2, pp. 68- 75, Feb. 2006.*

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. *Wadsworth & Brooks/Cole Advanced Book and Software, Pacific Grove, CA.*

Cai, Q. and Aggarwal, J. "Tracking human motion in structured environments using a distributed-camera system," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* (21:11), 1999, pp. 1241 -1247.

Calderara, S., Cucchiara, R. and Prati, A. "Bayesian-Competitive Consistent Labeling for People Surveillance," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* (30:2), 2008, pp. 354 -360.

van de Camp, F., Bernardin, K. and Stiefelhagen, R. "Person tracking in camera networks using graph-based bayesian inference", *Third ACM/IEEE International Conference on Distributed Smart Cameras, 2009. ICDSC 2009.* , 2009, pp. 1 -8.

Chang, T.-H. and Gong, S. "Tracking multiple people with a multi-camera system", *IEEE Workshop on Multi-Object Tracking, 2001. Proceedings. 2001* , 2001, pp. 19 -26.

Chellappa, R., Roy-Chowdhury, A. and Kale, A. "Human Identification using Gait and Face", *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07.* , 2007, pp. 1 -2.

Chen, Y., Liang, G., Lee, K. K. and Xu, Y. "Abnormal Behavior Detection by Multi-SVM-Based Bayesian Network", *International Conference on Information Acquisition, 2007. ICIA '07.* , 2007, pp. 298 -303.

Collins, R., Lipton, A., Fujiyoshi, H. and Kanade, T. "Algorithms for cooperative multisensor surveillance," *Proceedings of the IEEE* (89:10), 2001, pp. 1456 -1477.

Collins, R., Lipton, A., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., Tolliver, D., Enomoto, N., Hasegawa, O., Burt, P. and others "A system for video surveillance and monitoring: VSAM final report," *Robotics Inst., CMU-RI-TR-00-12* (), 2000.

Cong, D., Achard, C., Khoudour, L. and Douadi, L. "Video Sequences Association for People Re-identification across Multiple Non-overlapping Cameras", *in 'Image Analysis and Processing – ICIAP 2009'*, Springer Berlin / Heidelberg, 2009, pp. 179-189.

Crowley, J. L. and Parker, A. C. "A Representation for Shape Based on Peaks and Ridges in the Difference of Low-Pass Transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* (PAMI-6:2), 1984, pp. 156 -170.

Dalal, N. and Triggs, B. "Histograms of oriented gradients for human detection", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.* 1, 2005, pp. 886 -893 vol. 1.

Deng, Y. and Manjunath, B. "Unsupervised segmentation of color-texture regions in images and video," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* (23:8), 2001, pp. 800 -810.

Dixon, M., Jacobs, N. and Pless, R. "An efficient system for vehicle tracking in multi-camera networks", *Third ACM/IEEE International Conference on Distributed Smart Cameras, 2009. ICDSC 2009.* , 2009, pp. 1 -8.

Dockstader, S. and Tekalp, A. "Multiple camera fusion for multi-object tracking", *IEEE Workshop on Multi-Object Tracking, 2001. Proceedings. 2001* , 2001, pp. 95 -102.

Draper, B. A., Baek, K., Bartlett, M. S. and Beveridge, J. R. "Recognizing faces with PCA and ICA," *Computer Vision and Image Understanding* (91:1-2), 2003, pp. 115 - 137.

Dufournaud, Y., Schmid, C. and Horaud, R. "Matching images with different resolutions", *IEEE Conference on Computer Vision and Pattern Recognition, 2000. Proceedings.* 1, 2000, pp. 612 -618 vol.1.

Elgammal, A., Harwood, D. and Davis, L. "Non-parametric Model for Background Subtraction", *in* Vernon, D., ed.,'Computer Vision — ECCV 2000', Springer Berlin / Heidelberg, 10.1007/3-540-45053-X_48, 2000, pp. 751-767.

Eng, H.-L., Toh, K.-A., Kam, A., Wang, J. and Yau, W.-Y. "An automatic drowning detection surveillance system for challenging outdoor pool environments", *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings.* , 2003, pp. 532 -539 vol.1.

Farenzena, M., Bazzani, L., Perina, A., Murino, V. and Cristani, M. "Person re-identification by symmetry-driven accumulation of local features", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010* , 2010, pp. 2360 -2367.

Finlayson, G., Hordley, S., Schaefer, G. and Tian, G. Y. "Illuminant and device invariant colour using histogram equalisation," *Pattern Recognition* (38:2), 2005, pp. 179 - 190.

Florack, L., Ter Haar Romeny, B., Viergever, M. and Koenderink, J. "The Gaussian scale-space paradigm and the multiscale local jet," *International Journal of Computer Vision* (18), 1996, pp. 61-75.

Flusser, J. and Suk, T. "Pattern recognition by affine moment invariants," *Pattern Recognition* (26:1), 1993, pp. 167 - 174.

Freund, Y. and Schapire, R. "A desicion-theoretic generalization of on-line learning and an application to boosting", *in* Vitányi, P., ed.,'Computational Learning Theory', Springer Berlin / Heidelberg, 10.1007/3-540-59119-2_166, 1995, pp. 23-37.

Friedman, J. H., Bentley, J. L. and Finkel, R. A. "An Algorithm for Finding Best Matches in Logarithmic Expected Time," *ACM Trans. Math. Softw.* (3:3), 1977, pp. 209--226.

Gandhi, T. and Trivedi, M. "Pedestrian Protection Systems: Issues, Survey, and Challenges," *Intelligent Transportation Systems, IEEE Transactions on* (8:3), 2007, pp. 413 - 430.

Geusebroek, J.-M., van den Boomgaard, R., Smeulders, A. and Geerts, H. "Color invariance," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* (23:12), 2001, pp. 1338 -1350.

Gheissari, N., Sebastian, T. and Hartley, R. "Person Reidentification Using Spatiotemporal Appearance", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006* 2, 2006, pp. 1528 - 1535.

Gilbert, A. and Bowden, R. "Tracking Objects Across Cameras by Incrementally Learning Inter-camera Colour Calibration and Patterns of Activity", *in* Leonardis, A., Bischof, H. and Pinz, A., ed.,*'Computer Vision – ECCV 2006'*, Springer Berlin / Heidelberg, 10.1007/11744047_10, 2006, pp. 125-136.

Gouaillier, V. and Fleurant, A. "Intelligent Video Surveillance: Promises and Challenges," IEEE , 2009.

Gray, D., Brennan, S. and Tao, H. "Evaluating Appearance Models for Recognition, Reacquisition, and Tracking"'*10th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)'*, 2007.

Gray, D. and Tao, H. "Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features"*'Computer Vision – ECCV 2008'*, 2008, pp. 262--275.

Haritaoglu, I., Harwood, D. and Davis, L. "W4: real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* (22:8), 2000, pp. 809 -830.

Harris, C. and Stephens, M. "A Combined Corner and Edge Detection"' *Proceedings of The Fourth Alvey Vision Conference'*, 1988, pp. 147--151.

Hartley, R. and Zisserman, A. *Multiple View Geometry in Computer Vision*, Cambridge

University Press, 2004.

Heikkila, M. and Pietikainen, M. "A texture-based method for modeling the background and detecting moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* (28:4), 2006, pp. 657 -662.

Hu, M.-K. "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory,* (8:2), 1962, pp. 179 -187.

Hu, W., Tan, T., Wang, L. and Maybank, S. "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* (34:3), 2004, pp. 334 -352.

Hu, W., Xie, D., Fu, Z., Zeng, W. and Maybank, S. "Semantic-Based Surveillance Video Retrieval," *IEEE Transactions on Image Processing,* (16:4), 2007, pp. 1168 -1181.

Huang, J., Ravi Kumar, S., Mitra, M., Zhu, W.-J. and Zabih, R. "Spatial Color Indexing and Applications," *International Journal of Computer Vision* (35), 1999, pp. 245-268.

Huang, T. and Russell, S. "Object identification in a Bayesian context" *'IJCAI'97: Proceedings of the Fifteenth international joint conference on Artifical intelligence'*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997, pp. 1276--1282.

Huttenlocher, D., Klanderman, G. and Rucklidge, W. "Comparing images using the Hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* (15:9), 1993, pp. 850 -863.

Jain, R. and Wakimoto, K. "Multiple perspective interactive video", *Proceedings of the International Conference onMultimedia Computing and Systems, 1995,* , 1995, pp. 202 -211.

Javed, O., Rasheed, Z., Shafique, K. and Shah, M. "Tracking across multiple cameras with disjoint views", *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings.* , 2003, pp. 952 -957 vol.2.

Javed, O., Shafique, K. and Shah, M. "Appearance modeling for tracking in multiple non-overlapping cameras", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.* 2, 2005, pp. 26 - 33 vol. 2.

Javed, O., Shafique, K. and Shah, M. "A hierarchical approach to robust background subtraction using color and gradient information", *Motion and Video Computing, 2002. Proceedings. Workshop on* , 2002, pp. 22 - 27.

Jaynes, C. "Multi-view calibration from planar motion trajectories," *Image and Vision Computing* (22:7), 2004, pp. 535 - 550.

Jegou, H., Douze, M. and Schmid, C. "Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search", *in the European Conference on Computer Vision – ECCV 2008*, , 2008, pp. 304-317.

Junejo, I. N. and Foroosh, H. "Euclidean path modeling for video surveillance," *Image and Vision Computing* (26:4), 2008, pp. 512 - 528.

Kadir, T. and Brady, M. "Saliency, Scale and Image Description," *International Journal of Computer Vision* (45), 2001, pp. 83-105.

Kayumbi, G., Mazzeo, P. L., Spagnolo, P., Taj, M. and Cavallaro, A. "Distributed visual sensing for virtual top-view trajectory generation in football videos" *'CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval'*, ACM, New York, NY, USA, 2008, pp. 535--542.

Ke, Y. and Sukthankar, R. "PCA-SIFT: a more distinctive representation for local image descriptors", *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* 2, 2004, pp. II-506 - II-513 Vol.2.

Kettnaker, V. and Zabih, R. "Bayesian multi-camera surveillance", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999.* 2, 1999, pp. 259 Vol. 2.

Khan, S. and Shah, M. "A Multiview Approach to Tracking People in Crowded Scenes Using a Planar Homography Constraint" *'Computer Vision – ECCV 2006'*, 2006, pp. 146, 133.

Khan, S. and Shah, M. "Consistent labeling of tracked objects in multiple cameras with overlapping fields of view" *IEEE Transactions on Pattern Analysis and Machine Intelligence,* (25:10), 2003, pp. 1355 - 1360.

Khotanzad, A. and Hong, Y. "Invariant image recognition by Zernike moments" '*IEEE Transactions on Pattern Analysis and Machine Intelligence'*, (12:5), 1990, pp. 489 -497.

Kim, K., Chalidabhongse, T. H., Harwood, D. and Davis, L. "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging* (11:3), 2005, pp. 172 - 185.

Kim, K. and Davis, L. "Multi-camera Tracking and Segmentation of Occluded People on Ground Plane Using Search-Guided Particle Filtering", *in* Leonardis, A., Bischof, H. and Pinz, A., ed.,*'Computer Vision – ECCV 2006'*, pp. 98-109.

Ko, T. "A survey on behavior analysis in video surveillance for homeland security applications", *37th IEEE Applied Imagery Pattern Recognition Workshop, 2008. AIPR '08.* , 2008, pp. 1 -8.

Kovacs-Vajna, Z. "A fingerprint verification system based on triangular matching and dynamic time warping," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* (22:11), 2000, pp. 1266 - 1276.

Lantagne, M., Parizeau, M. and Bergevin, R. " VIP: Vision tool for comparing Images of People ," *Vision Interface* , 2003 .

Lee, L., Romano, R. and Stein, G. "Monitoring activities from multiple video streams:

establishing a common coordinate frame," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* (22:8), 2000, pp. 758 -767.

Levi, K. and Weiss, Y. "Learning object detection from a small number of examples: the importance of good features", *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on* 2, 2004, pp. II-53 - II-60 Vol.2.

Lindeberg, T. "Feature Detection with Automatic Scale Selection," *International Journal of Computer Vision* (30), 1998, pp. 79-116.

Lipton, A., Fujiyoshi, H. and Patil, R. "Moving target classification and tracking from real-time video", *Fourth IEEE Workshop on Applications of Computer Vision, 1998. WACV '98.* , 1998, pp. 8 -14.

Lowe, D. "Local feature view clustering for 3D object recognition", *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001.* 1, 2001, pp. I-682 - I-688 vol.1.

Lowe, D. "Object recognition from local scale-invariant features", *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.* 2, 1999, pp. 1150 -1157 vol.2.

Lowe, D. G. "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision* (60), 2004, pp. 91-110.

Madden, C., Cheng, E. and Piccardi, M. "Tracking people across disjoint camera views by an illumination-tolerant appearance representation," *Machine Vision and Applications* (18), 2007, pp. 233-247.

Makris, D., Ellis, T. and Black, J. "Bridging the gaps between cameras", *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* 2, 2004, pp. II-205 - II-210 Vol.2.

Matas, J., Chum, O., Urban, M. and Pajdla, T. "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing* (22:10), 2004, pp. 761 - 767.

McIvor, A. "Background subtraction techniques" *'Proc. of Image and Vision Computing, Auckland, New Zealand'*, Citeseer, 2000.

Meyer, D., Pˆsl, J. and Niemann, H. "Gait classification with HMMs for trajectories of body parts extracted by mixture densities" *'British Machine Vision Conference'*, 1998, pp. 459--468.

Mikolajczyk, K. and Schmid, C. "A performance evaluation of local descriptors" *IEEE Transactions on Pattern Analysis and Machine Intelligence,* (27:10), 2005, pp. 1615 -1630.

Mikolajczyk, K. and Schmid, C. "Scale & Affine Invariant Interest Point Detectors"

International Journal of Computer Vision (60), 2004, pp. 63-86.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T. and Gool, L. "A Comparison of Affine Region Detectors," *International Journal of Computer Vision* (65), 2005, pp. 43-72.

Mindru, F., Moons, T. and Gool, L. V. "Color-Based Moment Invariants For Viewpoint And Illumination Independent Recognition Of Planar Color Patterns"'Illumination Independent Recognition of Planar Color Patterns", *Proceedings ICAPR'98'*, 1998, pp. 113--122.

Mindru, F., Tuytelaars, T., Gool, L. V. and Moons, T. "Moment invariants for recognition under changing viewpoint and illumination" *Computer Vision and Image Understanding* (94:1-3), 2004, pp. 3 - 27.

Mittal, A. and Davis, L. S. " Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene," *International Journal of Computer Vision* (51), 2003, pp. 189-203.

Moeslund, T. B. and Granum, E. "A Survey of Computer Vision-Based Human Motion Capture," *Computer Vision and Image Understanding* (81:3), 2001, pp. 231 - 268.

Morevec, H. P. "Towards automatic visual obstacle avoidance" *'IJCAI'77: Proceedings of the 5th international joint conference on Artificial intelligence'*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1977, pp. 584--584.

Muja, M. and Lowe, D. G. "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration» *'International Conference on Computer Vision Theory and Application VISSAPP'09'*, INSTICC Press, 2009, pp. 331-340.

Nakajima, C., Pontil, M., Heisele, B. and Poggio, T. "Full-body person recognition system," *Pattern Recognition* (36:9), 2003, pp. 1997 - 2006.

Nakashima, H., Aghajan, H. and Augusto, J. C. *Handbook of Ambient Intelligence and Smart Environments*, Springer US, 2010.

Newsam, S. D. and Kamath, C. "Comparing shape and texture features for pattern recognition in simulation data" *'Image Processing: Algorithms and Systems'*, 2006, pp. 106-117.

Ning, H., Tan, T., Wang, L. and Hu, W. "People tracking based on motion model and motion constraints with automatic initialization," *Pattern Recognition* (37:7), 2004, pp. 1423 - 1440.

Nister, D. and Stewenius, H. "Scalable Recognition with a Vocabulary Tree", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006* 2, 2006, pp. 2161 - 2168.

Nistér, D. and Stewénius, H. "Linear Time Maximally Stable Extremal Regions", *in*

Forsyth, D., Torr, P. and Zisserman, A., ed., *'Computer Vision – ECCV 2008'*, Springer Berlin / Heidelberg, 2008, pp. 183-196.

de Oliveira, I. and de Souza Pio, J. "People Reidentification in a Camera Network", *Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, 2009. DASC '09.* , 2009, pp. 461 -466.

Oliver, N., Rosario, B. and Pentland, A. "A Bayesian computer vision system for modeling human interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* (22:8), 2000, pp. 831 -843.

Orten, B., Soysal, M. and Alatan, A. "Person identification in surveillance video by combining MPEG-7 experts", *Signal Processing and Communications Applications Conference, 2005. Proceedings of the IEEE 13th* , 2005, pp. 352 - 355.

Orwell, J., Remagnino, P. M. and Jones, G. A. "Optimal Color Quantization for Real-Time Object Recognition," *Real-Time Imaging* (7:5), 2001, pp. 401 - 414.

Park, U., Jain, A., Kitahara, I., Kogure, K. and Hagita, N. "ViSE: Visual Search Engine Using Multiple Networked Cameras", *18th International Conference on Pattern Recognition, 2006. ICPR 2006.* 3, 2006, pp. 1204 -1207.

Pham, T. and Smeulders, A. "Efficient projection pursuit density estimation for background substractions"'Proc. *IEEE Intern Workshop on Visual Surveillance'*, 2006.

Pham, T., Worring, M. and Smeulders, A. "A Multi-Camera Visual Surveillance System for Tracking of Reoccurrences of People", *First ACM/IEEE International Conference on Distributed Smart Cameras, 2007. ICDSC '07.* , 2007, pp. 164 -169.

Poppe, R. "A survey on vision-based human action recognition," *Image and Vision Computing* (28:6), 2010, pp. 976 - 990.

Porikli, F. "Inter-camera color calibration by correlation model function", *International Conference on Image Processing, 2003. ICIP 2003. Proceedings.* 2, 2003, pp. II - 133-6 vol.3.

Prosser, B., Gong, S. and Xiang, T. "Multi-camera Matching under Illumination Change Over Time" *'Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications'*, Andrea Cavallaro and Hamid Aghajan , Marseille France , 2008.

Rahimi, A., Dunagan, B. and Darrell, T. "Simultaneous calibration and tracking with a network of non-overlapping sensors", *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* 1, 2004, pp. I-187 - I-194 Vol.1.

Rinner, B. and Wolf, W. "A Bright Future for Distributed Smart Cameras," *Proceedings of the IEEE* (96:10), 2008, pp. 1562 -1564.

Rittscher, J., Kato, J., Joga, S. and Blake, A. "A Probabilistic Background Model for

Tracking" *'Computer Vision — ECCV 2000'*, 2000, pp. 336--350.

Ros, J. and Laurent, C. "Description of Local Singularities for Image Registration", *18th International Conference on Pattern Recognition, 2006. ICPR 2006.* 4, 2006, pp. 61 -64.

Rui, Y., She, A. C. and Huang, T. S. "Modified Fourier descriptors for shape representation – a practical approach" *'Proc of first international workshop on image databases and multimedia search'*, 1996.

Scheunders, P., Livens, S., Wouwer, G. V. D., Vautrot, P. and Dyck, D. V. "Wavelet-based Texture Analysis," *Int. Journal of Computer Science and Information Management, Special issue on Image Processing (IJCSIM* (1), 1998.

Schmid, C. "A structured probabilistic model for recognition", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999.* 2, 1999, pp. 490 Vol. 2.

Schmid, C. and Mohr, R. "Local grayvalue invariants for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* (19:5), 1997, pp. 530 -535.

Schmid, C., Mohr, R. and Bauckhage, C. "Comparing and evaluating interest points", *Sixth International Conference on Computer Vision, 1998.* , 1998, pp. 230 -235.

Schwartz, W. and Davis, L. "Learning Discriminative Appearance-Based Models Using Partial Least Squares", *XXII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI), 2009* , 2009, pp. 322 -329.

Shimada, A. and Taniguchi, R. "Hybrid Background Model Using Spatial-Temporal LBP", *Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, 2009. AVSS '09.* , 2009, pp. 19 -24.

Silpa-Anan, C. and Hartley, R. "Optimised KD-trees for fast image descriptor matching", *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008.* , 2008, pp. 1 -8.

Sivic, J. and Zisserman, A. "Efficient Visual Search for Objects in Videos," *Proceedings of the IEEE* (96:4), 2008, pp. 548 -566.

Sivic, J. and Zisserman, A. "Video Google: a text retrieval approach to object matching in videos", *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings.* , 2003, pp. 1470 -1477 vol.2.

Sklansky, J. "Image Segmentation and Feature Extraction," *IEEE Transactions on Systems, Man and Cybernetics,* (8:4), 1978, pp. 237 -247.

Stauffer, C. and Grimson, E. "Similarity templates for detection and recognition", *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001.* 1, 2001, pp. I-221 - I-228 vol.1.

Stauffer, C. and Grimson, W. "Adaptive background mixture models for real-time

tracking", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999.* 2, 1999, pp. 252 Vol. 2.

Tu, P. H., Doretto, G., Krahnstoever, N. O., Perera, A. A. G., Wheeler, F. W., Liu, X., Rittscher, J., Sebastian, T. B., Yu, T. and Harding, K. G. "An intelligent video framework for homeland protection", *in 'Proceedings of SPIE Defence and Security Symposium - Unattended Ground, Sea, and Air Sensor Technologies and Applications IX'*, Orlando, FL, USA, (Invited paper), 2007.

Tuytelaars, T. & Mikolajczyk, K. (2008), 'Local Invariant Feature Detectors: A Survey.' *Foundations and Trends in Computer Graphics and Vision* 3(3), 177-280.

Valera, M. and Velastin, S. "Intelligent distributed surveillance systems: a review," *IEE Proceedings -Vision, Image and Signal Processing,* (152:2), 2005, pp. 192 - 204.

Van Gool, L., Moons, T. and Ungureanu, D. "Affine / photometric invariants for planar intensity patterns", *in 'Computer Vision — ECCV '96'*, Springer Berlin / Heidelberg, 10.1007/BFb0015574, 1996, pp. 642-651.

Viola, P. and Jones, M. "Rapid object detection using a boosted cascade of simple features", *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001.* 1, 2001, pp. I-511 - I-518 vol.1.

Wang, L. and Healey, G. "Using Zernike moments for the illumination and geometry invariant classification of multispectral texture," *IEEE Transactions on Image Processing,* (7:2), 1998, pp. 196 -203.

Wang, L., Hu, W. and Tan, T. "Recent developments in human motion analysis," *Pattern Recognition* (36:3), 2003, pp. 585 - 601.

Wren, C., Azarbayejani, A., Darrell, T. and Pentland, A. "Pfinder: real-time tracking of the human body", *Proceedings of theSecond International Conference on Automatic Face and Gesture Recognition, 1996.,* , 1996, pp. 51 -56.

Yilmaz, A., Javed, O. and Shah, M. "Object tracking: A survey," *ACM Computing Surveys (CSUR)* (38), 2006.

Yu, Y., Harwood, D., Yoon, K. and Davis, L. "Human appearance modeling for matching across video sequences," *Machine Vision and Applications* (18), 2007, pp. 139-149.

Zhang, D. and Lu, G. "A comparative study of curvature scale space and Fourier descriptors for shape-based image retrieval," *Journal of Visual Communication and Image Representation* (14:1), 2003, pp. 39 - 57.

Zhong, J. and Sclaroff, S. "Segmenting foreground objects from a dynamic textured background via a robust Kalman filter", *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings.* , 2003, pp. 44 -50 vol.1.

Détection et ré-identification de piétons par points d'intérêt entre caméras disjointes

RESUME : *Avec le développement de la vidéo-protection, le nombre de caméras déployées augmente rapidement. Pour exploiter efficacement ces vidéos, il est indispensable de concevoir des outils d'aide à la surveillance qui automatisent au moins partiellement leur analyse. Un des problèmes difficiles est le suivi de personnes dans un grand espace (métro, centre commercial, aéroport, etc.) couvert par un réseau de caméras sans recouvrement. Dans cette thèse nous proposons et expérimentons une nouvelle méthode pour la ré-identification de piétons entre caméras disjointes. Notre technique est fondée sur la détection et l'accumulation de points d'intérêt caractérisés par un descripteur local.*

D'abord, on propose puis évalue une méthode utilisant les points d'intérêts pour la modélisation de scène, puis la détection d'objets mobiles. Ensuite, la ré-identification des personnes se fait en collectant un ensemble de points d'intérêt durant une fenêtre temporelle, puis en cherchant pour chacun d'eux leur correspondant le plus similaire parmi tous les descripteurs enregistrés précédemment, et stockés dans un KD-tree.

Enfin, nous proposons et testons des pistes d'amélioration, en particulier pour la sélection automatique des instants ou des points d'intérêt, afin d'obtenir pour chaque individu un ensemble de points qui soient à la fois les plus variés possibles, et les plus discriminants par rapport aux autres personnes. Les performances de ré-identification de notre algorithme, environ 95% d'identification correcte au premier rang parmi 40 personnes, dépassent l'état de l'art, ainsi que celles obtenues dans nos comparaisons avec d'autres descripteurs (histogramme de couleur, HOG, SIFT).

Mots clés : Vidéosurveillance, réseaux de caméras, points d'intérêt, ré-identification, détection, ré-acquisition

Pedestrian detection and re-identification using interest points between non overlapping cameras

ABSTRACT: *With the development of video-protection, the number of cameras deployed is increasing rapidly. To effectively exploit these videos, it is essential to develop tools that automate monitoring, or at least part of their analysis. One of the difficulties, and poorly resolved problems in this area, is the tracking of people in a large space (metro, shopping center, airport, etc.) covered by a network of non-overlapping cameras. In this thesis, we propose and experiment a new method for the re-identification of pedestrians between disjoint cameras. Our technique is based on the detection and accumulation (during tracking within one camera) of interest points characterized by a local descriptor.*

We present and evaluate a keypoints-based method for modeling a scene background and detecting new (moving) objects in it. Then we present and evaluate our method for identifying a person by matching the interest points found in several images. One of the originalities of our method is to accumulate interest points on sufficiently time-spaced images during person tracking, in order to capture appearance variability. We produce quantitative results on the performance of such a system to allow an objective comparison with other features (SIFT, Color, HOG). Finally, we propose and test possible improvements, particularly for the automatic selection of moments or interest points, to obtain a set of points for each individual which are the most varied and more discriminating to those of other people. This probabilistic variant of our method brings tremendous improvement to performance, which rises at 95% first rank correct identification among 40 persons, which is above state-of-the-art.

Keywords : Video-surveillance, camera networks, person identification and tracking, re-identification, reacquisition, interest points.

MINES ParisTech

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY