PhD Dissertation

International Doctorate School in
Information and Communication Technologies

Department of Information Engineering and Computer Science
University of Trento

Detection and Analysis Methods
for Unmanned Aerial Vehicle Images

Thomas Moranduzzo

Advisor: Prof. Farid Melgani, University of Trento

January 2015

Abstract

Unmanned Aerial Vehicles (UAVs), commonly known as drones, are aerial platforms that are gaining large popularity in the remote sensing field. UAVs derive from military technology, but in the last few years they are establishing as reference platforms also for civilian tasks. The main advantage of these acquisition systems lies in their simplicity of use. Indeed, a UAV can be used when and where it is needed without excessive costs. Since UAVs can fly very close to the objects under investigation they allow the acquisition of extremely high resolution (EHR) images in which the items are described with a very high level of details. The huge quantity of information contained in UAV images opens the way to develop novel applications but at the same time force us to face new challenging problems at methodological level. This thesis represents a modest but hopefully useful contribution towards making UAV images completely understood and easily processed and analyzed.

In particular, the proposed methodological contributions include: i) two methods devoted to the automatic detection and counting of cars present in urban scenarios; ii) a complete processing chain which monitors the traffic and estimate the speeds of moving vehicles; iii) a methodology which detects classes of objects by exploiting a nonlinear filter which combines image gradient features at different orders and Gaussian process (GP) modeling; iv) a novel strategy to "coarsely" describe extremely high resolution images using various representation and matching strategies.

Experimental results conducted on real UAV images are presented and discussed. They show the validity of the proposed methods and suggest future possible improvements. Furthermore, they confirm that despite the complexity of the considered images, the potential of UAV images is very wide.

Contents

List of Tables:

List of Figures

Glossary

ANN: artificial neural network

BOW: bag of visual words

DOG: difference of Gaussian

EHR: extremely high resolution

GIS: geographic information system

GLOH: gradient location and orientation histogram

GP: Gaussian process

HOG: histogram of oriented gradients

HSV: hue, saturation, value

LBP: local binary pattern

LDA: latent dirichlet allocation

LIDAR: light detection and ranging

MM: mathematical morphology

OAO: one against one

PCA: principal component analysis

RGB: red, green, blue

SAR: synthetic aperture radar

SE: structural element

SIFT: scale invariant feature transform

SURF: speeded up robust feature

SVM: support vector machines

SWIR: short-wave infrared

UAV: unmanned aerial vehicles

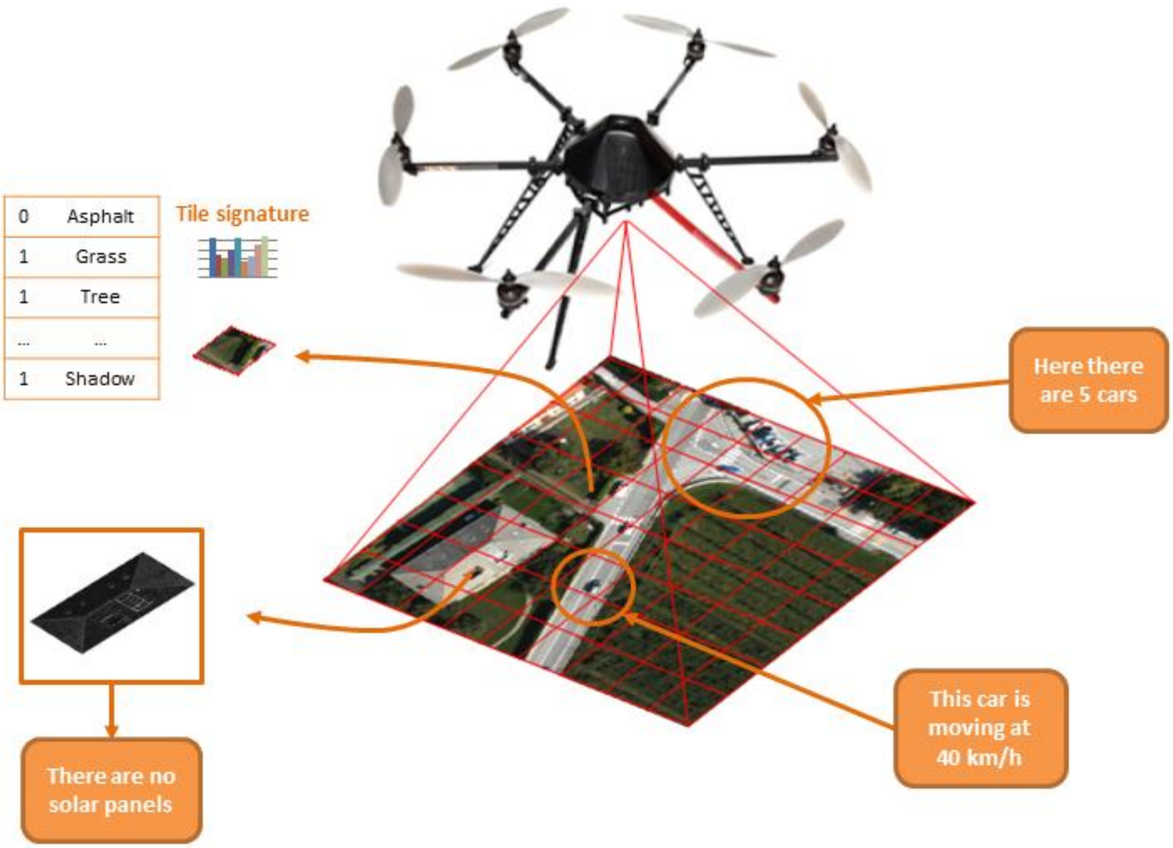
VNIR: visible and near infrared

VHR: very high resolution

WSV: wireless sensor network

1. Introduction and Thesis Overview

Abstract *– The main aim of this chapter is to give to the reader a complete overview about the general context in which the thesis is positioned. In a second part, the problems faced in the following chapters are introduced. Finally, we describe the proposed solutions and the thesis structure and organization.*



0	Asphalt
1	Grass
1	Tree
...	...
1	Shadow

1.1. The Context

Remote sensing is the science of acquiring information about objects or areas from distance, typically using satellites or aerial devices. Depending on the sensor used to collect information, remote sensing may be divided into: active and passive (Fig 1.1). Active remote sensing exploits sensors that emit microwave radiations and measure the backscattered energy. The study of the reflected energy allows the discrimination between the different objects and areas. The capacity of collecting information anytime, night or day, and in all atmospheric conditions represents the main advantage of active sensors. Typical examples of active sensors are synthetic aperture radar (SAR) and light detection and ranging (LIDAR). These last sensors measure the time difference between emission and return and thanks to that they recover information such as location, distance, elevation and speed of the objects. Passive sensors, instead of using an own energy to illuminate a target, measure the energy that the targets reflect from independent sources (*i.e.*, the Sun) or that they themselves produce (*i.e.*, thermal energy). Contrarily to active sensors, they do not need big energy sources for the production of energy but they completely depend on the external sources. In this category, we find the optical sensors which are classified depending on the number of spectral bands used *(i.e.*, multispectral/hyperpectral imaging systems). The wavelength region exploited by passive sensors usually ranges from the visible and near infrared (VNIR) to the short-wave infrared (SWIR). Different objects absorb and reflect the wavelengths in different way; therefore the objects are discriminated by studying their spectral reflectance signature [1]- [2].

The desire to observe the Earth from satellite or aerial devices has always fascinated the scientific community and it is the main reason of the fast development of remote sensing devices. A good example of this technological progress is the Landsat program. The first Landsat satellite (Landsat 1 with spatial resolution of 60 meters) was launched in the 1972 while the last Landsat satellite (Landsat 8 with spatial resolution of 15 meters) was launched in 2013. In four decades, the Landsat program has made great

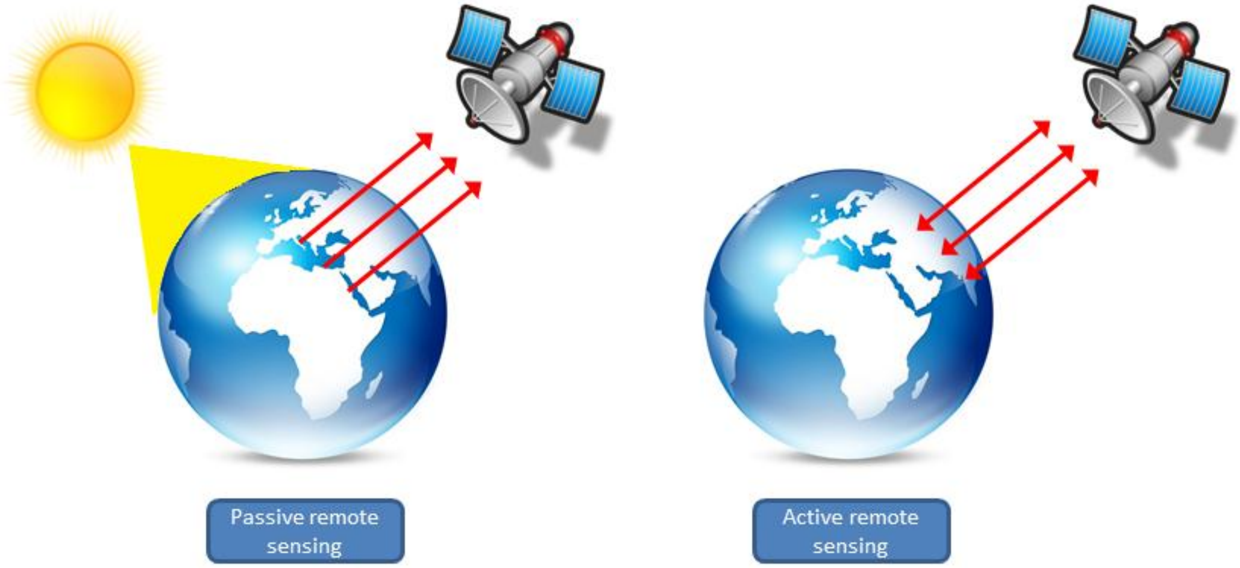


Figure 1.1. Passive and active remote sensing.

advancements and the quality of collected information has considerably increased (*i.e.*, the spatial resolution is quadruple). Landsat instruments have collected a massive amount of information, and are key resources in several applications such as: agriculture, geology, forestry, cartography, regional planning, surveillance and education [3]- [4]. The last generations of very high resolution (VHR) optical satellites, equipped with multispectral sensors capable of acquiring images characterized by spatial resolutions in the order of 40-50 centimeters, opens new opportunities thanks to the increased spatial information they convey. Examples of VHR optical satellites are: IKONOS-2, WorldView-1 and -2, QuickBird and GeoEye-1 and -2. The commercial availability of sub-metric resolution optical satellite images allows the extraction of more accurate and qualitatively different information which calls for new analysis and processing strategies. The huge quantity of information contained in VHR images consents the developments of innovative applications but at the same time introduces new challenging problems. For instance, the presence of shadows, which are distinctly visible in VHR images, represents an obstacle which could limit the potential of VHR images. In [5], the authors analyze the problem of reconstruction of areas covered by shadows. In particular, they investigate various criteria to understand if the information hidden under the shadows can be well recovered. Usually, satellite imagery is complemented with aerial photography, which has higher spatial resolution but at the expenses of smaller covered area (Fig. 1.2). Platforms for aerial photography include helicopters, balloons, fixed-wing aircrafts, Unmanned Aerial Vehicles (UAV) and many others. These platforms could be equipped with different sensors and consequently they could be involved in several tasks.

Among the fast growing remote sensing technologies, UAVs are certainly those that are emerging in the

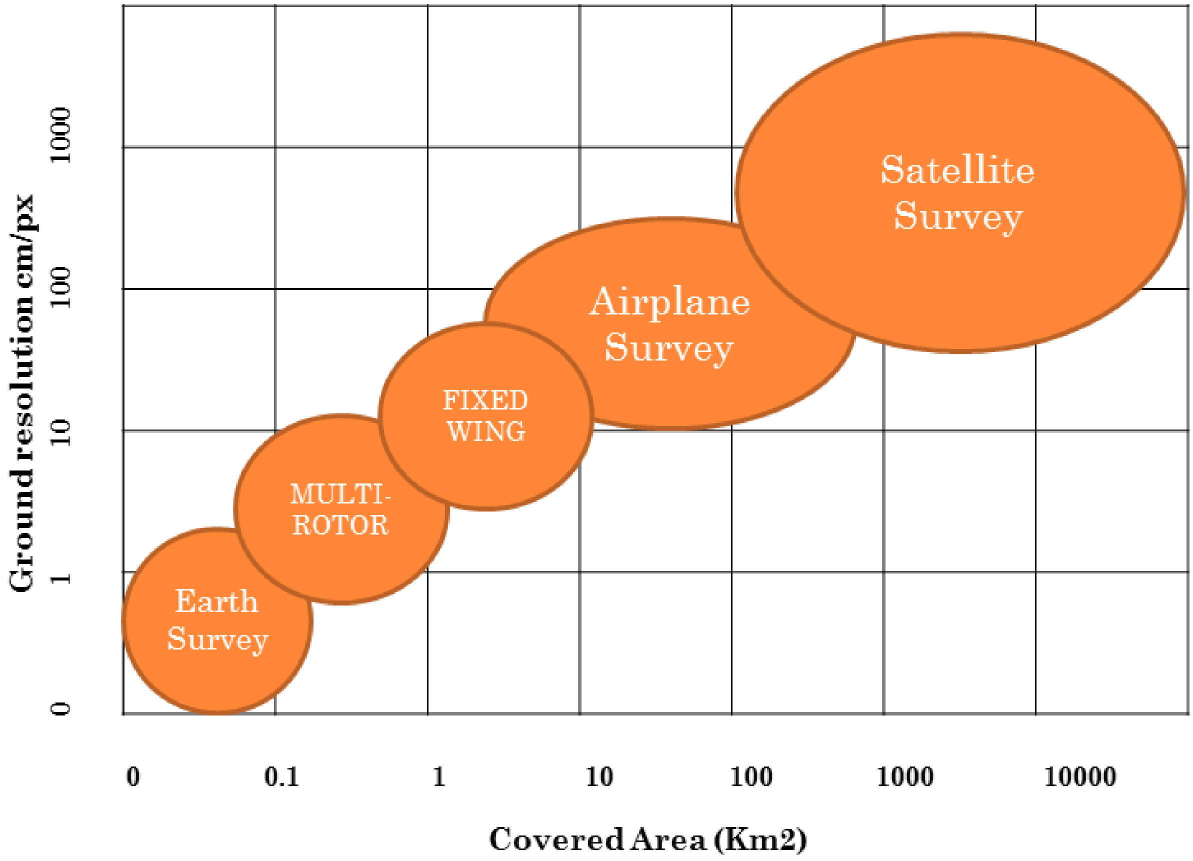


Figure 1.2. Comparison of spatial resolution with covered areas of the different remote sensing technologies.

last few years. UAVs have been initially developed for military purposes, but thanks to their great potential they have started to be used also for civilian applications and today they represent a valid alternative or a supplementary solution to satellite or to airborne devices especially for small coverage or inaccessible areas (Fig. 1.3). UAVs, commonly known also as drone, are small, ecologic and sometimes silent aerial platforms without pilot on board and controlled by ground stations. Thanks to their timely and more detailed (extremely high image spatial resolution (EHR)) capacity to collect data with respect to other acquisition systems, UAVs are distinguishing as innovative and cost-effective devices to perform innumerable survey tasks. Nowadays, UAVs are finding application in many fields, ranging from environmental monitoring to the inspection of big industrial plants. Moreover, since UAVs have demonstrated to be fast and very accurate in the acquisition of information, the advancements of this technology are by now one the main goal of many research centers.

Figure 1.3. Example of military and civilian UAVs.

A field where UAVs have been successfully introduced and are gaining a remarkable success is the agricultural field. The use of different sensors, such as thermal and multispectral sensors, combined with the extremely high spatial resolution of UAV imagery allows to provide suitable solutions depending on the analyzed crop. All farmers could benefit from this technology by focusing on specific problems or information, such as soil moisture or crop disease, and consequently they could become better managers of their production practices. Zhang *et al.* [6] present a complete overview about the exploitation of UAVs for precision agriculture purposes. In particular, they show the feasible applications and the range of available sensors and platforms. In [7], the authors estimated the water potential by using thermal and micro-hyperspectral sensors. Furthermore, Gonzalez-Dugo et al. [8] propose a work in which by using thermal aerial imagery they describe the spatial variability of crop water status within a commercial orchard where five different fruit tree species were grown. In [9], an architecture which combines UAV and a wireless sensor network (WSN) for spraying chemicals on crops is described. The UAV path is controlled by the ground network which takes the decision on the bases of atmospheric conditions and on the amount of chemicals already applied.

UAVs also represent a novel environmental remote sensing tool to monitor the status of air, water and ground. An interesting environmental application is presented in [10] where the authors describe a system for the coastal monitoring. High resolution images are used to create digital elevation models (DEM) which are used to investigate the modification of the topography. Another interesting environmental application is presented in [11]. In this work, the author uses thermal camera to detect the presence of small fawns in meadows in order to prevent their killing by mowing machines.

One of the main characteristics of UAV consists in the possibility to immediately acquire information when it is necessary without endangering the life of people since they are piloted from ground stations. This advantage makes UAVs particularly appealing for surveillance and emergency situations. For instance, in [12] the authors describe a path planning algorithm, which exploits infrared images, to track forest fire perimeters and to obtain information about the spreading of the fire. Choi *et al.* [13] present a flexible system for the low cost rapid mapping in emergency situation. To achieve this goal, the UAV platform has been equipped with two optical cameras and a LIDAR. In the context of surveillance, in [14] a study in which several state-of-the-art methods to automatically detect people lying on the ground in UAV images is presented and discussed. From that study, on could deduce that part-based models work better than other models since they are able to overcome problems related to partial occlusions. Another surveillance problem is presented in [15] where the authors propose a robust approach based on cascades of Haar classifiers for the real-time detection of people and vehicles.

The possibility to obtain information from inaccessible areas makes UAVs the perfect acquisition systems to monitor archaeological sites. The data collected by using UAVs allow performing analysis and measurements that otherwise would be impossible. In [16], the authors show how UAV technology offers practical and inexpensive solutions to support archaeological analysis. Moreover, Saleri *et al.* [17] describe a complete archaeological survey starting from the acquisition of the data to the creation of an accurate 3D model. Eisenbeiss *et al.* [18] propose two different approaches devoted to the generation of DSMs of cultural heritages. The UAV images are processed with a multi-image matching approach to create DSMs with 10 centimeters of resolution.

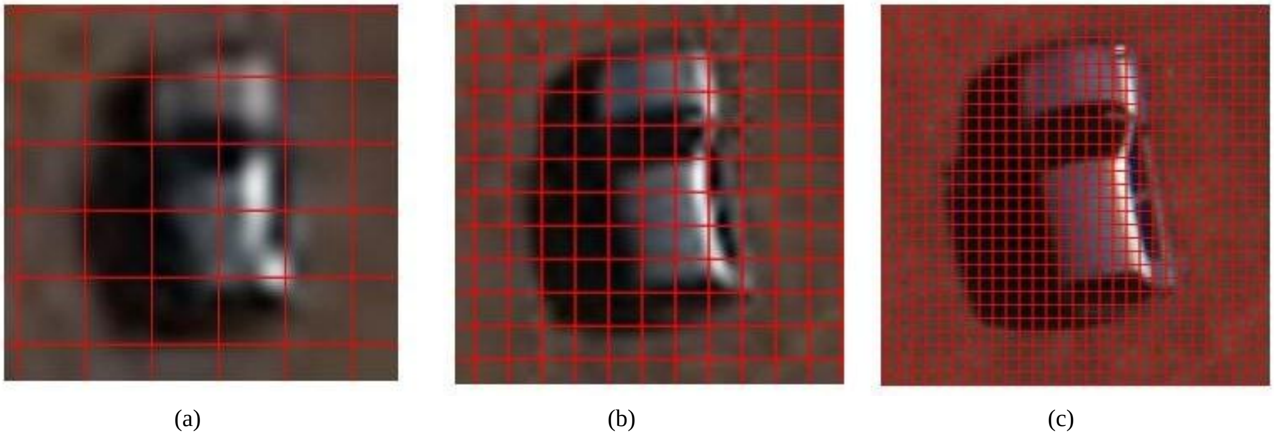


Figure 1.4. Image spatial resolution in connection with the number of pixel that cover the same object. (a) medium resolution; (b) high resolution; (c) extremely high resolution

The assortment of applications in which UAVs are involved is very wide, but despite the rapid development and the wide use, many studies have still to be done to well understand how to manage all the data that UAVs are able to acquire. As long as the resolution of the acquired images could be comparable with the dimension of the investigated objects (Fig 1.4), all the works were inspired by pixel-based approaches, but with the advent of UAV and consequently of EHR images novel techniques based on object-based analysis have been proposed. Blaschke *et al.* [19] raised the problem if it is still suitable, regardless of the spatial resolution of available imagery, handle remote sensing problems with pixel-based approaches. Furthermore, in [20], the author gives an overview of the development of object-based methods. EHR images contain a huge quantity of information therefore they need to be handled with suitable strategies in order to exploit all their potential. An example could be found in [21], where the authors expose the problem regarding the use of common texture measures for the analysis of UAV images. These measures have been widely exploited for pixel-based analysis but their usage in object-based analysis has not been investigated. Throughout their work the authors explore different texture measures and image scales for the classification of land cover and they show how object-based analysis is highly suitable for extremely high resolution imagery.

UAVs, by flying very close to the analyzed targets, collect images characterized by EHR resolution in which the objects are described by a high level of details and even the smallest particulars are very well defined. This outstanding quality of the images makes that some studies have adopted computer vision approaches for the processing and analysis of UAV data [22]- [23]. Shi *et al.* [24] present a novel object-based method for change detection which exploits the potential of EHR-UAV images. The approach combines segmentation information and the Scale Invariant Feature Transform (SIFT) [25] algorithm to register multi-temporal images and highlight the changes. Another example about the exploitation of computer vision techniques for the analysis of EHR remote sensing images is proposed in [26]. The authors investigate the use of SIFT and Speeded Up Robust Feature (SURF) [27] algorithms for the characterization of natural environments. In particular, they present an analysis about the detection and matching performance in visible and infrared domain of the two feature detection and description techniques.

1.2. Thesis Objective, Solutions and Organization

As mentioned in the previous subsection, next years will be characterized by a growth of the use of Unmanned Aerial Vehicles for civilian purposes. These acquisition systems collect images characterized by a huge potential but without appropriate processing techniques will not be completely exploited. For this reason, the rising market of UAVs, and consequently of EHR images, requires new image processing and analysis strategies able to handle all the information contained in such images. In this thesis work, we will propose new techniques inspired to computer vision studies able to efficiently manage UAV images to solve some common problems in urban scenarios.

After this first introductive part, the thesis is organized into 4 chapters. The first part, divided in two sections, describes the problems of car detection and counting and traffic monitoring in urban scenarios, whereas the second and the third parts are concerned with the problems of object detection and image classification, respectively.

In Chapter 2, two techniques devoted to the detection and counting of cars and an approach to monitor the traffic in urban areas by exploiting UAV images are presented. Both methods proposed in the first subsection rely on two well know computer vision algorithms: the SIFT and the Histogram of Oriented Gradients (HoG), respectively. The underlying idea of the first proposed technique consists to transform the image into a set of keypoints found according to the SIFT transform and to detect cars through such an image representation. Whereas, the second proposed car detection technique describes the image with features highlighting the shape of the objects (*i.e.*, HoG features) and identifies the presence of cars by exploiting this representation and opportune similarity measures computed on the basis of a catalog of cars. In the second subsection of this chapter, a procedure for the automatic detection of moving vehicles and for the estimation of their speed is presented. The procedure takes advantage of the ability of UAVs of acquiring sequence of EHR images with a very small time lag over the same area. Experimental results conducted on real UAV datasets are exposed and commented for all proposed approaches.

In Chapter 3, we propose an innovative methodology for the developing a filter that it is able to automatically detect a specific class of objects from EHR remote sensing imagery. The proposed filter aims at detecting the desired class of targets exploiting the intrinsic characteristic of the objects such as shape, appearance and dimension that are typically not similar among different classes of objects and which are particularly emphasized in EHR images. The described algorithm after the extraction of features related to the structure and shape of the objects uses a nonlinear model to identify the areas where the objects under investigation are potentially present. The filter is created to be completely customizable, by supervised training, giving the possibility to decide the class of object to identify. One main aim that we pursued with this work is to propose an approach that generates the final results in short time. A technique that is able to quickly manage the enormous amount of data conveyed in UAV images is of great importance when it comes to real time or near real time applications. Results obtained from the detection of two classes of objects very common in urban scenario (*e.g.*, vehicles and solar panels) are presented and discussed.

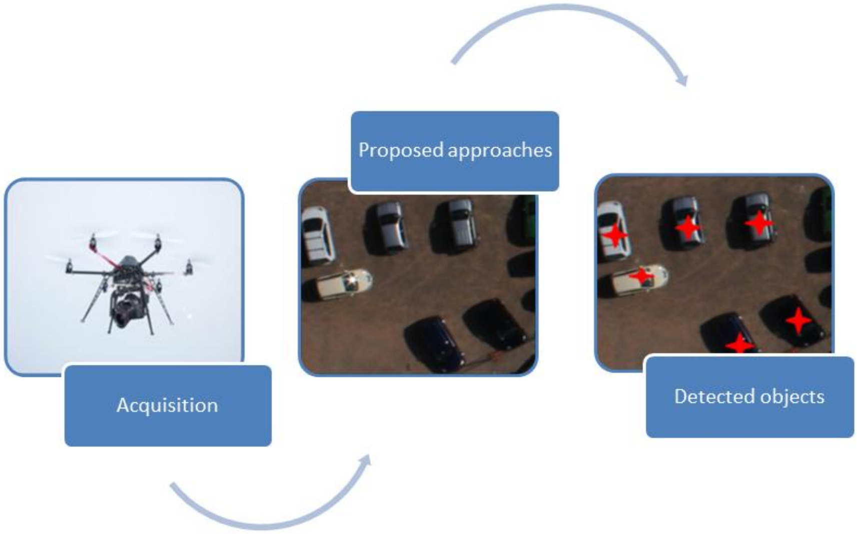
In the beginning of this thesis, we proposed three techniques to detect and monitor a particular class of objects (*i.e.,* vehicles) and then, in the second chapter, we presented the construction of a filter that aims at identifying specific classes of objects depending on a supervised training. Now making a step forward, in Chapter 4, we propose a technique that automatically describes a query image. Considering EHR images, a standard pixel-based classification may provide very time consuming and unsatisfactory results which are usually both undesirable. For this reason, we propose a “coarse” description technique which provides global results for the considered images. Differently from pixel-based classification techniques, a “coarse” description strategy does not assign to each single pixel a value, which denotes the membership class, but it simply indicates which of the possible classes of objects are present in the considered image or in the investigated part of image. In the context of this work, different images representation techniques and similarity measures are taken into consideration and compared. Also for this proposed technique experimental results are proposed and analyzed.

In the end, the last chapter is devoted to the final conclusions. After a short summary, some discussions about achieved results, open issues and future developments are given.

This dissertation has been written supposing that the Reader is familiar with the basic concepts regarding the image processing, remote sensing and pattern recognition fields. Otherwise, the Reader is recommended to consult the references which are available at the end of this dissertation. They are useful to give a complete and well-structured overview about the topics discussed throughout the manuscript. The following chapters have been written in such a way to be independent between each other to give to the Readers the possibility to read only the chapter/s of interest, without loss of information.

2. Traffic Monitoring Strategies in UAV images

Abstract *– This chapter presents innovative methods for the automatic detection and counting of cars and for the estimation of their speed in Unmanned Aerial Vehicle (UAV) images acquired over urban contexts. UAV images are characterized by an extremely high spatial resolution, which makes the detection of cars particularly challenging. The proposed car detection and counting methods start with a screening operation in which the asphalted areas are identified in order to make the processes faster and more robust. Then, after the screening operation, both methods proceed with their respective core part. The first method performs a feature extraction process based on Scalar Invariant Feature Transform (SIFT) thanks to which a set of keypoints is identified and opportunely described. Successively, it discriminates between keypoints assigned to cars and all the others, by means of a Support Vector Machine (SVM) classifier. Differently, the second method carries out filtering operations in the horizontal and vertical directions to extract Histogram of Gradient (HoG) features and to yield a preliminary detection of cars after the computation of a similarity measure with a catalogue of cars used as reference. Three different strategies for computing the similarity are investigated. Successively, for the image points identified as potential cars, an orientation value is computed by searching for the highest similarity value in 36 possible directions. In the end, both methods group the points (i.e., the car keypoints or the image points associated to the car class) belonging to the same car in order to get a "one point – one car" relationship thanks to a spatial clustering. Furthermore, in this chapter, a method to monitor the traffic by tracking and estimating the speed of moving cars from sequences of images is described. The method begins with the automatic registration of pairs of consecutive images of a sequence by using a geometric transformation obtained from the matching of invariant points. By comparing the images and thanks to mathematical morphology operations the moving cars are isolated. In the last step, the spatial coordinates in both images of each car are extracted and thanks to the derived spatial shift estimations about the speeds are inferred. Interesting experimental results obtained with all the proposed strategies and conducted on sets of real UAV images are presented and discussed.*



2.1. Introduction

In the last decade, many studies focused on the improvement of life quality in urban centers have been conducted and they have demonstrated that a good level of life quality is strictly tied to the efficiency of the transportation system. Transport planners showed that a good transportation system not only improves the velocity of transports but also reduces problems connected to noise and environmental pollution, security and waste of resources. For this reason, being able to constantly monitor the concentration of vehicles and their speed inside urban environments could help to improve the quality of transportation systems. Indeed, many local municipalities are moving in this direction and are putting in effect strategies to predict and prevent traffic jams and congestions, to reduce road traffic accidents from traffic collisions, to limit the environmental impact and to improve the traffic safety. One approach that has established to be particularly useful to monitor urban areas consists in the exploitation of aerial technologies. Aerial platforms by flying over cities provide complete overview about the status of the transportation system in very short time.

In the current literature, one has the possibility to find several car detection techniques which mainly exploit low resolution images, many of them are based on satellite imagery. Satellites allow observing very wide areas but they lack of spatial resolution with respect to airborne or UAV platforms. In [28], the authors propose an approach to detect cars from optical satellite images. By using Haar-like features, they aim at detecting single vehicles and queue of vehicles from which single cars are identified by means of line extraction technique. In [29], a three-step framework is presented for the automatic extraction of moving vehicles and determination of their velocity. The framework exploits the time lag present between panchromatic and multispectral images. To solve the vehicle detection problem, the use of light detection and ranging (LiDAR) systems has also been explored. In [30], two methods (*i.e.*, grid-cell and 3D pointcloud-analysis-based methods) which extract vehicles are compared.

However, for precise urban monitoring applications, the use of more suitable acquisition systems such as helicopters or aircrafts is usually preferred. The opportunity to collect high resolution images in very short time about the areas of interest only when it is necessary is the main advantage of aerial platforms with respect to satellites. In [31], the authors aim to detect and count cars from aerial images by describing their geometry using a 3-D model. Zhao *et al.* [32] developed a technique to detect cars by exploiting shadows, color intensities and a Bayesian network. Furthermore, in [33] the authors proposed an interesting method to recognize cars based on an operator to highlight the edges.

Recently, the improvements of the acquisition techniques have led to the development of more accurate car detection methods which are able to consider the intra-class variability by exploiting shape and texture features. In [34], the authors propose a framework for car detection which group together different features: Histogram of Gradient (HoG), Local Binary Pattern (LBP) and Opponent Histogram. In [35], a strong classifier was generated by passing HoG, LBP and Haar like features to an on-line boosting algorithm. In [36], a framework which uses an on-line boosting technique and additional 3D data is proposed to detect cars in aerial images. Another method devoted to the detection of cars from airborne imagery is presented in [37]. In this work, the authors, before exploiting a combination of boosting technique and HoG features to extract the cars, reduced the areas to analyze by using disparity maps and a fast region growing algorithm.

Despite the encouraging results obtained by car detection techniques mentioned in the literature, many of them are not suitable for UAV images. The reasons are due to the fact that this kind of images is extremely complex. Indeed, all the objects are described with extremely high resolution and, consequently, the outstanding level of details makes the automatic discrimination between simple elements very challenging (*i.e.,* dormers over roofs are often confused with windshield or back windows of cars). Furthermore, UAV images are characterized by illumination, rotation and scale changes which increase the analysis complexity. In [38], Gleason *et al*. face the problem of vehicle detection in UAV images by combining a fast detection process with a classification stage. Man-made objects are isolated and classified thanks to a cascade of detection algorithm and a suitable classifier, respectively. The work shows interesting results over rural environments where man-made objects have completely different aspects with respect to the environments.

Contrary to the detection of cars, the problem of vehicle speed estimation from remote sensing images has not yet attracted sufficient attention mainly because previous technologies were not able to acquire temporal sequences of very high resolution images over the same area. Indeed, if the resolution of images is rough, the determination of the image spatial coordinates becomes inaccurate and, consequently, the estimation of the velocity results erroneous. With the advent of extremely high resolution images, the determination of the precise spatial position of the objects in two different times has become feasible. Therefore, the vehicle speed estimation problem from such images has started to be taken into consideration. For instance, Yamazaki *et al.* [39] suggest deriving the velocity of cars by exploiting the shadow of cars and by removing big objects such as roads, roofs and trees from pairs of images with small time lag.

In this chapter, both car detection and speed estimation problems are faced and novel techniques related to the traffic monitoring issue, especially in urban scenarios, are presented.

In the first part of this chapter, we present two alternative methods to detect and count cars for extremely high resolution images (few centimeters) representing urban areas and acquired by means of UAV. The idea behind the development of the first proposed technique lies in transforming the image into a set of keypoints found according to the SIFT transform and to detect cars through such an image representation. We believe this representation domain is more adapted to handle the very high level of details characterizing UAV images. The key idea, which guides the second methodology, consists in describing the image with features highlighting the shape of the objects and to identify the presence of cars by exploiting this representation and opportune similarity measures computed on the basis of a catalog of cars.

The second part of this chapter describes a procedure for the automatic detection of moving vehicles and for the estimation of their speed. UAVs have the capacity to acquire images with a temporal distance of a few seconds or even less depending on the sensor. Usually, two images acquired with such a small time lag are very similar and the only differences are represented by the moving objects. In this context, since the images have been acquired over roads, it is possible to easily assume that the only changes between two subsequent images are the moving vehicles. This consideration is the central pillar of this proposed three-stage procedure.

2.2. Problem Formulation

Let us consider an extremely high resolution image $I(x, y)$ (where (x, y) stands for pixel coordinates in image I) acquired by means of an UAV over an urban area. All the information which characterizes EHR images is certainly useful to accurately describe the objects, but at the same time it represents also a problem because it tends to emphasize the smallest differences (Fig. 2.1). To deal with classification and detection problems, this high level of intra class variability requires suitable object representation techniques. The fundamental idea which leads both strategies is to find a suitable image representation technique which emphasizes the presence of vehicles. Both methodologies adopt a two-level processing schema, where the first one (*i.e.*, screening) works at pixel-level and is aimed at separating between asphalt and non-asphalt areas while the second one is intended to identify the suitable image representation. The two proposed strategies exploit two different image representation techniques which derive from the computer vision field. The first representation exploits sets of points of interest while the second representation is based on feature which highlights the shape of the objects.

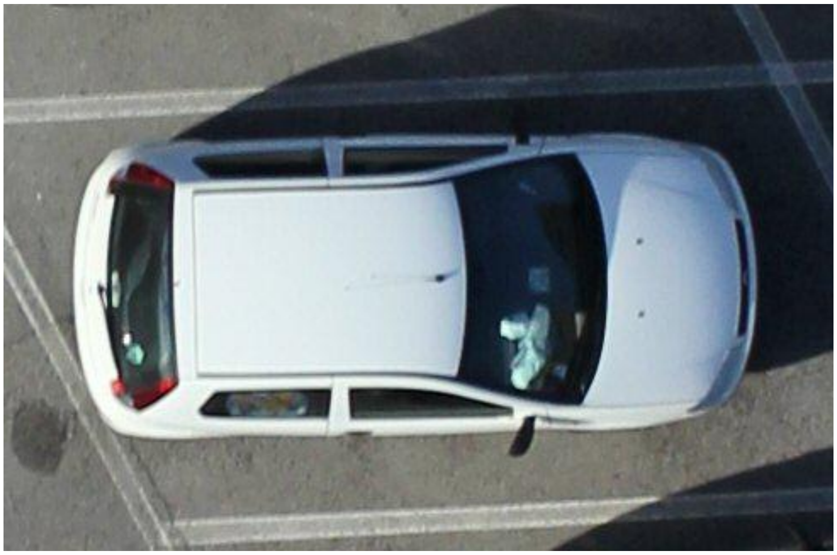


Figure 2.1. Example showing the level of details captured by a typical UAV image.

2.3. Screening

Assuming that usually cars in urban scenarios lie only over asphalted areas (*i.e.*, roads or parking lots), we will restrict the investigated areas only to these regions. This provides us two significant advantages: 1) to improve the velocity of detection by limiting the areas to analyze; and 2) to reduce the number of false alarms.

The recognition of roads and parking lots can be envisioned in two manners. In the first one, the most accurate one, the mask to isolate the regions covered by asphalt is obtained from road maps possibly available in a Geographic Information System (GIS) covering the areas under analysis. In this way, no new screening is required since all asphalted areas are known a priori, making it easy to build the desired mask. In the second manner, screening is performed with an automatic procedure applied on the acquired image and made up of two steps: 1) a classification phase, which allows discriminating between asphalted and non-asphalted regions; and 2) a cascade of two morphological filters applied to improve the results of the classification.

The technique used to perform the classification is based on a Support Vector Machine (SVM). Let us consider a training set composed by N samples $x_i \epsilon \Re^d$ $(i = 1,2, \dots N)$ taken from the d-dimensional feature space X. To each training sample, a label $\{-1, +1\}$ is associated depending on its class (*i.e.*, asphalt and non-asphalt). During the training phase, the SVM classifier tries to find a hyperplane in a kernel-induced feature space $(\Phi(X) \in \Re^{d'}, d' > d))$ which best separates the two groups of training samples. This hyperplane allows deriving a discriminant function $f(x)$ useful to take the membership decision:

$$f(x) = w^* \Phi(x) + b^* \tag{2.1}$$

The training step consists basically in the estimation of w*, the weight vector normal, and b*, the bias, by solving a linearly constrained quadratic optimization problem defined as:

$$\min_{w,b,\varepsilon}(\frac{1}{2} w^T w + C \sum_{i=1}^{N} \varepsilon_i) \tag{2.2}$$

subject to the following constraints:

$$\begin{cases} y_i(w \cdot \Phi(x_i) + b) \geq 1 - \varepsilon_i \\ \varepsilon_i \geq 0 \end{cases} \qquad \text{for } i = 1,2, \cdots, N \tag{2.3}$$

where the slack variable ε_i allows functional margin constraint violations and C is a nonnegative regularization constant, which controls the smoothness of the discriminant function in the original feature space. The final result is a discriminant function expressed as a function of the data in the original (lower) dimensional feature space X:

$$f(x) = \sum_{i \in S} a_i^* y_i k(x_i, x) + b^* \tag{2.4}$$

where $K(\cdot,\cdot)$ is a kernel function and a_i^* 's are Lagrange multipliers. For more details about SVMs, we refer the reader to [40] [41] [42] [43] [44].

Since pixel-based classification is typically characterized by salt-and-pepper noise and since we do not need a map of very high accuracy but just a mask where to search for cars, we will perform a refinement procedure based on the use of the mathematical morphology (MM) theory. The MM is very popular in the image processing field. Originally, it was developed for binary images but then it was adapted also to grayscale images. The main idea of MM is to investigate an image through the use of a basic element called structuring element (SE). The SE is run all over the image $I(x, y)$, and at each spatial position (x, y) the relationship between the element and the image is analyzed. A SE can assume any shape but the most common shape is the disk. The two main morphological operations, strongly related to the Minkovsky addition [45], are the Dilation and the Erosion.

The Dilation operation on $I(x, y)$ by SE is defined by:

$$I \oplus SE = \{z: (SE^s)_{+z} \cap I \neq \emptyset\} = \bigcup_{y \in SE} I_{+y} \tag{2.5}$$

where the $I_{+y} = \{x + y: x \in I\}$ is the translation of $I(x, y)$ along vector y, and $SE^s = \{x: -x \in SE\}$ is the symmetric of SE with respect to the origin.

Similarly, the Erosion operation on $I(x, y)$ by SE can by defined as:

$$I \ominus SE = \{z: SE_{+z} \subseteq I\} = \bigcup_{y \in SE} I_{-y} \tag{2.6}$$

In our method, first we apply an erosion operation, using a small SE, on the classification map in order to reduce the presence of noise, and then we perform a dilation operation to create homogenous asphalted regions and to recover areas such as pedestrian crossings and "holes" caused by the presence of cars on asphalt (not classified as asphalt by the classifier).

2.4. An Automatic Car Counting Method for Unmanned Aerial Vehicle Images

The first proposed strategy, which combines SIFT algorithm and a SVM classifier, is structured over 3 main steps: feature extraction, feature classification and merging (Fig. 2.2).

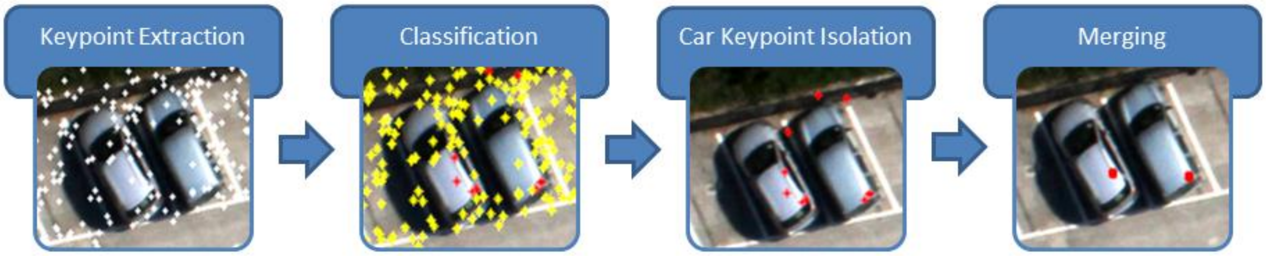


Figure 2.2. Illustration of the keypoints extraction, classification and merging stages.

After the restriction of the areas of analysis only to the regions covered by asphalt thanks to the screening step, now we will focus on the identification of the features which can help us in the detection of cars. Since our work is based on the recognition of a specific class of objects, we want to find features which are typical of this class. These features have to be invariant to image scale, rotation and translation and not affected by illumination changes because we want to recognize them in all image acquisition conditions. Several object descriptors fulfilling these requirements can be found in the computer vision literature such as: scale-invariant feature transform (SIFT) [25], gradient location and orientation histogram (GLOH), shape context [46], spin images [47], steerable filters [48], and differential invariants [49]. Such descriptors are based on the extraction of histograms which describe the local proprieties of the points of interest. The main differences between them lie in the kind of information conveyed by the local histograms (*e.g.*, intensity, intensity gradients, edge point locations and orientations) and the dimension of the descriptor. An interesting comparative study was proposed in [50], where it is shown that SIFT descriptors perform better than other typical local descriptors. In this work, we will also rely on the SIFT algorithm proposed by Lowe, [25], in order to localize and characterize the keypoints in a given image.

The SIFT algorithm is widely used in the computer vision community for its effectiveness in describing high resolution objects in complex scenes. It thus potentially fits well our scope, i.e., the detection of cars in UAV images typically characterized by extremely high spatial resolution. Moreover, the intrinsic variability of cars in the shape, the color, and the variable position conditions (rotation and scale problems) make the Lowe's method a good solution candidate for our problem, which also can potentially overcome the problem of partial occlusions (i.e., cars partially hidden by trees or shadows) common in urban environments.

The algorithm starts with the extraction of Scale Invariant Feature Transform (SIFT) keypoints which are highly distinctive. Each identified keypoint is then characterized by a feature vector which describes the area surrounding such keypoint. Since these keypoints are invariant to many transformations of the images, we think they can potentially be appropriate features for the characterization of the vehicles in an image.

The process used to produce the SIFT features is composed mainly by four steps.

The first of the four steps is devoted to the identification of possible locations which are invariant to scale changes. This objective is carried out by searching for stable points across various possible scales of a scale space properly created by convolving I with a variable scale Gaussian filter:

$$L(x, y, \sigma) = I(x, y) * \frac{1}{2\pi\sigma^2} exp(-\frac{x^2 + y^2}{2\sigma^2}) \tag{2.7}$$

where ‘ $*$ ’ is the convolution operator and σ a scale factor.

The detection of stable locations is done by identifying scale-space extrema in the difference-of-Gaussian (DoG) function convolved with the original image:

$$D(x, y, \sigma) = L(x, y, \sigma) - L(x, y, k\sigma) \tag{2.8}$$

where k is a constant multiplicative factor which separates the new image scale from the original image. To identify which points will become possible keypoints, each pixel in the DoG is compared with the 8 neighbors at the same scale and with the other 18 neighbors of the two neighbor scales. A pixel is called keypoint if it is larger or smaller than all the other 26 neighbors. The points getting extrema in the DoG are then classified as candidate locations. DoG function is sensitive to noise and edges, hence a careful procedure to reject points with low contrast and poorly localized along the edges is necessary. This improvement is done considering the Taylor expansion of the scale-space function and shifting the $DoG(x, y, \sigma)$ so that the origin is at the sample point:

$$D(x) = D + \frac{\partial^2 \Omega}{\partial u^2} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \tag{2.9}$$

where D and its derivatives are evaluated at the sample point and $X = (x, y, \sigma)^T$ is the offset from this point. The location of the extremum $\hat{X}$ is determined by taking the derivative of this function with respect to X and setting it to zero, giving:

$$\hat{X} = -\left(\frac{\partial^2 D}{\partial X^2}\right)^{-1} \frac{\partial D}{\partial X} \tag{2.10}$$

If $\hat{X} > 0.5$ then it means that the extrema lies closer to a different sample point. In this case, the interpolation is performed. If we substitute equation (2.9) into (2.10), we obtain a function useful to determine the points with low contrast and reject them:

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \tag{2.11}$$

The locations with a $|D(\hat{x})|$ smaller than a predefined threshold are discarded.

The DoG produces a strong response along the edges, but the locations along the edges are poorly determined and could be unstable even with small amount of noise. So, a threshold to discard the points poorly defined is essential. Usually a poorly defined peak in the DoG has large principal curvature across the edge and small curvature in the perpendicular direction. The principal curvatures are computed from a 2×2 Hessian matrix H estimated at the location and scale of the keypoint:

$$H = \begin{pmatrix} D_{xx} & D_{yx} \\ D_{xy} & D_{yy} \end{pmatrix} \tag{2.12}$$

The derivatives are estimated by taking differences of neighboring sample points. The eigenvalues of H are proportional to the principal curvatures of D. Let α be the eigenvalue with the largest magnitude and β be the smallest one. We can compute the sum and the product of the eigenvalues from the trace and from the determinant of H:

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \tag{2.13}$$

$$Det(H) = D_{xx}D_{yy} - (D_{yy})^2 = \alpha\beta \tag{2.14}$$

Let r be the ratio between the largest eigenvalue and the smallest one, then:

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \tag{2.15}$$

To check that the ratio of principal curvatures is below some threshold, we need to check whether

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \tag{2.16}$$

A set of scale-invariant points is now detected, but as we stated before we need locations invariant also to the rotation point of view and this goal is reached by assigning to each point a consistent local orientation. The scale of the keypoint is used to select the Gaussian smoothed image L with the closest scale, so that all computations are performed in a scale-invariant manner. For each image sample $L(x, y)$ at this scale, the gradient magnitude $m(x, y)$ and the orientation $\theta(x, y)$ are evaluated using pixel differences:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{2.17}$$

$$\theta(x,y) = tan^{-1}(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}) \tag{2.18}$$

A region around a sample point is considered and an orientation histogram is created. This histogram is composed by 36 bins in order to cover all the 360 degrees of orientation (each bin holds 10 degrees). Each sample added to the histogram is weighted by its gradient magnitude and by Gaussian-weighted circular window. The highest peak of the histogram is detected and together with the peaks within the 80% of the main peak is used to create a keypoint with that orientation.

In the last step of the method proposed by Lowe, at each keypoint, a vector is assigned which contains image gradients to give further invariance, especially with respect to the remaining variations (*i.e.*, change in illumination and 3D viewpoint), at the selected locations. The gradient magnitude and the orientation at each location are computed in a region around the keypoint location to create the keypoint descriptor. These computed values are weighted by a Gaussian window. They are then accumulated into orientation histograms summarizing the contents over 4×4 subregions, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. The descriptor is formed as a vector, which is made up by the values of all the orientation histogram entries.

We will adopt the common 4×4 array of histograms with 8 orientation bins, which means that the feature descriptor will be composed of $4 \times 4 \times 8 = 128$ features. Finally, the descriptor is normalized to unit length to reduce the effects of illumination change. Any change in contrast in a pixel value multiplied by a constant will multiply gradients by the same constant, so this contrast change is cancelled by vector normalization. At the end of the feature extraction procedure we are able to associate at each keypoint two vectors: the first one contains four values: two spatial positions, orientation and scale value and the second vector conveys the 128 features of the detectors. We will use the first vector to determine the spatial position of the keypoints inside the image and the second one to perform the classification of such keypoints.

Once the set of keypoints, with their respective descriptors, is extracted, the goal of the next stage of the process is the discrimination between keypoints which belong to cars and keypoints which represent all other objects ("background"). Since the dimension of the extracted features is relatively large, it is recommended to adopt a suitable classification method such as the Support Vector Machine (SVM) classifier. Before applying a classification based on a SVM classifier, we will add further information to the keypoint descriptor in order to potentially improve its discrimination power.

The first six features we will add are related to color information. Indeed, we think the addition of some proprieties strongly associated to the object itself can lead to a better discrimination. Even if car colors can be very heterogeneous, in numerous cases, their colors appear dissimilar to the appearance of dominant objects in the contextual environment (*e.g.,* asphalt, houses and green areas). For this reason, we think that the use of features linked to colors spaces can help in the discrimination. In particular, we will add information related to the most common color representation system, namely the RGB system, and information from a representation system commonly used in the computer vision field, *i.e.,* the HSV system. Differently from the standard RGB system, HSV has a cylindrical representation and was introduced to yield a more intuitive and perceptually relevant representation than RGB. In HSV, instead of using the three primary colors (red, green and blue) for the representation as in RGB, a color is represented by three parameters h (hue), s (saturation) and v (value). These three parameters can be simply computed starting from the three main colors and using three basic transformations [51]. Hence, the first six features we select to add to the descriptor of each keypoint are the three primary colors and the three components of the HSV system.

Recently, in high resolution satellite imagery, morphological operators have shown particularly effective in boosting the classification accuracy [52]- [53]. We think that such operators could be even more useful in UAV images where object geometry plays a more significant role (compared to satellite images). Consequently, in addition to the six color features, we will include also morphological features. We first apply a cascade of dilation filters and then a cascade of erosion filters on image $I(x, y)$. In both operations, at each step of the cascade, the SE used becomes bigger to produce two sets of images having different resolutions. These two sets give us the opportunity to characterize each keypoint at different resolutions obtaining more information about the same spatial position. The cascade of erosion and dilation filters is characterized by three steps. Accordingly, at the end of the filtering steps, the two sets of images are composed as follows:

$$S_d = \{Id_{SE1}, Id_{SE2}, Id_{SE3}\} \tag{2.19}$$

$$\mathrm{S_e} = \{\mathrm{Ie_{SE1}}, \mathrm{Ie_{SE2}}, \mathrm{Ie_{SE3}}\} \tag{2.20}$$

where S_d and S_e contain the dilated and eroded images respectively, with growing sizes of SE.

Considering these two sets, 18 new features (the RGB values at each scale) are added to the descriptor of each keypoint forming final descriptors composed of 152 features (Fig. 2.3).

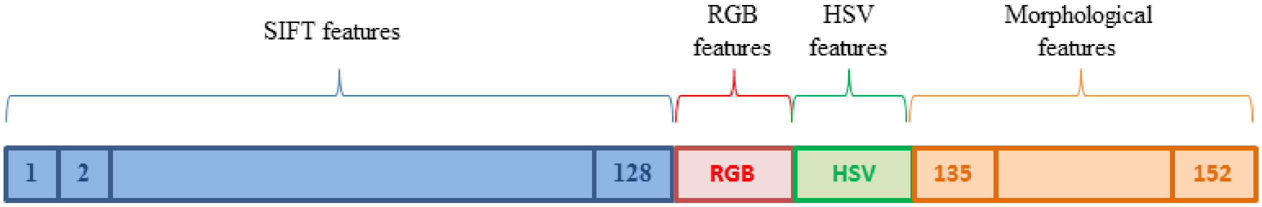


Figure 2.3. Structure of the SIFT descriptor.

Once the features are determined, the “car” versus “background” discrimination problem can be faced. As mentioned before, this problem will be handled by means of another SVM classifier, which will produce two sets of keypoints (*i.e.*, car and background keypoints). Since we are interested only in one set (the set composed of car keypoints), the background keypoints are discarded.

At the end of the keypoint classification procedure, the number of keypoints associated to the car class can be larger than the number of cars itself. The reason is that it is likely that a single car is marked by more than one keypoint. Let $K_c = \{k_1, k_2, \dots, k_N\}$ be the set of N keypoints found for the car class in the considered image $I(x, y)$, the goal is to estimate the number of cars present in $I(x, y)$ and to identify them in an univocal manner. To pursue this scope, we will develop an algorithm to group the keypoints which belong to the same car. Since the merging is performed in the spatial image domain, it will rely on a merging criterion based on a spatial distance between the keypoints in order to identify neighboring keypoints and possibly merge them into a unique keypoint representing the car on which they lie (see Figure 2.4).

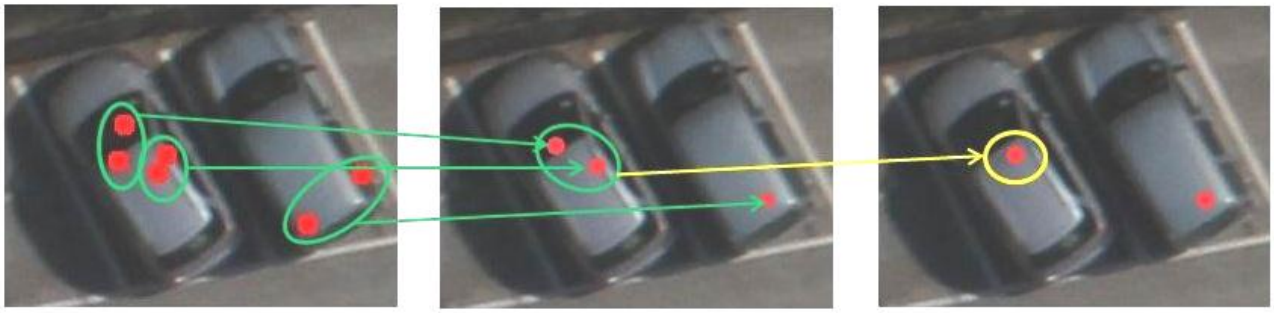


Figure 2.4. Example illustrating the merging process.

In the following, the main steps of our merging algorithm are summarized:

Step 1. The spatial coordinates of the keypoints contained in the set $\mathrm{K_c}$ are used as input of the algorithm;

Step 2. To the vector of parameters, a further parameter m is added and initialized to 1. It will act as a counter to keep trace of the number of “merging operations” done with that keypoint;

Step 3. A matrix $N \times N$ containing the Euclidean distances in the spatial domain between all keypoints is computed;

Step 4. The two keypoints (k_i, k_j) with the smallest distance d_{min} are selected;

Step 5. If $d_{min} < T_m$ (threshold) → k_i and k_j are merged into a new point k_t which will replace the two keypoints in the set K_c;

Step 6. The matrix containing the distances is then recomputed with the new point;

Steps 3-6 are repeated until $d_{min} > T_m$;

Step 7. Assuming that the points with a value of m smaller than two are isolated points only the points with $m > 1$ are kept. The number of resulting merged keypoints represents finally the estimation of the number of cars present in the scene. This step is useful to detect isolated keypoints and discard them since viewed as false alarms.

The value of the threshold T_m needs to be estimated by relating the expected car dimensions with the actual spatial resolution of the considered images. For instance, if the expected car width is around 180 [cm] and the sensor resolution is of 2 [cm], the best threshold value could be empirically searched for around 90.

In order to take into account the number of times each keypoint has contributed in the merging up to a current iteration of the algorithm, a weighted merging is implemented. Let k_i and k_j be the two keypoints with the smallest distance and k_t the new point coming from the merging of the two keypoints, the new parameters of k_t will be:

$$X_t = \frac{X_i \cdot m_i + X_j \cdot m_j}{m_i + m_j} \tag{2.21}$$

$$Y_t = \frac{Y_i \cdot m_i + Y_j \cdot m_j}{m_i + m_j} \tag{2.22}$$

$$\theta_t = \frac{\theta_i \cdot m_i + \theta_j \cdot m_j}{m_i + m_j} \tag{2.23}$$

$$s_t = \frac{s_i \cdot m_i + s_j \cdot m_j}{m_i + m_j} \tag{2.24}$$

$$m_t = m_i + m_j + 1 \tag{2.25}$$

2.5. Detecting Cars in UAV Images with a Catalogue-Based Approach

The key idea, which guides the second methodology (see Figure 2.5), consists to describe the image with features highlighting the shape of the objects and to identify the presence of cars by exploiting this representation and opportune similarity measures computed on the basis of a catalog of cars. The second step of the process, which follows the asphalt screening procedure, is dedicated to the extraction of features useful for the identification of cars. Since we are dealing with extremely high resolution images, a feature representation adapted to this type of images is necessary. In this work, we will opt for the Histogram of Oriented Gradients (HoG) features [54] which have been demonstrated to be suitable for the detection of complex classes of object, in particular for pedestrian detection. Generally, cars exhibit high diversity in terms of texture and color but have close shapes (Figure 2.6). This common property of cars makes intuitive the use of features like HoG which emphasize on structural information.

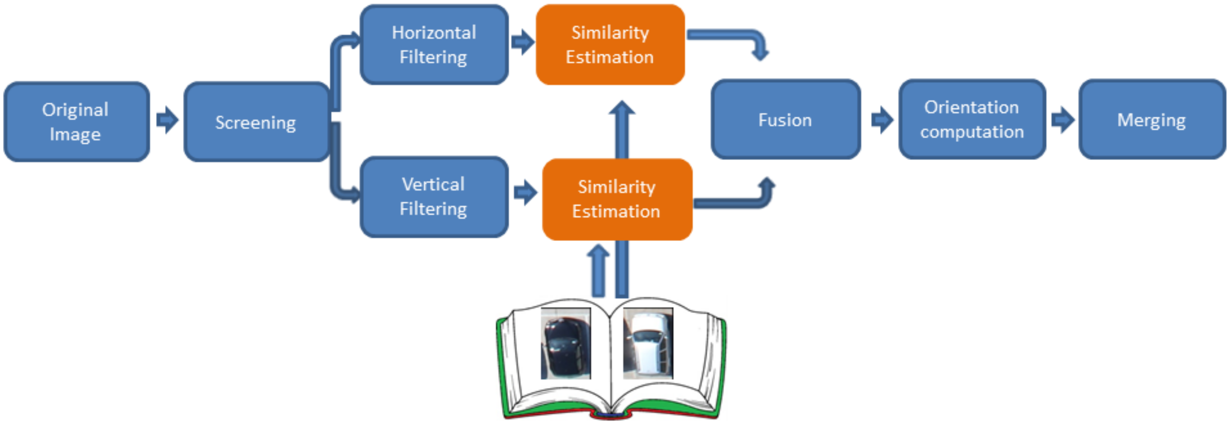


Figure 2.5. Flow chart of the proposed car detection methodology.

HoG features are computed by dividing the analyzed image into a set of overlapping cells and by extracting from each cell histograms of gradient directions. The histograms are grouped together in order to form the descriptors, which are then normalized to make them invariant to illumination and shadowing changes over blocks of $N \times N$ cells (see Figure 2.7). Standard HoG parameters for generating the histograms are: 9 bins on cells of 12 × 12 pixels, block size of 2 × 2 cells overlapping by one cell size.

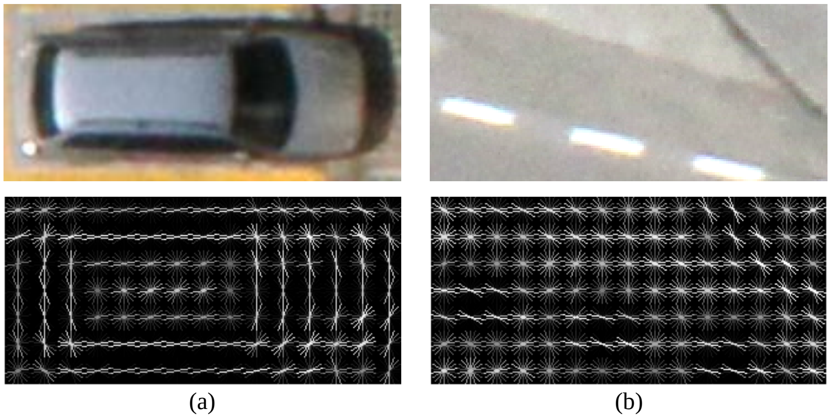


Figure 2.6. Example of window samples: (a) car and (b) background. In the first row the original images, while in the second one the HoG features.

From the investigated image, HoG features are extracted by means of two sliding window processes: one horizontal and the other vertical. This double extraction process is applied to make the detection more efficient and robust since cars could be found with any orientation. At the end of the filtering steps, each point of the image (window center) is characterized by two sets of features related to the horizontal and vertical information, respectively. The size of the window used to extract horizontal and vertical features needs to be estimated by relating the expected car dimensions with the actual spatial resolution of the considered image. For instance, if the expected standard car length and width are around 4.5 [m] and 1.8 [m], respectively, and the sensor resolution is 2 [cm], a suitable window size could be around 180 × 80 pixels.

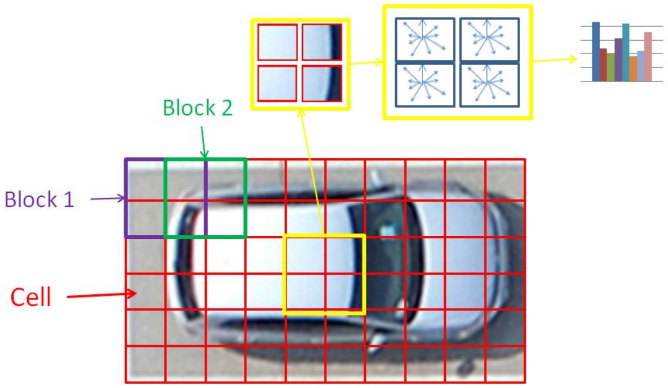


Figure 2.7. Extraction of the HoG descriptor from a sample image.

The next step aims at evaluating which points are likely to be associated to cars. The discrimination between "car point" and all the others is performed in three different ways. The first two strategies assign to each point two similarity values (i.e., horizontal and vertical) by comparing the horizontal and vertical features with a catalogue of cars (Figure 2.8) of size N (number of cars) beforehand converted into HoG feature vectors and by taking the mean values of the similarity measures computed in the two directions. The similarity measures used in these two strategies are the normalized cross-correlation and the mutual information measures, respectively.

Figure 2.8. Examples of cars composing the catalogue.

The normalized cross-correlation measures the relationship between two random variables. Let $X(x,y)$ be the HoG feature vector of length d and extracted at pixel coordinates (x,y) and $Y^{(k)}$ be the k-th template (car) in the catalogue:

$$r\left(X;Y^{(k)}\right) = \frac{\sum_{i=1}^{d}(X_i-\bar{X})\cdot\left(Y_i^{(K)}-\overline{Y^{(k)}}\right)}{\sqrt{\sum_{j=1}^{d}(X_j-\bar{X})^2\cdot\sum_{j=1}^{d}\left(Y_j^{(K)}-\overline{Y^{(k)}}\right)^2}} \quad \text{where} \quad \begin{cases}\bar{\mathrm{X}} = \mathrm{mean(X)} \\ \overline{\mathrm{Y^{(k)}}} = \mathrm{mean(Y^{(k)})}\end{cases} \quad (2.26)$$

The correlation measure is very popular in the computer vision field, in particular for image matching, because of its simplicity, implementation easiness and real-time application suitability.

The mutual information measures the mutual dependence between two variables in the following way:

$$I\left(X;Y^{(k)}\right) = \sum_{X_i}\sum_{Y_j^{(k)}} p\left(X_i,Y_j^{(k)}\right) log\left(\frac{p\left(X_i,Y_j^{(k)}\right)}{p(X_i)p\left(Y_j^{(k)}\right)}\right) \quad (2.27)$$

where $p\left(X_i,Y_j^{(k)}\right)$ is the joint probability function, $p(X_i)$ and $p\left(Y_j^{(k)}\right)$ are the marginal distribution probability of X and $Y^{(k)}$, respectively. Mutual information, or relative entropy [55]- [56], does not assume prior functional relationship between the current window and the considered template but it describes their statistical relationship by using the joint entropy. Being usually efficient and robust to outliers, it has been widely exploited especially for image registration [57]. In our case, since the mutual information measure refers to the comparison between the two HoG feature vectors (X and $Y^{(k)}$, respectively), the joint and marginal distributions are estimated through simple relative frequency computation.

Once the similarity measure is computed by means of one of the above two strategies, the discrimination between car and background classes is carried out thanks to a thresholding operation performed in both directions. A point corresponds to a car if its similarity value is larger than a predefined threshold value.

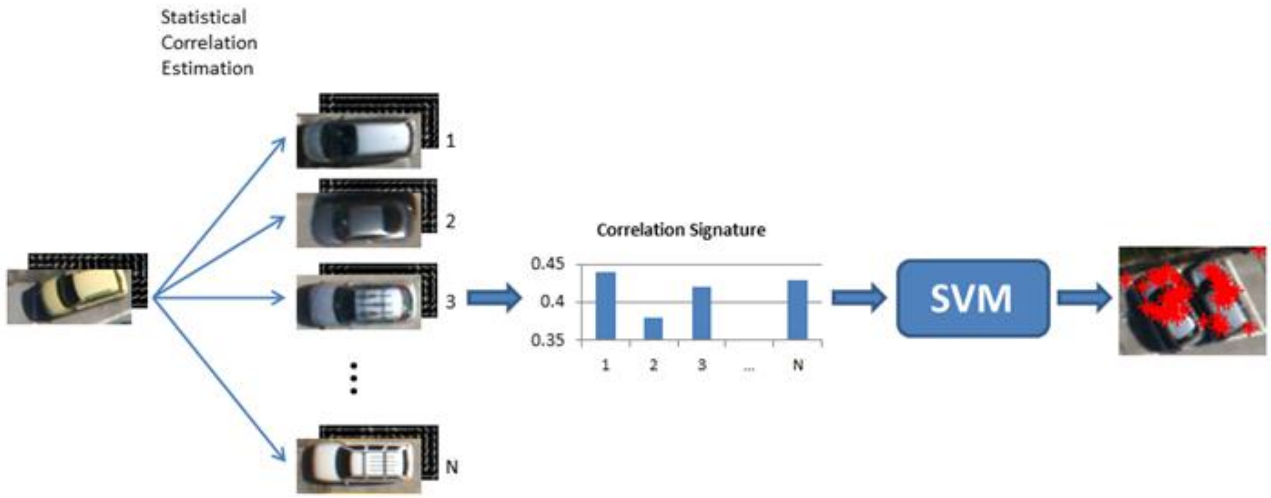


Figure 2.9. Illustration of signature generation for SVM-based similarity computation.

The third similarity measure combines the normalized cross-correlation measure with a Support Vector Machine classifier (SVM [40] [41] [42] [43] [44]. In this strategy, for each sliding window position, two correlation signatures (*i.e.*, horizontal and vertical) are assigned (see Figure 2.9). The signatures are generated by computing the correlation between the considered window and each car of the catalogue in the two directions. As it can be shown, when the window covers a car, the distribution takes a quadrimodal behavior because of symmetries. By contrast, if the window does not cover a car, it is expected the distribution to be arbitrary. Thus, the dimension of the signature is equal to N (*i.e.*, the number of templates in the catalogue). Depending on the dimension of the catalogue, it is possible to handle signatures characterized by large dimensions. Therefore it is suitable to adopt a classifier which is able to handle satisfactorily high dimensional data. SVM classifiers have demonstrated to be particularly recommended for such situations. Differently from the two previous strategies, here, an additional class of objects is added. By observing that pedestrian crossings present a structural shape similar to that of cars and by exploiting the capacity of SVM classifiers to discriminate between more than two classes, we included in the analysis also the pedestrian crossing class. Thanks to this extension, we expect to reduce the number of false alarms.

SVMs are intrinsically binary classifiers. They can however easily be adapted to multiclass problems by adopting binary decomposition tricks such as the One-Against-One (OAO) method [42]. Very briefly, the binary SVM starts by assuming a training set defined in a N-dimensional feature space, each sample z (in our case, the vector of similarity measures) being properly labeled $\{-1, +1\}$. During its training, the goal of the SVM classifier is to find the hyperplane characterized by a weight vector w and a bias b in a kernel-induced feature space ($\Phi(X) \in \Re^{N'}, N' > N$)) which best separates the two considered classes. From this hyperplane, it is possible to derive a discriminant function f(z) used to take the class assignment decision:

$$f(z) = w^{*}\Phi(z) + b^{*} \tag{2.28}$$

At the end of the classification, the output of this step consists of an image conveying potential car points derived from horizontal and vertical filtering by applying a simple "OR" fusion operator. By contrast, if the normalized cross-correlation or the mutual information are used, the images that represent the potential car points are generated by using an "AND" fusion operator. This difference in the choice of the fusion operator is motivated by the less accurate discrimination of potential car points produced by these last measures with respect to the SVM. Therefore, a more precautionary operator is preferred to reduce the risk of false alarms.

In this second methodology, in addition to identifying the position (*i.e.*, spatial coordinates) of cars, we propose the estimation of their orientation. This is done by taking again advantage of the car catalogue and of the first two similarity measures defined above (i.e., correlation and mutual information). Around each point classified as car, a mask, with dimensions equal to those used for the filtering step, is rotated in 36 directions (*i.e.,* 10-degrees step), and for each direction the similarity measure is computed for each car of the catalogue. The direction with the highest average similarity measure is identified as the main car orientation (Figure 2.10). After the orientation assignment step, only the points which present an average similarity measure higher than a predefined threshold (*i.e.*, T_{SM}) are considered for the next step. We expect this threshold should assume a higher value with respect to the one used throughout the previous filtering step. Indeed, since the filtering operations are performed in two directions (*i.e.*, horizontal and vertical) it is necessary to keep smaller threshold values in order to detect cars with orientations which are not necessarily horizontal or vertical. In the orientation assignment step, the threshold value can be raised because the detection is performed along much more directions.

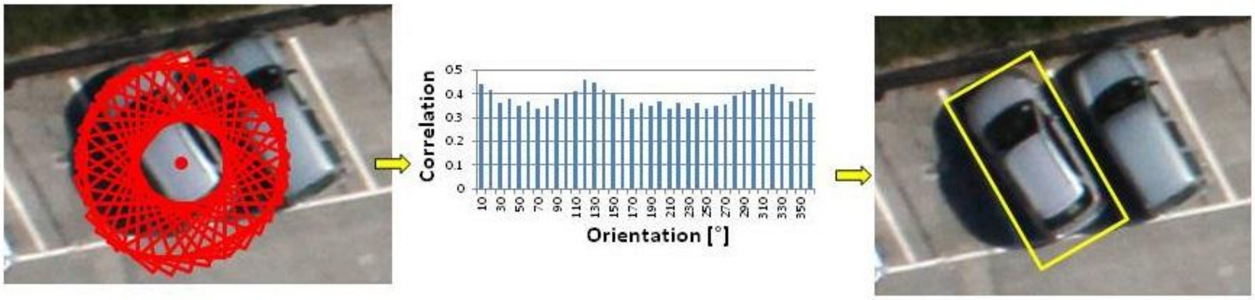


Figure 2.10. Illustration of the orientation assignment step.

At the end of the orientation assignment step, we have to deal with a set of points, classified as car points and characterized by four parameters: spatial coordinates (*i.e.*, x and y), similarity measure (*i.e.*, *sm*) and the main orientation (*i.e.*, o). Similarly to the first proposed car detection approach, the number of points associated to the car class can be larger than the number of cars itself. In this case, the reason is that it is likely that more than one window in the detection stage cover the same car due to the small step-size of the sliding window procedure. Let $K_c = \{k_1, k_2, \dots, k_N\}$ be the set of N points found for the car class in the considered image $I(x, y)$, the goal is to estimate the number of cars present in $I(x, y)$ and to identify them in an univocal manner with the correct orientation.

Therefore, in the end, as done for the first methodology, we will make use of the same merging algorithm to group the points which potentially belong to the same car and obtain an univocal estimation about the number of cars.

2.6. Experimental Results

In this section, we present the results of the various simulations conducted on a set of real UAV images acquired over an urban area. The following results aim at testing and comparing the two presented car detection strategies.

2.6.1. Dataset Description and Experimental Setup

In the context of these works, we use a set of images acquired by means of an UAV equipped with imaging sensors spanning the visible range. The images were taken over the Faculty of Science of the University of Trento (Italy) on the 3rd October 2011 at 12:00 am. Nadir acquisitions were performed with a picture camera Canon EOS 550D characterized by a CMOS APS-C sensor with 18 megapixels. The images are characterized by three channels (RGB) and by a spatial resolution of approximately 2 cm. All the acquired images have sizes of 5184×3456 pixels with 8 bits of radiometric resolution. From the whole dataset, we identified 10 images (see Figure 2.11) and we divided them into three groups:

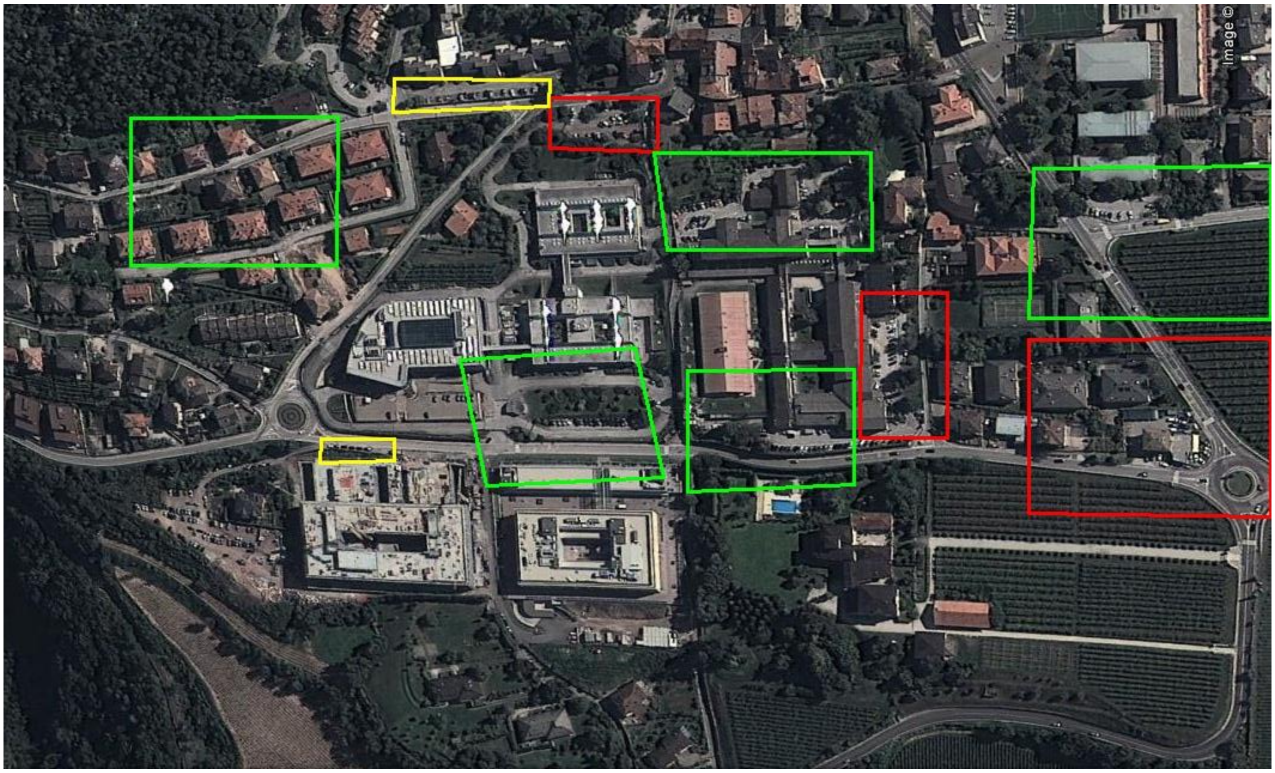


Figure 2.11. View of the whole scene. Red rectangles represent the training images, rectangles in yellow are the validation areas and green rectangles are the test images.

a) *Training Group*. It is composed by 3 images (*i.e.*, red rectangles in Figure 2.11). Inside the images, it is possible to recognize 80 cars (26 in the first image, 21 in the second and 30 in the third image). For both car detection strategies, these images are used for the training of the SVM classifier devoted to the screening step. In particular, to train the classifier, we collected 30807samples divided into: 12463 samples of asphalt and 18344 samples of background. This classifier is fed with three features, namely the three original color components (red, green and blue) of the acquired images. Moreover, this group of images is used also for the training of the SVM classifier for the discrimination of the keypoints in the first strategy and for the creation of the catalogue and for the training of the SVM classifier for the horizontal and vertical filtering direction in the second methodology. The SVM classifier of the first strategy devoted to the classification of keypoints was trained with 28632 SIFT keypoints, of which 1171 represent car keypoints and 27461 are the background keypoints. On the other hand, the catalogue of the second methodology is composed by 58 cars. They are used for the training of the SVM classifier for the horizontal and vertical filtering directions. In particular, by using a step size equal to 10 for the filtering steps, we extracted 33414 and 37701 points for the horizontal and vertical directions, respectively. By using manually-built masks (*i.e.*, car masks and pedestrian crossing masks), we constructed the training set of samples representing car, pedestrian crossing and background. For such purpose, assuming that if a window is covered for more than 30% by the car mask or by the pedestrian crossing mask, the corresponding sample is considered car or pedestrian crossing, respectively. Otherwise, it is classified as background. This 30% threshold value was found as a good compromise between number of training points and false alarm risk. From the two training sets, we identified 1977 and 1922 car windows for horizontal and vertical filtering, respectively. Instead for what concerns the pedestrian crossing class, it was possible to obtain 364 and 354 points for horizontal and vertical filtering, respectively.

b) *Validation Group*. It is made up of 2 images (*i.e.*, yellow rectangles in Figure 2.11). The images belonging to this group represent two parking lots; in which the cars (6 in the first image and 17 in the second one) are easily recognizable. This group will be used to calibrate all free parameters. In particular, all parameters regarding the SVM classifiers are tuned with respect to these two images. The free parameters of the SIFT algorithm and the sizes of the structural elements used in the first procedure and the thresholds of the orientation assignment and the values of the discrimination threshold used in the first two similarity estimation strategies exploited in the second methodology are estimated using the validation images. Furthermore the T_m threshold used for the merging stages in both strategies is determined using this group of images. This step is particularly important since it allows choosing threshold values generalizable to other UAV flights at similar acquisition conditions. The selected values are those which empirically produce the best ratio between number of cars correctly identified and number of false alarms.

c) *Test Group*. It is composed by 5 images (*i.e.,* green rectangles in Figure 2.11). This is the group on which we assessed the accuracy of the developed methodology. It includes different images in order to test the method in different conditions. We selected two images representing two standard urban areas with a medium density of cars (19 and 15 cars), two images representing two big parking lots full of cars (51 and 31 cars, respectively) and an image with only 3 cars to assess our technique in a situation where the presence of cars is rare and the presence of other objects (*e.g.,* solar panels and dormers) could affect the automatic analysis.

We organized the works in two main stages. The first part is devoted to the training and to the calibration of the parameters, thus we worked only with the first two groups of images. Once the training of the SVMs and the estimation of the free parameters were completed, we moved to the test part in which we collected the final results. In greater detail, coming back to the training and validation steps, for the screening operations, we had to set the parameters of the SVM classifier, i.e., the regularization parameter C and the kernel parameter γ. We found, by 5-fold cross-validation, that the best values for these parameters are 500 and 2, respectively. Another issue in the screening operation is the correct choice of the shape and the sizes of the SE used for the morphological filters. We observed that the disk is the best SE that could be used for both erosion and dilation operation. However, the size (the radius) of the disk used in the two operations is different. For the erosion phase, we decided to use a disk with dimension 15 (30 [cm]) in order to remove most of the noise. By contrast, the SE involved in dilation step has a dimension of 150 (300 [cm]). This specific choice has been done by observing the original mask obtained before the morphological refinement. The screening aims at the identification of all the asphalted areas, thus, without the morphological improvements there could be misclassification problems especially due to cars present on the sides of the roads or to holes left on the asphalted areas by the screened cars.

Considering the classification procedures, additional parameters are involved and differ depending on the car detection strategies exploited. In the case the first strategy the parameters to be estimated are those of the second SVM classifier (C and γ) estimated by cross-validation, and those of the feature extraction procedure (the peak threshold $|D(\hat{x})|$, the value r of edge threshold $(r+1)^2/2$ determined by maximizing the accuracy measure, Acc, on the validation images. For the SVM parameters, the estimated values of C and γ are 25 and 0.25, respectively. For $|D(\hat{x})|$, the parameter which filters small peaks in the DoG scale space, the best extracted value is equal to 1, while for r, the best value which eliminates peaks of the DoG scale space with small curvature, is equal to 9. Working with the second methodology, the parameters that have to be estimated are those regarding the SVM classifiers (C and γ) and the value of T_{SM}. We found, by 5-fold cross-validation, that the best values for the SVM classifier are 75 and 2 for the horizontal filtering and 100 and 2 for the vertical filtering. By observing the results on the two validation images, we decided to consider a point as car if it is correctly classified with a confidence value (posterior probability derived with Platt's function [58]) higher than 95%. This allows us to obtain a high level of correct car detection while considerably reducing the number of false alarms. Furthermore, we had to estimate the thresholds of the similarity measures for the discrimination of the car points from those that belong to the background class. We found that the best values are 0.32 and 0.06 for the normalized cross-correlation and the mutual information measures, respectively. The values of T_{SM} were also determined in an empirical way. We found a value of 0.44 for the correlation measure and a value of 0.105 for the mutual information measure. For both car detection procedures, the parameter $\mathrm{T_m}$ has been determined by doing a simple assumption: generally a car has a width of about 1.80 [m], a length of about 4.50 [m] and usually between two cars, parked or along roads, there is at least 1 [m]. Accordingly, we selected a T_m value of 80 (1.60 [m]).

All the results are presented with producer's accuracy versus user's accuracy. Producer's accuracy (*i.e., Pacc*) shows the number of correct points with respect to the real number of points (and consequently it considers the missed alarms), instead the user's accuracy (*i.e., Uacc*) compares the number of correct points with the number of identified points:

$$Pacc = \frac{TP}{M} \tag{2.29}$$

$$Uacc = \frac{TP}{TP + FP} \tag{2.30}$$

where *TP* is the number of cars correctly identified, *FP* is the number of false alarms and *M* represent the number of cars really present in the scene. Furthermore, we present also the average of the two accuracies (*i.e., Acc*) in order to have a complete and fast overview about the goodness of the method for each investigated image.

Another important goal pursued in the second methodology is the estimation of position (*i.e.,* spatial coordinates) and orientation of each car. To evaluate position and orientation errors, we extracted manually the true spatial coordinates and orientation from the test images in order to compare them with those obtained at the end of the merging step for each true positive. In particular, let $(B_{xk},\ B_{yk})$ and O_{true-k} be the coordinates and orientation, respectively, of the barycenter of the k-th car and $(X_k,\ Y_k)$ and O_k be its coordinates and orientation, respectively, estimated at the end of the merging step, the position and orientation errors are determined as follows:

$$coords_{err(k)} = \sqrt{(B_{xk} - X_k)^2 + (B_{yk} - Y_k)^2} \tag{2.31}$$

$$orient_{err(k)} = |O_{true-k} - O_k| \tag{2.32}$$

All the experiments were conducted on an Intel Core i7-3537U CPU @ 2 GHz with 8 GB RAM and by using Matlab. The results of the proposed methodologies are resumed in the following sections. For reasons of clearness and completeness, in the next two sections the results of each strategy are presented singularly, whereas in the last section they are compared.

2.6.2 Results of the First Method

For what concern the first strategy, we will report the results of all the stages of the methodology starting from the validation step and concluding with the final results. At the beginning, we will show the results of feature extraction and classification steps to allow us to evaluate each single step. Then, for the test step, we will show only the final results of the procedure because they are the ones to which we are really interested. We will also report the final results considering different kinds of screening: without screening, with automatic screening and with GIS-based screening. This comparison is useful to understand the importance of this step, by analyzing its impact on the other stages.

TABLE 2.I. SCREENING ACCURACIES IN PERCENT BEFORE AND AFTER THE APPLICATION OF THE MORPHOLOGICAL OPERATIONS.

	Before Morphology		**After Morphology**	
	Asphalt Accuracy	**Background Accuracy**	**Asphalt Accuracy**	**Background Accuracy**
Image Test 1	33.07	90.91	74.59	84.77
Image Test 2	26.38	90.97	64.21	69.95
Image Test 3	46.61	85.64	83.76	48.86
Image Test 4	58.37	93.13	88.33	74.30
Image Test 5	62.57	95.94	99.83	85.33

Before analyzing the final results, we will first underline the importance of the morphological operations performed on the masks obtained at the end of the screening phase and the usefulness of the addition of color and morphological features to the original SIFT descriptors. By observing the results reported in Table 2.I, we can notice how the accuracies after the application of the morphological filters are substantially improved. The average asphalt accuracy obtained before the morphological stage is about 45.4%. It becomes 82.1% after the morphological operation. This substantial gain of accuracy compensates widely the drop of accuracy incurred for the background class (from 91.26% to 72.64%). This will be further supported by the final results discussed later. The reason behind our decision to integrate the SIFT description vector, obtained at the end of the feature extraction procedure, takes origin from the results achieved with extra color and morphological features. Indeed, the addition of 24 features (*i.e.,* RGB, HSV and 3×6 morphological features) allows improving the results of the keypoint classification as can be seen in Table 2.II. One moves from a correct detection of 178 car keypoints, by using only the 128 original SIFT features, to a right detection of 495 car keypoints by integrating the original descriptors with the above mentioned 24 features. This feature integration conveys also the advantage of making the detection process more robust since it increases the number of car keypoints found for each car, and thus renders more likely that isolated car keypoints represent false alarms.

TABLE 2.II. (a) KEYPOINT CLASSIFICATION ACCURACY IN PERCENT AND (b) CAR KEYPOINT DETECTION AND FALSE ALARMS.

(a)

Features	Car	Background	Total
SIFT	27.72	98.49	**98.15**
SIFT + Color	50.07	98.61	**98.37**
SIFT + Morphology	52.75	98.67	**98.40**
SIFT + Color + Morphology	48.81	98.7	**98.34**

(b)

Features	Detection	False Alarms
SIFT	178	464
SIFT + Color	331	330
SIFT + Morphology	412	369
SIFT + Color + Morphology	495	519

The results of the keypoint classification are shown in greater detail in Table 2.III (a). Generally, for a car, it is possible to identify tens of keypoints but the classifier will not associate all of them to the car class and accordingly the car keypoint classification accuracy is not very high. Nonetheless, what it is really important is that all cars are characterized by at least some correct car keypoints. Indeed, to reach our goal, we are not particularly interested in the correct identification of all the keypoints belonging to the car class. To confirm this, we can look at the results reported in Table 2.III (b). Observing these results, we notice that we were able to detect 19 cars over the 23 cars present in the two validation images. This result confirms that a perfect discrimination of the keypoints is not mandatory to solve our problem. By tuning better the values of the parameters of the SVM classifier used for the discrimination of the keypoints, since background keypoints are dominant in the image, we can obtain better results in terms of total classification accuracy of background and car keypoints, but at the end of the whole detection process we may obtain worse results in terms of car detection. For this reason, we opted for parameter values which allow us to find the highest number of cars despite of a poorer result in terms of keypoint classification accuracy. Note that single non-car keypoints can be misclassified. If they are isolated, they will be anyway considered as false alarms by our method. If they are close to car-keypoints, they will be merged with them to correctly represent cars. By contrast, if misclassified non-car keypoints are spatially close to each other, after merging, they can incur in wrong car detections.

TABLE 2.III. (a) KEYPOINT CLASSIFICATION AND (b) CAR DETECTION ACCURACIES IN PERCENT OBTAINED ON THE VALIDATION IMAGES.

(a)

	Car Point accuracy	Background Point Accuracy	Total Accuracy
Validation	42	98.03	90.96

(b)

	Producer's Accuracy	User's Accuracy	Accuracy
Validation	82.61	86.36	84.49

At the end of the validation step, we moved to the test stage. First, we tested our procedure on the five test images without applying the procedure of screening in order to understand how well the classification stage works. The results of these first tests are resumed in Table 2.IV.

TABLE 2.IV. ACCURACIES IN PERCENT ACHIEVED ON THE TEST IMAGES WITHOUT SCREENING.

	TP (true positive)	FP (false positive)	N (cars present)	Producer's Accuracy	User's Accuracy	Total Accuracy
Image Test 1	43	32	51	84.31	57.33	**70.82**
Image Test 2	19	20	31	61.29	52.78	**57.03**
Image Test 3	16	60	19	84.21	21.05	**52.63**
Image Test 4	9	10	15	60.00	53.68	**47.37**
Image Test 5	1	21	3	33.33	4.55	**8.94**
TOTAL	88	143	119	73.95	38.10	**56.02**

TABLE 2.V. ACCURACIES IN PERCENT ACHIEVED ON THE TEST IMAGES WITH THE PROPOSED AUTOMATIC SCREENING.

	TP (true positive)	FP (false positive)	N (cars present)	Producer's Accuracy	User's Accuracy	Total Accuracy
Image Test 1	40	9	51	78.43	81.63	**80.03**
Image Test 2	15	6	31	48.39	71.43	**59.91**
Image Test 3	13	27	19	68.42	32.50	**50.45**
Image Test 4	9	8	15	60.00	52.94	**56.47**
Image Test 5	1	1	3	33.33	50.00	**41.67**
TOTAL	78	51	119	65.55	60.47	**63.01**

Analyzing these results, we can notice that despite a good capability of detection (88 cars over 119) there is a large number of false alarms (*i.e.*, 143). We can note also that the merging algorithm works satisfactorily because we are able to detect 74% of cars present in the images. The main problem regards the fact that we need to reduce the number of false alarms. We notice that most of the false alarms are concentrated:

1. On the solar panels present on the roof of the buildings. Due to the very high resolution of the images, keypoints found on the solar panels are associated by the classifier to the windscreen washer of cars;

2. On the fences which present a complex geometry similar to the geometry of cars.

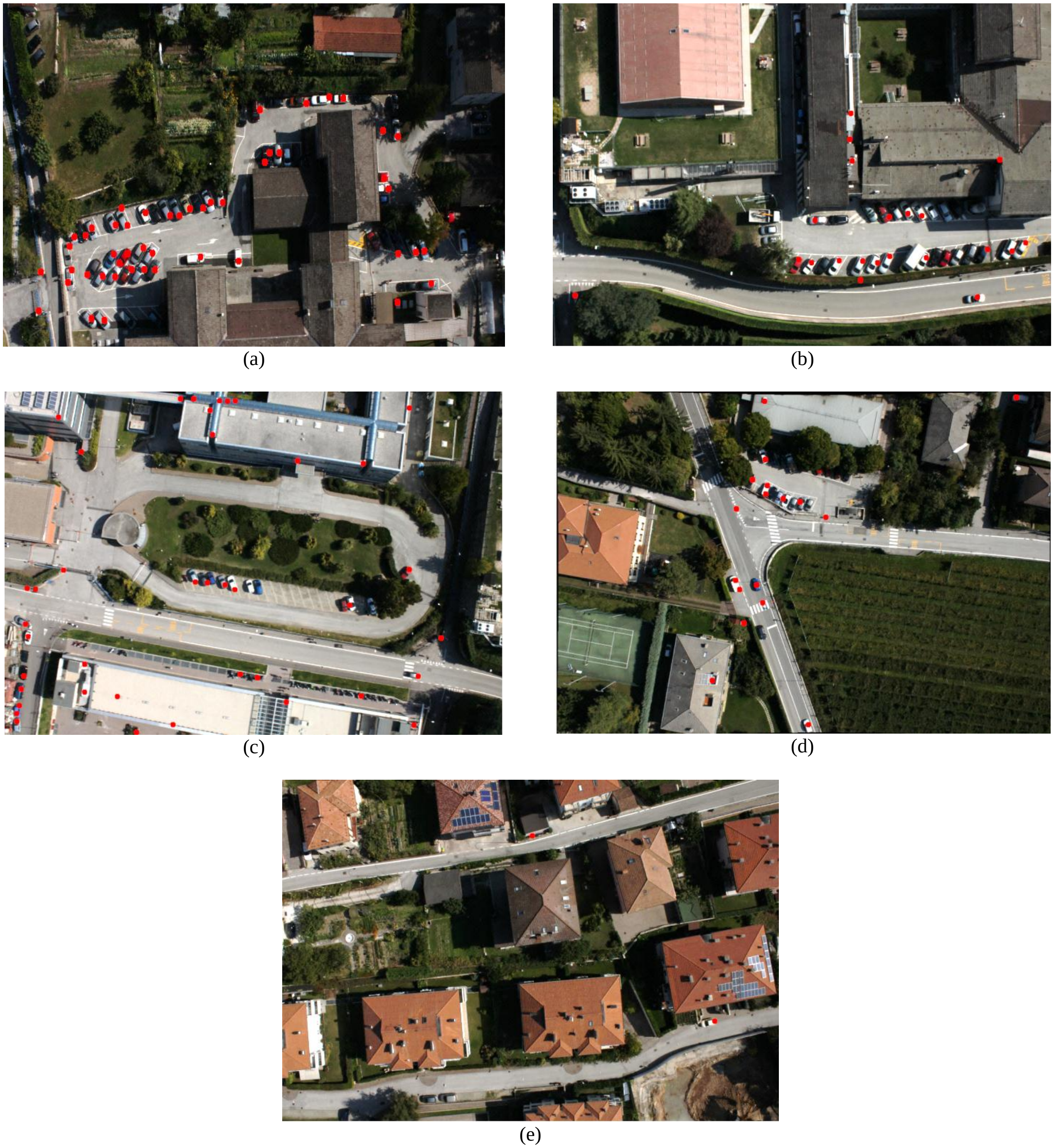
This makes us think that a screening of the images could produce better results because we should be able to eliminate the non-asphalted regions from our analysis and consequently to reduce the number of false alarms.

As expected, performing the automatic screening of the images allows us to improve the results as reported in Table 2.V and Figure 2.12. With respect to the previous case, we "lost" 10 cars but we have reduced of one third the number of false alarms. We eliminated most of the keypoints localized on the roofs and on the fences; the majority of the remaining false alarms (27 over a total of 51) are in the third test image. In this particular situation, the building roofs have a color similar to the asphalt and, at first glance, they really look like parking lots. The SVM classifies these areas as asphalt since it bases the classification on the sole three color components of the image. In Table 2.I, for the third test image, one can notice how the classifier provides poor results in terms of background classification accuracy (*i.e.,* 48.86%) because it confuses the roofs with asphalted areas due to their very similar natural appearances. The user's accuracy obtained on this test image is poor (*i.e.*, 32.5%) and it strongly affects the final accuracy. An accurate GIS-based masking could solve drastically these problems. Still considering these last results, we can observe that the lost cars are in particular those hidden under the shadows. These regions are not classified as asphalted regions and consequently are not considered in the search for car keypoints. It is possible to solve this problem still using a GIS-based screening.

TABLE 2.VI. ACCURACIES IN PERCENT ACHIEVED ON THE TEST IMAGES WITH IDEAL SCREENING.

	TP (true positive)	FP (false positive)	N (cars present)	Producer's Accuracy	User's Accuracy	Total Accuracy
Image Test 1	43	12	51	84.35	78.18	**81.25**
Image Test 2	19	3	31	61.29	86.36	**73.83**
Image Test 3	16	3	19	84.21	84.21	**84.21**
Image Test 4	9	5	15	60.00	64.29	**62.14**
Image Test 5	1	0	3	33.33	100	**66.67**
TOTAL	88	23	119	73.95	79.28	**76.61**

In the last part of our analysis, we assessed the proposed method using a GIS-based screening obtained with the use of road maps. Analyzing the results reported in Table 2.VI, we can observe a substantial improvement, as expected. The false alarms are further reduced and the number of detected cars is increased. Considering this ideal situation, the non-identified cars are especially those present over non asphalted areas, those strongly covered by the shadows and those partially covered by branches of trees. The cars, which are not covered by any kind of obstacles (*e.g.,* shadows) and have not been detected, are 16 of which 11 completely black. The black cars are not recognized because assimilated by the classifier to shadows. The false alarms present in this last situation are only 23 and most of them, precisely 14 false alarms, are caused by the double recognition of the same car (two keypoints characterizing the same car). Among the 31 missed alarms, only 4 cars (about 3% of the total number of cars) have not been detected because they are too near to other cars and their keypoints have been consequently merged. This result suggests that the value adopted for the merging threshold T_m is adequate since just very few cars have been lost. Reducing further its value would solve this problem (close cars) but at the same time increase the acuity of the above mentioned double recognition issue. It is worth noting that we were able to detect 11 cars partially covered by shadows or trees. This was possible thanks to the SIFT capability for retrieving occluded objects.



(a) (b) (c) (d) (e)

Figure 2.12. Final results obtained on the test images. (a) first image, (b) second image, (c) third image, (d) fourth image, (e) fifth image.

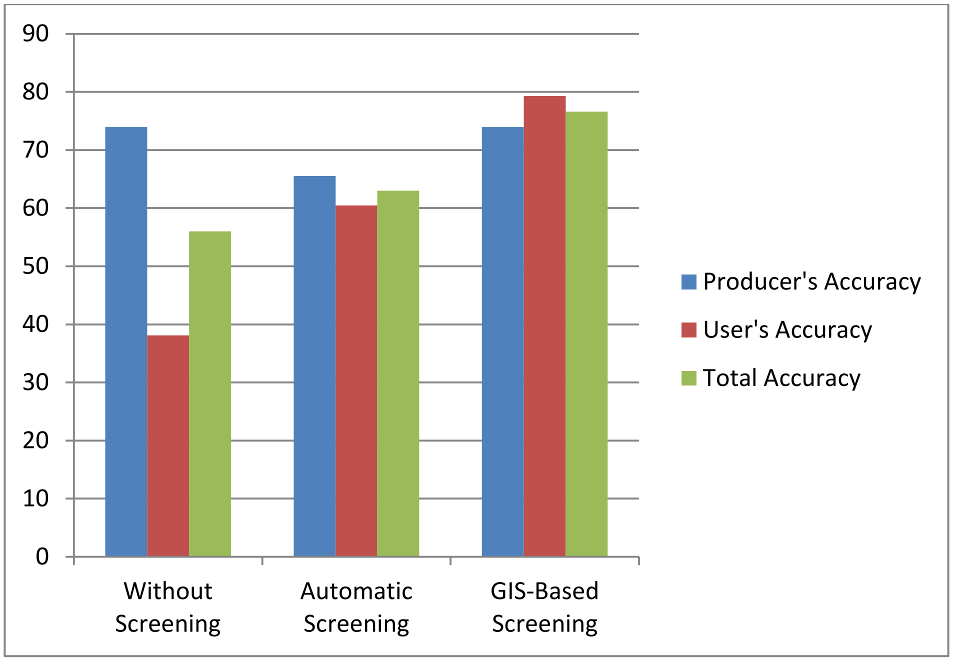


Figure 2.13. Accuracies achieved with the different screening scenarios.

To sum up the results regarding the first methodology, we can observe from Figure 2.13 that our merging algorithm works well and that the introduction of the screening step leads to detection improvements. Without any kind of screening, we can obtain good results in terms of producer's accuracy, which are however mitigated by a low user's accuracy.

2.6.3 Results of the Second Method

By taking into consideration the second methodology, we will report the final results showing first the performance obtained without the screening operation (Tables 2.VII-2.IX) and then with the GIS-Based screening (Tables 2.X-2.XII). In the previous subsection, during the discussion about the results obtained with the first methodology, we already investigated the importance of the screening procedure. For this reason, to avoid being repetitive and to better compare the most significative results we preferred to report only the achievements without and with the ideal screening.

In both situations (*i.e.*, without and with screening step), the best results are those obtained by using SVM classifier in the similarity computation, while the worst results are those obtained by means of the normalized cross-correlation as similarity measure without the screening operation and those obtained by using the mutual information with the screening step. In both situations, when the discrimination is carried out by considering the correlation as similarity measure, the number of cars detected is higher (*i.e.*, 69 versus 57), but this result is influenced by a large number of false alarms (*i.e.*, 265 versus 107 without screening and 57 versus 39 with the GIS-based screening).

By comparing the two situations, for all the three strategies, it is possible to observe how the car detection capability keeps unvaried. Indeed, the producer's accuracies of the three discriminant operations obtained without the screening step remain stable even when the masking is applied. The main difference between the two situations is represented by the number of false alarms. The false alarms contribute with the true positives to determine the user's accuracy and consequently the final accuracies are strongly conditioned.

TABLE 2.VII. ACCURACIES IN PERCENT ACHIEVED ON THE TEST IMAGES USING NORMALIZED CROSS-CORRELATION AS SIMILARITY MEASURE WITHOUT SCREENING.

	TP (true positive)	**FP (false positive)**	**M (cars present)**	**Producer's Accuracy**	**User's Accuracy**	**Total Accuracy**
Image 1	31	49	51	60.78	38.75	**49.76**
Image 2	22	49	31	70.97	30.99	**50.97**
Image 3	7	57	19	36.94	10.94	**23.88**
Image 4	7	45	15	46.67	13.46	**30.06**
Image 5	2	65	3	66.67	2.98	**34.82**
TOTAL	69	265	119	57.98	20.65	**39.32**

TABLE 2.VIII. ACCURACIES IN PERCENT ACHIEVED ON THE TEST IMAGES USING MUTUAL INFORMATION AS SIMILARITY MEASURE WITHOUT SCREENING.

	TP (true positive)	**FP (false positive)**	**M (cars present)**	**Producer's Accuracy**	**User's Accuracy**	**Total Accuracy**
Image 1	21	10	51	41.18	67.74	**54.56**
Image 2	18	19	31	58.06	48.65	**53.36**
Image 3	8	28	19	42.11	22.22	**32.16**
Image 4	9	29	15	60	23.68	**41.84**
Image 5	1	21	3	33.33	45.45	**18.94**
TOTAL	57	107	119	47.89	34.75	**41.33**

TABLE2.IX. ACCURACIES IN PERCENT ACHIEVED ON THE TEST IMAGES USING NORMALIZED CROSS-CORRELATION AND SVM CLASSIFIER WITHOUT SCREENING.

	TP (true positive)	**FP (false positive)**	**M (cars present)**	**Producer's Accuracy**	**User's Accuracy**	**Total Accuracy**
Image 1	35	10	51	68.63	77.77	**73.20**
Image 2	20	15	31	64.52	57.14	**60.08**
Image 3	16	30	19	84.21	34.78	**59.50**
Image 4	14	17	15	93.33	45.16	**69.24**
Image 5	2	39	3	66.67	4.87	**35.77**
TOTAL	87	111	119	73.11	43.39	**58.52**

TABLE 2.X. ACCURACIES IN PERCENT ACHIEVED ON THE TEST IMAGES USING NORMALIZED CROSS-CORRELATION AS SIMILARITY MEASURE WITH SCREENING.

	TP (true positive)	**FP (false positive)**	**M (cars present)**	**Producer's Accuracy**	**User's Accuracy**	**Total Accuracy**
Image 1	31	14	51	60.78	68.89	**64.84**
Image 2	22	11	31	70.97	66.67	**68.81**
Image 3	7	13	19	36.94	35	**35.92**
Image 4	7	22	15	46.67	31.81	**39.24**
Image 5	2	4	3	66.67	33.33	**50**
TOTAL	69	57	119	57.98	54.76	**56.37**

TABLE 2.XI. ACCURACIES IN PERCENT ACHIEVED ON THE TEST IMAGES USING MUTUAL INFORMATION AS SIMILARITY MEASURE WITH SCREENING.

	TP (true positive)	FP (false positive)	M (cars present)	Producer's Accuracy	User's Accuracy	Total Accuracy
Image 1	21	8	51	41.18	72.42	**56.79**
Image 2	18	7	31	58.06	72	**65.03**
Image 3	8	8	19	42.11	50	**46.05**
Image 4	9	13	15	60	40.91	**50.45**
Image 5	1	3	3	33.33	25	**29.17**
TOTAL	57	39	119	47.90	59.37	**53.64**

TABLE 2.XII. ACCURACIES IN PERCENT ACHIEVED ON THE TEST IMAGES USING NORMALIZED CROSS-CORRELATION AND SVM CLASSIFIER WITH SCREENING.

	TP (true positive)	FP (false positive)	M (cars present)	Producer's Accuracy	User's Accuracy	Total Accuracy
Image 1	35	5	51	68.63	87.5	**78.06**
Image 2	20	1	31	64.52	95.2	**79.88**
Image 3	16	1	19	84.21	94.12	**89.16**
Image 4	14	4	15	93.33	77.78	**85.56**
Image 5	2	0	3	66.67	100	**83.33**
TOTAL	87	11	119	73.11	88.76	**80.94**

The screening step allows improving considerably the performance, passing from 58.52% to 80.94% if we consider the situation in which the SVM classifier is used. In this case, the number of false alarms is really limited: just 11 false alarms for all the 5 test images, while if the screening stage were not applied the number of false alarms would be 111. For the other two strategies, the use of a screening operation allows obtaining less marked improvements: the accuracies increase of 12-15%. By looking at the final results, one can notice that most of false alarms derive from dormers or from road signs which present rectangular shapes similar to those of cars. In urban areas, where it is likely to find objects with shape similar to that of cars, a screening step which reduces the areas to analyze and makes the detection procedure more robust is thus highly recommended.

By exploiting the shape of objects, we obtained interesting advantages and it allows us to overcome the problems of variability in color, texture and aspect that are typical for the car class. With a signature of length equal to 58 elements (*i.e.*, the number of templates in the catalogue), we were able to correctly identify 87 over the 119 cars present in the scenario achieving a final accuracy of 80.94% (see Figure 2.14). This means that, even if we worked with a limited catalogue, it is possible to identify most of the cars and at the same time keep the computational time contained.

Still analyzing the best results, one can observe that missed alarms correspond to cars hidden by shadows or by trees because these obstacles do not allow the extraction of the full shape producing low values of similarity. The cars inside outdoor parking lots or along roads, where there are no problems of shadows or trees, are almost all detected.

The comparison between the two strategies which exploit normalized cross-correlation and mutual information, respectively, is in favor of the first one which yields a difference of accuracy higher than 3%.

With correlation as similarity measure, it is possible to detect 12 cars more than what detected with the mutual information measure.

TABLE 2.XIII. POSITION AND ORIENTATION ERRORS ACHIEVED ON THE TEST IMAGES USING THE THREE STRATEGIES.

Strategy	Average Position Error [cm]	Average Orientation Error [°]
Normalized Cross-Correlation	39.5	6.2
Mutual Information	67	16.8
SVM-Correlation	30.8	8.8

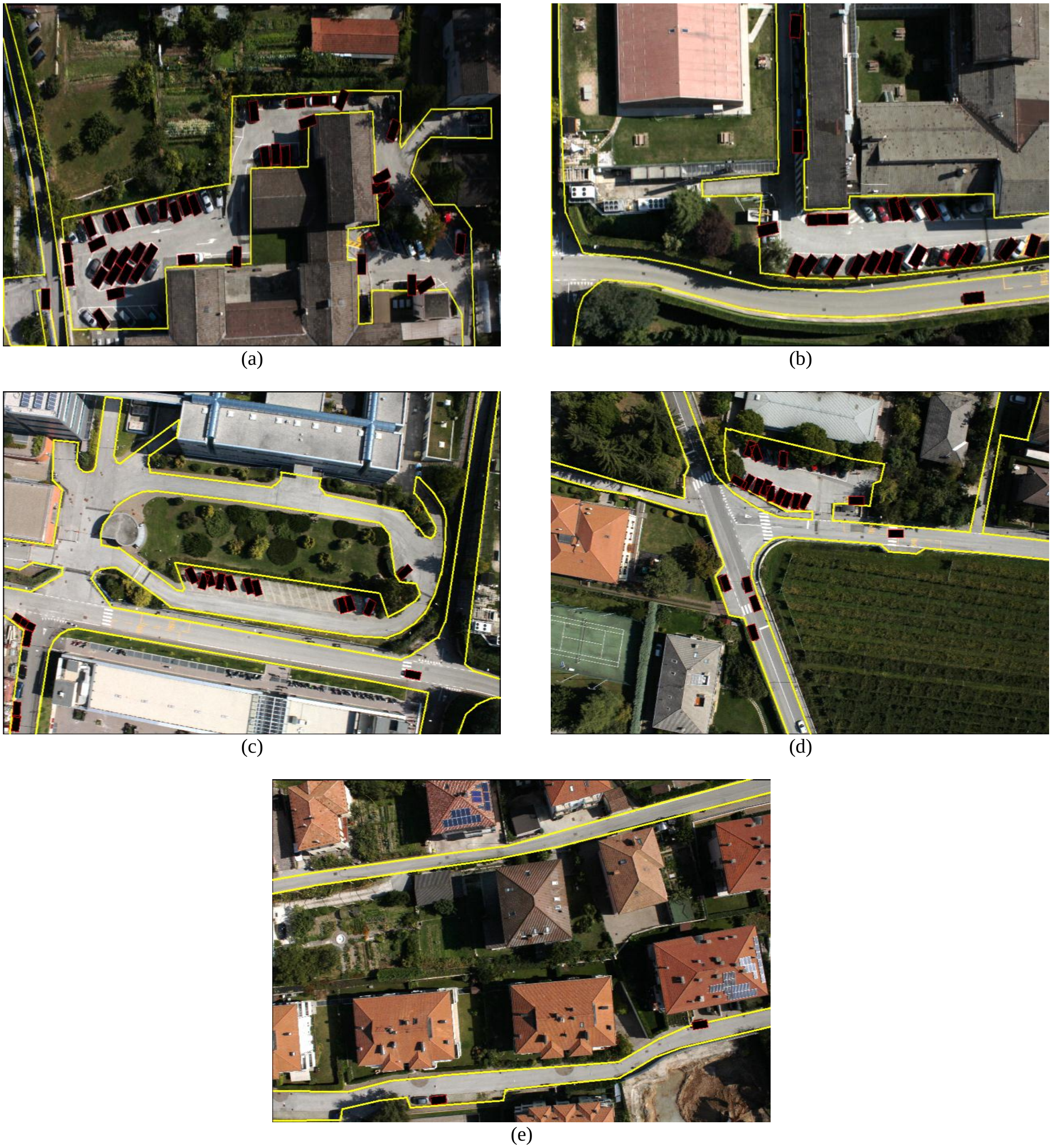
In addition to the detection goal, in this work we aimed at estimating the correct spatial position and orientation of each car. From Table 2.XIII, we can notice how the accuracies of the detection process for all the strategies are good. In particular, thanks to the strategy which exploits correlation and SVM classifier, we can obtain interesting results: the average position error by considering 87 true positives is of 30.8 [cm]. The strategy which exploits only the normalized cross-correlation measure produces a position error very close to that produced by the combination of normalized cross-correlation and SVM, namely 39.5 [cm]. By contrast, the strategy based on the mutual information yields a higher error (67 [cm]).

For what concerns the average orientation error, the best results are obtained by using only the correlation measure. The average error is just 6.2° against 8.8° for the combination of correlation and SVM classifier. It is noteworthy that these values are smaller than the step-size (10°) used for the orientation estimation. The error achieved with the combination of correlation and SVM classifier is higher than that obtained by using only the correlation measure because more cars, in particular some of those hidden by occlusions, are detected. Since some of these cars are partially occluded, the orientation estimation for these vehicles is not accurate producing a higher average error which is anyway very satisfactory. The error obtained by using the mutual information measure is higher (16.8°) but can be considered acceptable.

To provide a further element to compare the different strategies we reported the computational times (see Table 2.XIV) required for each similarity measure divided into: extraction time, matching time and time for the determination of the orientation for one car window and moreover, we pointed out the average time for image.

TABLE2.XIV. COMPARISON OF THE COMPUTATIONAL TIMES OF THE THREE STRATEGIES.

	Normalized Cross-Correlation	Mutual Information	SVM-Correlation
Extraction HoG Features	2.3 ms	2.3 ms	2.3 ms
Matching	56.6 ms	61.54 ms	56.6 ms
Orientation Determination	3.45 s	3.52 s	3.45 s
Average Time for Image	3.926 h	4.387 h	3.148 h


(a)

(b)

(c)

(d)

(e)

Figure 2.14. Final results obtained on the test images. (a) first image, (b) second image, (c) third image, (d) fourth image, (e) fifth image. The yellow lines delimit the screened areas.

2.6.4 Comparison of the Proposed Car Detection and Counting Methods

In order to assess further the goodness of the present methodologies, we compared them. In Tables 2.XV-2.XVI, the comparisons of the results obtained without and with the GIS-Based screening step, respectively, are reported for both car detection approaches. It is possible to observe that the detection capabilities are similar. Indeed, the first strategy detects 88 cars versus 87 for the second strategy. The difference, that influences the final accuracies, regards the number of false alarms. Without screening, we can notice that the technique which exploits the catalogue and similarity measures is able to reduce the number of false alarms of 32. With screening, the number of false alarms is of 12 samples lower. These differences on the number of false alarms allow increasing the final accuracy in the two situations of 2.5% and 4%, respectively. Detecting a smaller number of false alarms could be interesting especially if one aims at making the method completely independent from the screening steps that could not be applicable in some situations.

TABLE 2.XV. COMPARISON OF THE DETECTION PERFORMANCES OBTAINED BY THE TWO PROPOSED METHODOLOGIES WITHOUT THE SCREENING OPERATION.

		First Strategy			**Second Strategy**		
	M (cars present)	**TP (true positive)**	**FP (false positive)**	**Total Accuracy**	**TP (true positive)**	**FP (false positive)**	**Total Accuracy**
Image Test 1	51	43	32	**70.82**	35	10	**73.20**
Image Test 2	31	19	20	**57.03**	20	15	**60.08**
Image Test 3	19	16	60	**52.63**	16	30	**59.50**
Image Test 4	15	9	10	**47.37**	14	17	**69.24**
Image Test 5	3	1	21	**8.94**	2	39	**35.77**
TOTAL	119	88	143	**56.02**	87	111	**58.52**

TABLE2.XVI. COMPARISON OF THE DETECTION PERFORMANCES OBTAINED BY THE TWO PROPOSED METHODOLOGIES WITH THE SCREENING OPERATION.

		First Strategy			**Second Strategy**		
	M (cars present)	**TP (true positive)**	**FP (false positive)**	**Total Accuracy**	**TP (true positive)**	**FP (false positive)**	**Total Accuracy**
Image Test 1	51	43	12	**81.25**	35	5	**78.06**
Image Test 2	31	19	3	**73.83**	20	1	**79.88**
Image Test 3	19	16	3	**84.21**	16	1	**89.16**
Image Test 4	15	9	5	**62.14**	14	4	**85.56**
Image Test 5	3	1	0	**66.67**	2	0	**83.33**
TOTAL	119	88	23	**76.61**	87	11	**80.94**

From a qualitative viewpoint, the first strategy shows better results only in the first image, while the performances in the other 4 images are lower. The main reason is because the first image is the one with the highest number of cars covered by trees or shadows and as we said before the second strategy suffers from this kind of obstacles while the first one is more effective when some occlusions are present. However, in the other four cases where the number of occlusions is limited, second methodology shows better results and moreover gives information about exact position and orientation of the vehicles, which are not provided by the first one.

2.7. Proposed Vehicle Speed Estimation Approach

UAVs have the capacity to acquire images over the same area with a temporal distance of a few seconds or even less depending on the sensor. Usually, two images acquired with a small time lag are very similar and the only differences are represented by the moving objects. By acquiring images over roads or parking lots, it is possible to easily assume that the only changes between two subsequent images are represented by moving vehicles. This consideration is the central pillar of the approach devoted to the vehicle speed estimation, which is organized in three main steps (Figure 2.15).

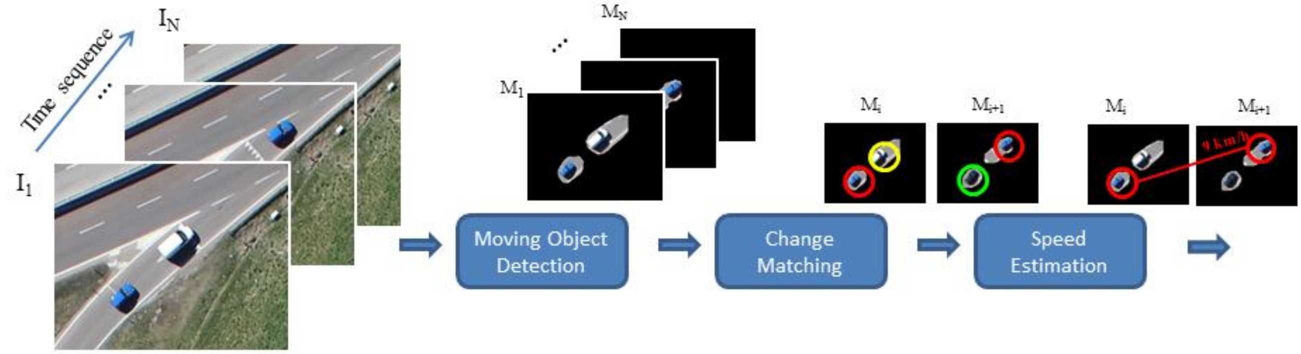


Figure 2.15. Flow chart of the proposed car speed estimation technique.

Let us consider two images I_i and I_{i+1} close in time in the sequence of *N* images acquired by means of a UAV. To detect the possible changes, the two images need to be compared. Despite the huge potential of UAVs, images generated with these systems are not perfectly aligned even if acquired from the same point and with a very small time difference. This problem, usually caused by atmospheric interferences and by small movements of the platform, can be solved thanks to a preprocessing step based on a registration process. The two images are aligned by applying a geometric transformation to I_{i+1}. By using the Scale Invariant Features Transform (SIFT) algorithm [25], two sets of invariant points used to derive the geometric transformation are extracted. The algorithm proposed by Lowe collects the maxima and the minima of a scale space of images generated by the convolution of the original image with a cascade of Gaussian functions. From all the extracted points, those which are poorly localized along the edges or which present a low contrast are discarded. Each of the remaining points is described with a feature vector. The vectors represent the local proprieties of the image in the surrounding of each point, indeed the features which compose the vectors are extracted from a 16×16 regions around each points. From these regions, 128 values which represent the Gaussian orientations are obtained. Once the two sets of points are created, to retrieve the geometric transformation it is necessary to detect the similar points between the two images, thus a matching step is performed. The matching process, as suggested by Lowe [25], is based on the computation of the Euclidean distances between all couples of points. Those points with the smallest distance are considered as matching points. Since the two images are very similar because acquired with a small time difference, we decided to improve the robustness of the matching by using a higher threshold value to determine the possible matching with respect to that proposed in the Lowe's algorithm. After the removal of the outlier points, the elements of the perspective transformation matrix T used for the geometric transformation are derived:

$$T = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ m_{31} & m_{32} & 1 \end{bmatrix} \tag{2.33}$$

where *t11, t21, t12* and *t22* are the scaling and rotation factors, *t13* and *t23* stand for translating factors, and *m31* and *m32* are transformation factors [59]. All the coordinates of the matching points compose an equation system, which is then solved in the least square sense to find an estimation of the parameters of *T*.

At this point, the two images are aligned, therefore the potential changes are identified by subtracting I_{i+1} to I_i. Considering that the two images are aligned, the subtraction of two pixels that represent a "no change" generates a value close to 0. Instead, the subtraction of two pixels representing a "change" produces a value larger than 0. By assuming the presence of just two classes of pixels (*i.e.*, "change" and "no change"), the possible changes are isolated by exploiting a simple thresholding process. The latter uses the Otsu's method [60] to identify the threshold values that is used to convert the image into a binary mask $\mathrm{I_m}$ in which the pixels with value equal to "one" represent the changes. The Otsu's algorithm, which is based on the assumption of a bimodal histogram, estimates the value of threshold for which the combined intra-class variance is minimal.

Usually the registration process is not able to produce a perfect alignment of the images making the subtraction step not precise. In particular, it is possible that the edges of the objects are not well aligned making the binary mask I_m affected by salt-and-pepper noise or by long-limbed changes typical of the misalignment of objects such as buildings or roads. To face this problem, the mask I_m is refined with a mathematical morphology (MM) procedure (see Figure 2.16). The key idea of MM is to process the considered binary image with basic elements called structuring elements (SE). $I_m(x, y)$, where *(x,y)* represent the spatial coordinates, is explored with the SE and at each spatial position the relationship between the elements and the mask is investigated. To remove noise, an opening operation which consists of an erosion operation followed by a dilation operation is applied. The opening operation is defined as follows:

$$I_m \circ SE = (I_m \ominus SE) \oplus SE \tag{2.34}$$

where $\ominus$ and $\oplus$ stand for the erosion and dilation operation, respectively. In this work, a small SE has been used for the erosion operation useful for the removal of the misalignment noise, and a bigger SE for the dilation operation to close possible holes.

Thanks to I_m, the areas which refer to possible changes are isolated in both images. The next step consists in the matching of each change in I_i with its respective in I_{i+1}. The objective of this part of the proposed technique is to associate to each change two couples of coordinates, one for each image, which will be used in the following for the computation of the speed. For this aim, the two sets of invariant points and the matching algorithm are again taken into consideration. In this case, instead of using all the points that belong to the two sets, only the points that potentially identify a change, namely those covered by the mask, are considered. In particular, a vector $V_{matchings}$, which contains the spatial image coordinates of the matching points, is created.

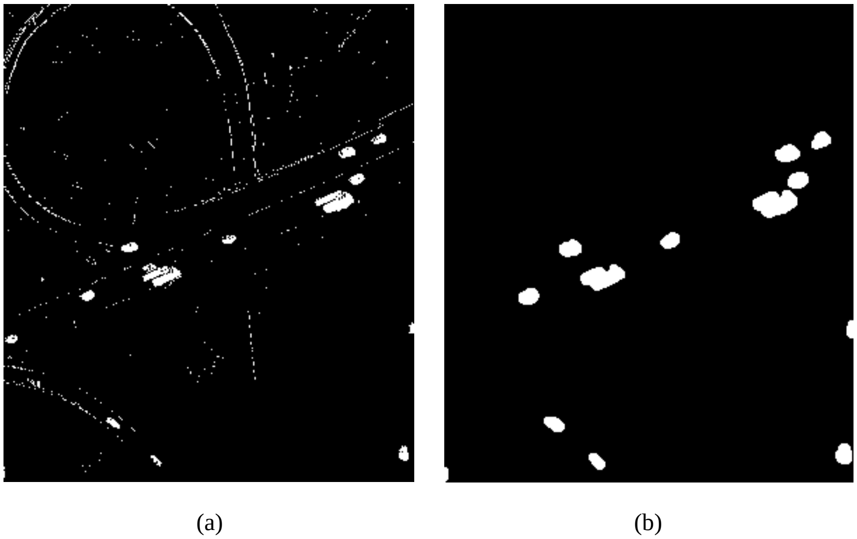


Figure 2.16. Image (a) obtained after the registration step. The misalignments of the street in the left corner are evident. Image (b) obtained after the MM procedure.

The two considered images are acquired with a time difference of a few seconds, therefore they are supposed to be very similar to each other. In this situation, it is likely to detect a large number of matching points for the same vehicle since the processing is performed on extremely high resolution images. As said before, the goal is to find the image coordinates of each change in I_i and I_{i+1}. Therefore to produce an univocal identification of the moving objects and to remove unfeasible matching (*i.e.*, couple of points with spatial coordinates too distant or too close), the vector $V_{matchings}$, which contains the coordinates of the matching points, is processed with an iterative algorithm for grouping the points spatially close in both images. At each step of the algorithm, the two spatially closest points p_k and p_j in I_i are considered. If their respective matching points, p_{k+1} and p_{j+1} in I_{i+1} are also closer than a threshold *Th*, they are merged together generating a new couple of matching points $\mathrm{p_z}$ and $\mathrm{p_{z+1}}$ with spatial coordinates in $\mathrm{I_i}$ and $\mathrm{I_{i+1}}$ equal to the average of the spatial coordinates, namely $(p_k + p_j)/2$ and $(p_{k+1} + p_{j+1})/2$, respectively (see Figure 2.17). The process continues until the Euclidean distance between the two closest points in $\mathrm{I_i}$ is higher than Th. In the proposed work, *Th* is empirically determined by considering the standard dimension of a car (i.e., standard length and width of a vehicle equal to 4.5 [m] and 1.8 [m], respectively) and the resolution of the acquired images.

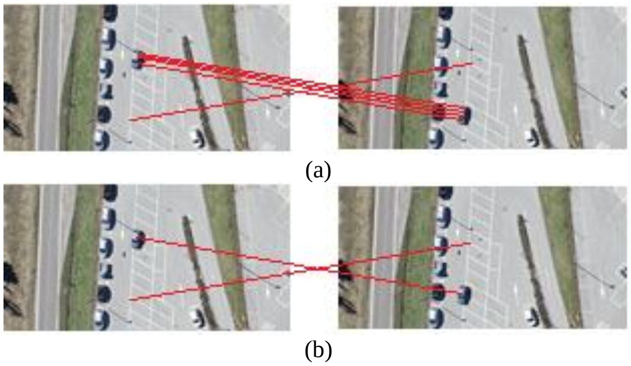


Figure 2.17. Example of how the merging step works. (a) before and (b) after he merging step.

The matching step combined with the merging process allows us to identify the spatial image coordinates for each possible moving object. The spatial distance covered by each moving object can be simply computed as follows:

$$d_n = \sqrt{\left(x_{n,i+1} - x_{n,i}\right)^2 + \left(y_{n,i+1} - y_{n,i}\right)^2} \tag{2.35}$$

where $x_{n,i}$, $y_{n,i}$, $x_{n,i+1}$ and $y_{n,i+1}$ represent the spatial coordinates of vehicle n in I_i and I_{i+1}, respectively.

Now, by considering that the two images are acquired with a small time difference, it is possible to assume that the changes of position of all the moving objects are very small and they are carried out on a flat plane. This assumption allows us to simply determine the velocities as follows:

$$v_n = \frac{d_n \times R_{pix}}{T_{i,i+1}} \tag{2.36}$$

where R_{pix} is the spatial resolution of the acquired images and $T_{i,i+1}$ is the time difference between I_i and I_{i+1}.

2.8. Experimental Results

In this section, we expose the results of the various simulations conducted on real sequences of UAV images acquired over roads and parking lots.

2.8.1. Dataset Description and Experimental Setup

The sequences of images have been acquired over two motorways near the city of Trento, Italy. In the first case, the sequence is composed by 31 images. All images, acquired from an altitude of 200 meters, are characterized by a spatial resolution of 5 centimeters. This first sequence allows us to monitor the motorway for a time interval of 50.9 seconds. The second acquisition is composed by 28 images acquired from 300 meters and consequently characterized by a spatial resolution of 7.5 cm. In this second case, the time interval is 87.1 seconds. For the second sequence, the true speed of an accomplice car is known, thus giving us the opportunity to verify the accuracy of the proposed technique. To better analyze the final results, the moving objects have been divided in different classes depending on where they are moving. For the first test, it is possible to discriminate between vehicles that are moving on the motorway and vehicles moving slowly on the parking lot; while for the second test, to the two previous classes, the class of cars that are moving on the motorway link roads has been added.

2.8.2. Experiment 1

The results obtained on the first test area are reported in Tables 2.XVII-2.XVIII and in Figure 2.17. From these first results, it is possible to see that the proposed technique works well. From the 94 moving objects present in the 31 images, 80 are correctly identified and monitored. With moving objects, we mean an object that is correctly identified and matched in a couple of images. The number of false alarms is contained if one considers the number of images which has been analyzed. Of the 32 false alarms, 7 are caused by the double recognition of the same object. Long vehicles that exceed the standard dimension of a vehicle, at the end of the merging step could be identified by more than just one matching point. In the first simulation during the 50 seconds of monitoring, three long trucks moved along the motorway. Since the threshold of the merging step is calibrated on the standard dimension of a car, when we are in the presence of a long truck it is likely to identify two separate moving objects: the truck and the trailer. For this reason, these 7 false alarms cannot be considered as real false alarms. For what concerns the estimated speeds, they are very realistic. The vehicles which are moving on the parking lots have an average speed of 15.3 km/h while the average speed of the moving objects on the motorway is 78.5 km/h. By considering the scenario where the acquisition has been performed these average speeds seem to be truthful. Another confirmation that the estimated speeds are reliable is provided by the monitoring of the changes of the speed of one specific object. For instance, by looking at the cars in the red and in the green circles (see Figure 2.18), it is possible to appreciate how the speeds become higher since the cars are leaving a roundabout. Contrarily, the speed of the car in the blue circle is decreasing because it is approaching the roundabout. Similar consideration could be done also for the cars in the parking lot. The car in the light blue circle has just started moving when the acquisition began and this is confirmed by the monitoring of the speed that is low in the beginning and becomes higher with the moving of the car.

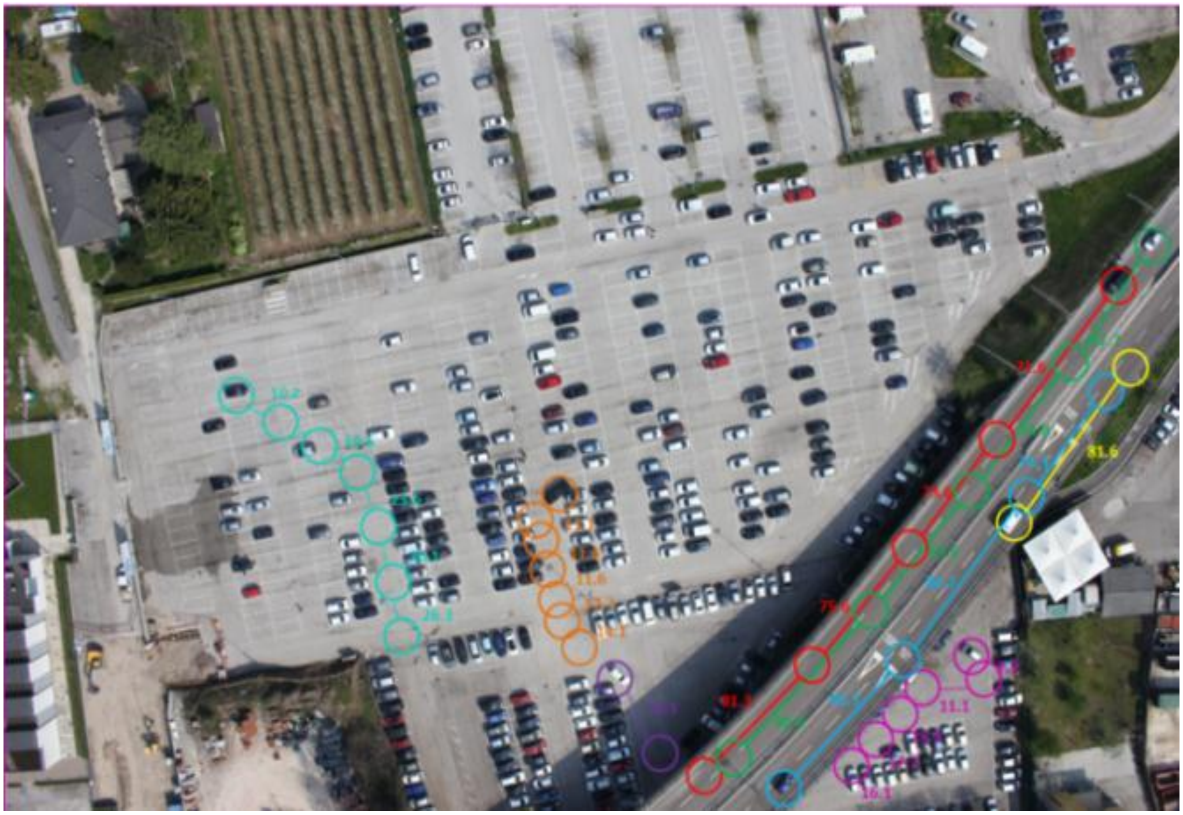


Figure 2.18. Monitoring of 10.8 seconds of the first test area. Different colors represent different moving objects. The circles of same color represent the temporal monitoring of a given vehicle. The numbers near the line joining two successive circles.

TABLE 2.XVII. DETECTION RESULTS ON THE FIRST TEST AREA

True Positives	False Alarms	Missed Alarms	Producer's Acc.	User's Accuracy
80	32	14	85.10%	71.42%

TABLE 2.XVIII. ANALYSIS OF THE SPEED IN KM/H OF CARS ON THE FIRST TEST AREA

	Parking Lot			Motorway	
Max Speed	Min. Speed	Average Speed	Max Speed	Min. Speed	Average Speed
30.2	5.9	15.3	93.3	55.9	78.5

2.8.3. Experiment 2

The results achieved on the second dataset, reported in Tables 2.XIX-2.XX and Figure 2.19, are a bit less satisfactory than those obtained on the first dataset because of the more complex scenario. Indeed, the monitored motorway and the underlying motorway link roads were more congested causing a higher number of false alarms. If the number of cars increase and the number of roads to monitor is higher, the risk to get more false alarms is higher. Another factor that influences the number of false alarms is the quality of the registration process. In a complex scenario, it is possible that some areas of the images (*i.e.*, the borders of roads and buildings) are wrongly registered and therefore they could be seen as moving objects. Consequently, the interest points are also extracted from these areas generating more false alarms. It is worth

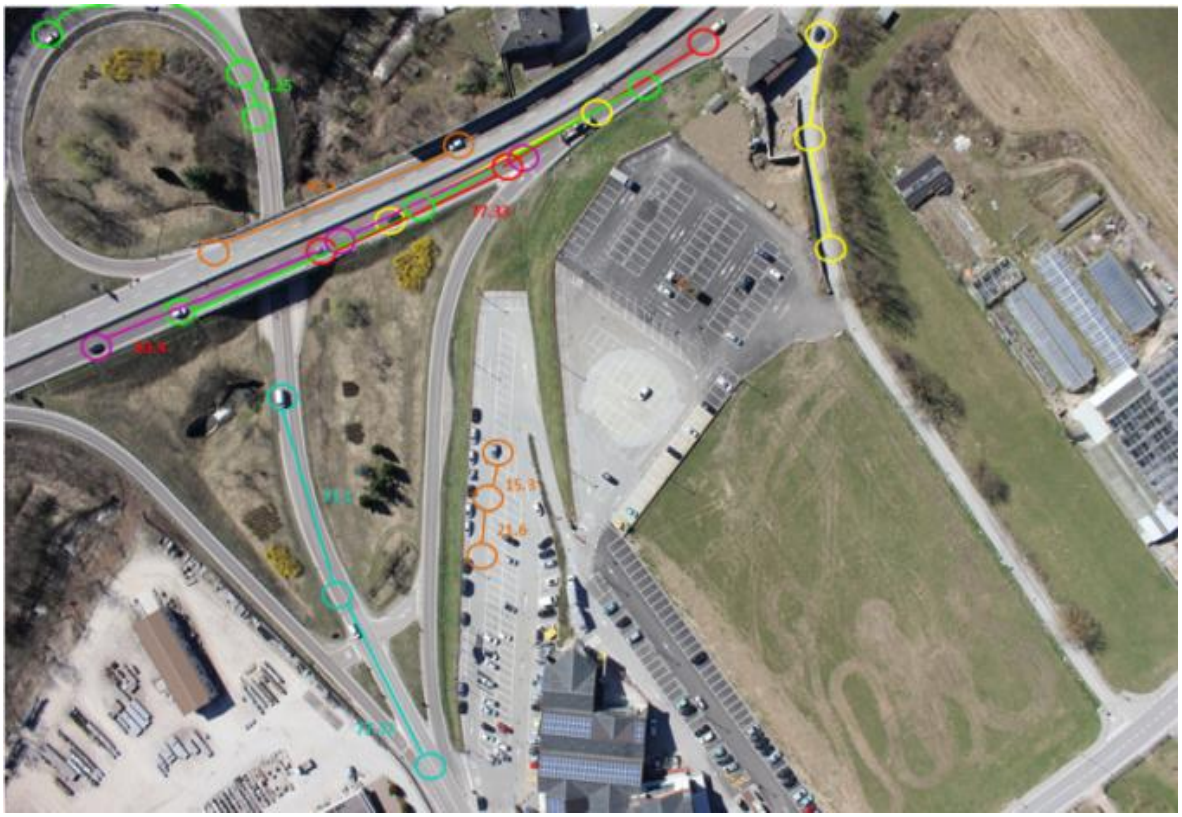


Figure 2.19. Monitoring of 9.3 seconds of the second test area. Different colors represent different moving objects. The circles of same color represent the temporal monitoring of a given vehicle. The numbers near the line joining two successive circles.

to highlight that this acquisition has been performed from an altitude of 100 meters higher with respect to that of the first experiment. If the resolution of the images is poorer, the objects are described with a lower level of detail and consequently the matching step becomes more ambiguous.

Despite the lower quality of the images, the final results are in general satisfactory. The number of moving objects correctly detected is 95. Instead the false alarms are 46 of which 6 are related to double recognitions. Actually, during the 80 seconds of monitoring, two trucks have passed through the motorway and motorway link roads. Also in this second test, the analysis of the speed of the vehicles on the different areas suggests that they are reliable. The average speed on the parking lot is 16.6 km/h, very similar to that detected in the first dataset. Instead the average speed on the motorway is higher with respect to the average speed on the other motorway. In the previous experiment, the vehicles on the motorway are entering or are moving away from a roundabout. Therefore, the speeds cannot be too high. Here, there are no intersections on the motorway, thus the vehicles can keep higher speeds. The average speed on the motorway link roads is compatible with the scenario.

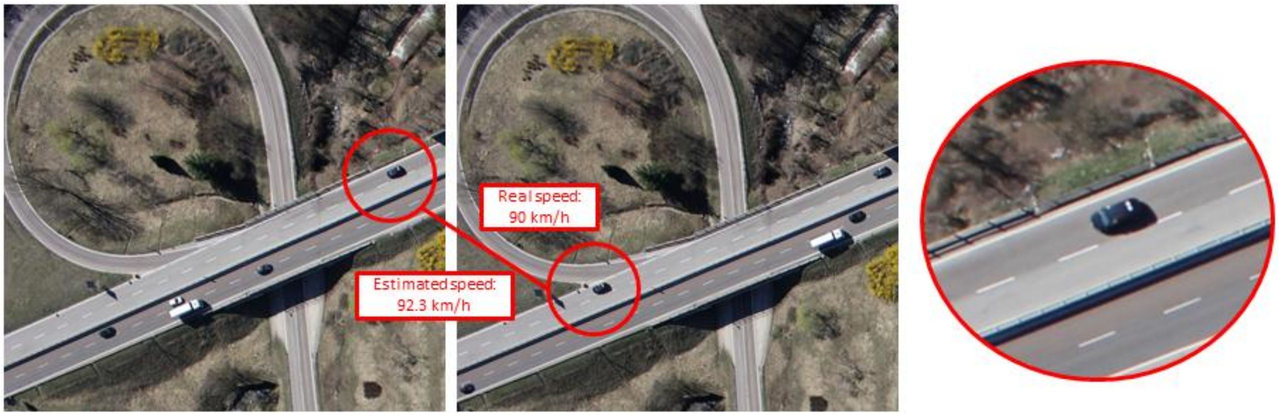


Figure 2.20. Monitoring of the accomplice car.

In this second acquisition, the true speed of an accomplice car was a priori known (Figure 2.20) and it has been used as “ground truth” to evaluate the quality of the proposed technique. The accomplice car moved along the motorway with a constant speed of 90 km/h (computed with the onboard odometer). The algorithm has correctly detected the car and estimated its velocity to 92.3 km/h. This final result confirms the high level of accuracy that the algorithm is able to provide.

TABLE 2.XIX. DETECTION RESULTS ON THE SECOND TEST AREA

True Positives	False Alarms	Missed Alarms	Producer’s Accuracy.	User’s Accuracy.
95	46	52	67.37%	64.62%

TABLE 2.XX. ANALYSIS OF THE SPEED IN KM/H OF VEHICLES

Parking Lot			Motorway			Motorway Link Road		
Max Speed	Min. Speed	Average Speed	Max Speed	Min. Speed	Average Speed	Max Speed	Min. Speed	Average Speed
31.7	5.3	77.3	14.3	47.3	92.5	77.3	14.3	47.3

2.9. Conclusions

In the first part of the chapter, two new methods for the automatic detection and counting of cars in UAV images are developed, whereas in the second part of the chapter, a strategy devoted to the estimation vehicles speed form sequences of images is proposed.

The first car detection methodology starts with a screening step in which through the use of a supervised classifier we detect the regions covered by asphalt assuming that usually cars in an urban scenario lie over asphalted regions (*e.g.*, roads and parking lots). This procedure permits us to reduce the areas of investigation making the algorithm faster and with fewer false alarms. Moreover, the screening has been performed exploiting GIS information instead of using the automatic classifier. The second step is focalized on the extraction of a set of points that are invariant to affine transformations. All these points are characterized by a vector which represents some proprieties of the surrounding area around each point. In order to give more information to each keypoint, we added other spectral and morphological features in the keypoint descriptor. Afterwards, a classifier properly trained allows us to discriminate the points corresponding to the car class. In the last part of the procedure, an algorithm is implemented to merge the car keypoints belonging to the same car. This step is necessary because, at the end of the keypoint classification, a car is typically identified by more than one keypoint. We showed results considering the three typologies of screening. This analysis was important to assess the impact of the screening step and to understand where it is possible to find potential improvements. Analyzing the obtained accuracies which are in general encouraging, we can conclude that the screening step is fundamental especially to reduce the number of false alarms. The effective number of detected cars is more or less constant in the three situations, so it makes us thinking that the merging step is correct and it works well independently form the screening step. This is confirmed by the three producer's accuracies that are all over 65%.

The second proposed method is organized in four main steps. As the previous method, the first step consists in the screening of areas over which it is possible to find cars. In the context of this second strategy, the screening procedure exploits only GIS information from which one knows a priori where the asphalted areas are. Then, a double filtering is applied on the images in order to assign to each position of a sliding window two sets of HoG features: one for the horizontal and one for the vertical filtering directions. The successive discrimination between car and background points is performed by means of one of the three proposed strategies. The first two are based on the normalized cross-correlation and mutual information measures, respectively. The third one is a combination of the correlation measure and SVM classification. In the third step, to the points classified as car points is associated an orientation value computed by searching for the highest value of similarity measure in 36 possible directions. As before, the last step of the approach is dedicated to the merging of the points which belong to the same car because, due to the extremely high resolution of the images, at the end of the discrimination step it is likely that a car is identified by more than one point. For what concern the second car detection approach the results are reported without and with the ideal screening operation and include the three proposed similarity estimation strategies. As for the first approach, it is easy to understand the importance of the screening step to reduce the number of false alarms. Among the three similarity measures, it comes out that the best results are those obtained thanks to the use of the SVM classifier which allows identifying 87 cars over the 119 present in the images, with a position error of 30.8 [cm]. Another consideration that is possible to draw is about the error of the orientation estimation which is below 10°. By comparing the two proposed method it is possible to observe that the number of detected cars is similar while the number of false alarms is lower in the second case.

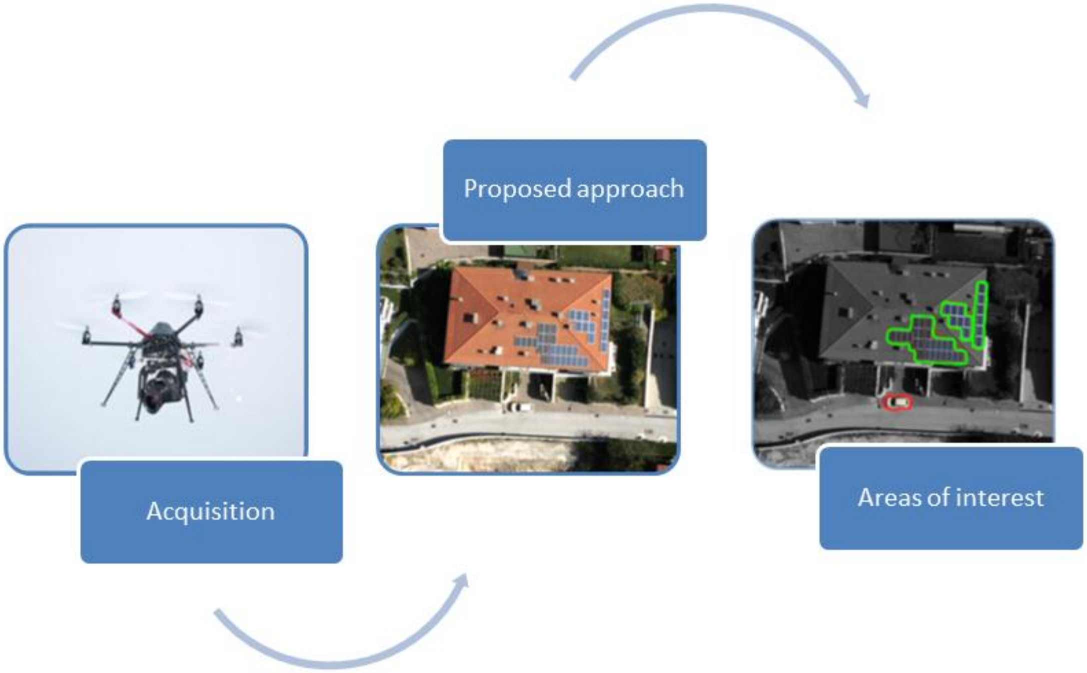
In the second part of the chapter, we developed a three-stage procedure for the automatic detection and speed estimation of vehicles present in a sequence of images acquired by means of an UAV. We demonstrated how the presented approach could be used as an instrument to monitor the speed of vehicles, through which local municipalities can take appropriate decisions on the management of the urban traffic. The images belonging to the sequence are considered in pairs and the tracking of the objects in time allows the estimation of the speeds. The proposed technique starts with the determination of the appropriate geometric transformation to apply to the second image (of the pair) in order to make it perfectly aligned with the first one. The registration step is followed by a refinement process after which the moving objects are isolated. By exploiting the invariant points previously extracted and used for the registration step, each moving object in the first image is matched with its respective in the second image. Also in this case, since UAV acquire extremely high resolution images, the possible matching detected for each moving object could be more than one, a merging step has been implemented. At the end, thanks to the knowledge of the positions of the objects in both images and of the spatial resolution, the speed of all the moving objects is inferred.

The final results are in general promising. Especially in the first experiment where the scenario was less complex, the detection of the moving objects is higher than the 85 % and the number of false alarms contained. In the second dataset, the scenario is more problematic but the final detection results are still good. The estimated speeds are realistic and suitable to the contexts. The average speeds in the different areas are truthful. The accuracy of the achieved results is confirmed by the knowledge of the speed of an accomplice car that was moving along the motorway during one acquisition.

The proposed methods allow obtaining good results, but additional developments could be envisioned to make the methods more accurate. Possible improvements regarding the detection and counting of cars could be achieved by using a multiresolution analysis approach to face the problem of different resolutions that could affect images acquired by UAV systems or by spending efforts by assessing other kinds of descriptors and detectors (e.g., Harris and Gabor filters, local binary patterns) as potentially alternatives to SIFT in terms of complexity and discrimination power. For both techniques, the car keypoint merging has been performed by means of a simple solution based on spatial clustering. This step of the proposed procedure could be potentially improved by adopting more sophisticated solutions involving for instance spatio-spectral clustering. By analyzing the vehicles speed estimation approach, additional developments could be envisioned for the registration and the matching processes. The improvements of the registration process could allow considering images with a larger temporal difference, while the enhancements of the matching step is important for dealing with more complex scenarios

3. Filter-Based Object Detector for UAV Images

***Abstract** – Unmanned Aerial Vehicles (UAV) acquire images characterized by an exceptional level of detail which calls for processing and analysis methods capable to efficiently exploit their rich information content. In particular, the detection of specific classes of objects (e.g., cars, roofs) represents an important but challenging task for these images. Most of the related literature has aimed at proposing methods capable to provide satisfactory detection accuracies. However, they typically refer to a specific class of objects and give little attention to the processing time. In this chapter, we present a novel and fast methodological alternative. In addition of being particularly fast, the proposed method is a general detection approach which can be customized to any class of objects after an opportune training phase. It consists in the design of a nonlinear filter which combines image gradient features at different orders and Gaussian process (GP) modeling. High-order image gradients permit to capture detailed information regarding the structure of the investigated class of objects. The GP model fed with high-order gradients yields an estimate of the presence of the object of interest for a given position of the sliding window within the image. Two separate sets of experiments were conducted, each aiming at assessing the proposed method to detect a given class of objects common in urban scenarios, namely vehicles and solar panels, respectively. Results on real UAV georeferenced images characterized by 2-centimeter resolution and by three channels (RGB) are provided and discussed, showing particularly interesting properties of the detector*.



3.1 Introduction

The detection of predefined objects represents one of the major interests in the remote sensing technologies. Several applications such as environmental monitoring, surveillance and image retrieval are strongly related to the detection of particular classes of objects. Thanks to the continuous improvements in terms of spectral and spatial resolutions of remotely sensed images, the number of studies around the object detection issue has grown significantly. For instance, in order to estimate the urban extension, Duriex et al. [61] introduce a building extraction methodology for urban sprawl monitoring. Corbane et al. [62] present a system for the automatic ship detection from optical satellite imagery to monitor the maritime traffic. The problem of target detection in hyperspectral imagery has been widely studied [63]- [64]. In this context, the techniques exploit the rich spectral information to detect the presence of the investigated targets.

As long as the spatial resolution of the acquired images is of the same order of the dimension of the investigated objects, pixel-based approaches or their variants could be suitable. However, with the advent of very high resolution (VHR) images, object-based analysis approaches are potentially more adapted for individuating specific objects in an image under investigation. Object-based approaches typically rely on the idea of exploiting the spectro-geometric characteristics of objects such as shape, appearance and dimensions. For instance, in [65], the authors propose a Bag-of-Visual words (BOW) representation for the classification of VHR images. Other studies perform a preliminary segmentation of the image to simplify the detection of the objects. For instance, in [66] the authors present an unsupervised technique for the detection and segmentation of orchards. Some other works exploit local descriptors such as Scale Invariant Feature Transform (SIFT) [25] or Speeded Up Robust Features (SURF) [27] to identify the targets. In [67], the authors introduce an interesting method based on SIFT keypoints and graph theory for the detection of buildings in VHR satellite images. Lately, Felzenszwalb et al. [68] describe an object detection model that aims at representing the classes of objects by a collection of multiscale deformable parts model. This model has demonstrated to be extremely effective for ground-shot images, but presents some limitations on remote sensing images. Objects in ground-shot images usually assume local appearance proprieties combined with spatial relationships that are unambiguous and allow obtaining efficient and accurate results. These characteristics are difficult to meet in remote sensing images and for this reason a technique which considers a collection of parts model is not appropriate.

Thanks to the advent of Unmanned Aerial Vehicles (UAV), the quality of the acquired data has raised exponentially and the range of possible applications has become wider. UAVs are aerial platforms that can be quickly used in time of need and over the desired areas of interest. Their deployment is fast and since they can fly in a completely autonomous way, without pilot onboard, these platforms are proving to be ideal to collect extremely high resolution images in critical situations or after natural calamity. In [69], a technique for fire detection in visual and infrared images is proposed while in [70] the authors examine techniques to identify building damages from images acquired by means of a UAV. The possibility to equip these platforms with different sensors (i.e., optical or thermal sensors) combined with the capability of UAV to collect data over the same area at different times with a high temporal resolution has allowed to develop novel techniques also in the precision farming field [6]. Recently, the development of techniques for the automatic identification of specific targets or classes of objects in urban areas with UAV is one of the applications on which part of the scientific community has focused its attention. In particular, the detection of people or vehicles could permit the local municipalities to take quick and appropriate decisions regarding the management of crowed places or heavily congested routes. For instance, Gleason et al. [38] propose a method for the automatic detection of vehicles from aerial images by comparing different feature extraction methods. In [71], the authors present an efficient method to detect and count cars in urban scenario by exploiting similarity measures and a Support Vector Machine (SVM) classifier. In [72], the authors expose a method based on the combined use of SIFT features and SVM classifier to detect vehicles in UAV images. Thermal and visible images are used in [15] to perform the real time detection of people and vehicles form UAV imagery.

As can be seen above, most of the related literature has aimed at proposing methods capable to provide satisfactory detection accuracies. However, they typically refer to a specific class of objects and give little attention to the processing time. In this chapter, we present a novel and fast methodological alternative to detect any desired class of objects for extremely high resolution (EHR) images (few centimeters) acquired through UAV. In addition of being particularly fast, the proposed method is a general detection approach which can be customized to any class of objects after an opportune training phase. Features based on the traditional image gradient computation may not be sufficient to deal with the great geometric information content of EHR images. As a consequence, here, we propose to build features from high-order gradients of the image as a means to extract detailed information regarding the structure of the investigated class of objects. Relying on a linear model for handling all these information and for capturing the differences of aspect of objects belonging to the same class may result not effective. For this reason, we develop a nonlinear filter based on the Gaussian Process (GP) regression model [73]- [74], known for its capability to tune the free parameters of the model in an automatic way and for its short processing time. The proposed filter works thus in two phases. In a first phase, it undergoes a training operation to learn to detect a predefined class of objects. In the second one, it is applied on the desired images by computing high-order gradients in a given sliding window and assigning to each window position an estimate of the presence or absence of the predefined object. The proposed method was experimentally assessed to detect two different classes of objects, namely cars and solar panels, in urban areas. Results are provided and discussed, showing particularly interesting properties of the method.

This chapter is organized as follows. Section 3.2 describes the whole methodology in which the design of the filter is thoroughly explained. Section 3.3 is devoted to the description of the GP regression model while in Section 3.4 we describe the real datasets used for the experiments and provide and comment the experimental results. The final conclusions of the work as well as the future developments are presented in Section 3.5.

3.2 Process Description

3.2.1 Problem Formulation

Let us consider an EHR image $I(p,q)$ (where (p,q) stands for pixel coordinates in image I) acquired by a UAV. Usually, UAVs by flying at low altitude acquire images in which single objects are described by thousands of pixels and consequently with a high level of detail. This huge amount of information represents a considerable advantage in the description of the objects but, at the same time, it can be a problem because even objects belonging to the same class may look very different.

The objective of this work is to develop a novel filter capable to detect in image I a specific class of objects accurately and in a reasonable amount of time. As illustrated in Figure 3.1, first, for a given position of a sliding window, a set of gradient features at different orders is extracted. These features are suitable for our goal because they permit to emphasize the geometric characteristics of the objects and thus to potentially better capture the intra-class object variability. The extraction of high-order gradient features produce feature vectors of very large dimensions which may call for a feature reduction process. After the reduction step, an estimate of the presence of the investigated object is carried out by a GP regression model. The final decision of the presence or not of the object is performed through a simple thresholding task whose value is empirically determined. The key elements of the method mentioned in the previous brief synthesis are detailed in the following.

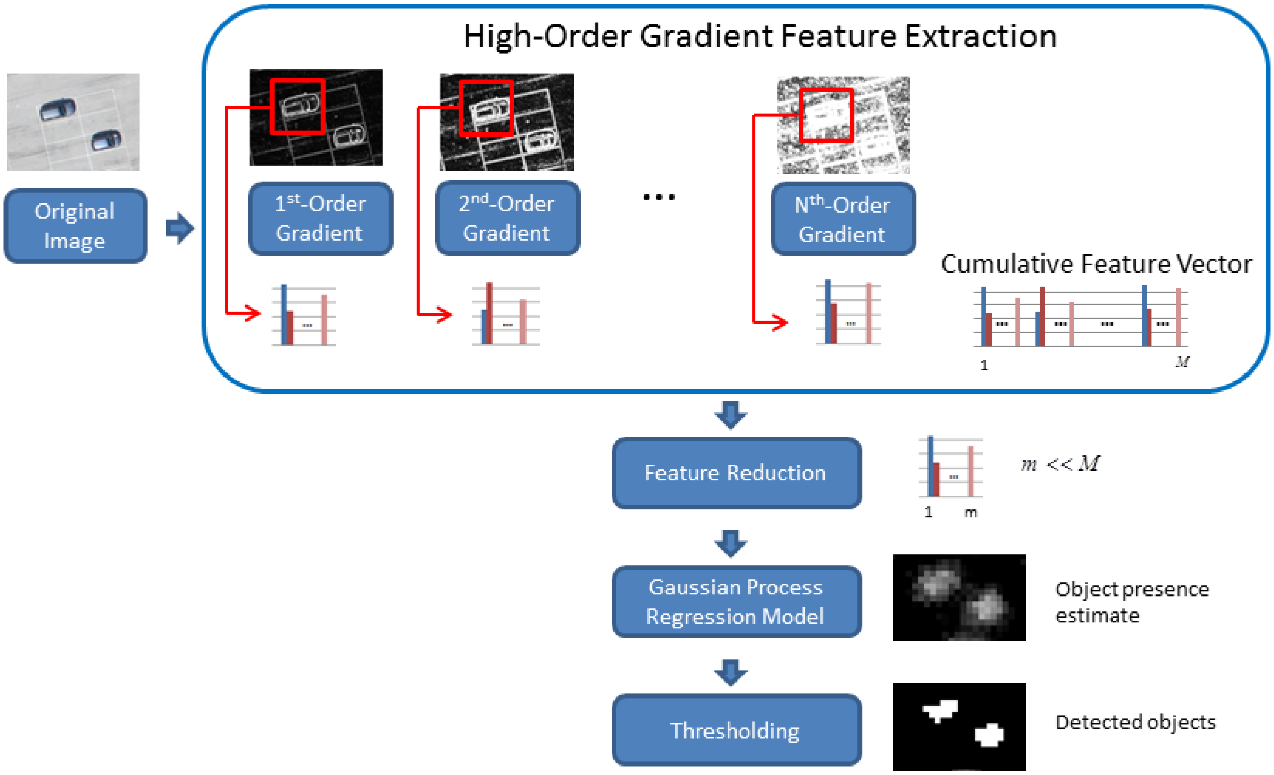


Figure 3.1. Flow chart of the proposed object detection method.

3.2.2 High-Order Gradient Features

The gradient image is widely recognized as one of the most efficient ways to highlight the edges and thus the shape of an object. For this reason, it has been selected for our goal. A gradient image is generated by convolving the original image with a filter. In this work, the Sobel filter will be used for its velocity and simplicity. Each pixel of a gradient image represents the intensity change of the same pixel in the original image in a given direction. The gradient of an image is computed as follows:

$$\nabla I = \frac{\partial I}{\partial p}\hat{p} + \frac{\partial I}{\partial q}\hat{q} \quad \text{where} \quad \begin{cases} \dfrac{\partial I}{\partial p} & \text{is the gradietn in the x direction} \\ \dfrac{\partial I}{\partial q} & \text{is the gradient in the y direction} \end{cases} \tag{3.1}$$

In the literature, one can find several works which exploit image gradient features to detect classes of objects. For instance, in [54] the authors introduce a method based on histogram of oriented gradients to detect people.

In EHR images, the objects are described with a so high level of details, that may render features based on the traditional image gradient computation not enough powerful to capture suitably the object structure. In this work, we propose to push further the gradient-based image analysis by building features from high-order gradients of the image as a means to extract more detailed information regarding the structure of the investigated class of objects (see Figure 3.2). The gradient image features of the *N*-th order are computed as follows:

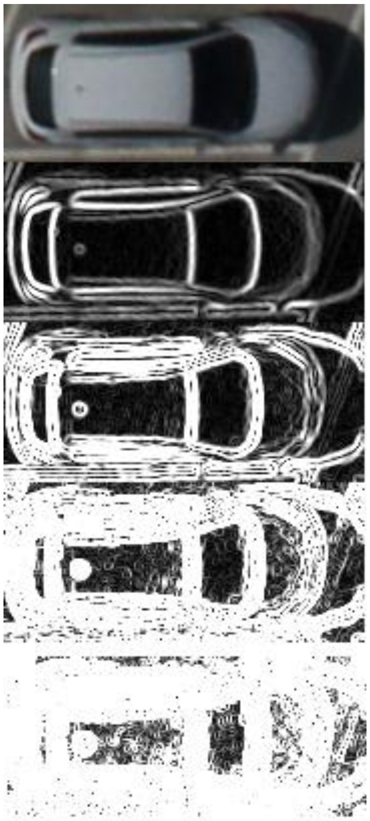


Figure3.2. Example of gradient features extracted at different orders (from 1 to 4).

$$\nabla I_N = \frac{\partial I_{N-1}}{\partial p}\hat{p} + \frac{\partial I_{N-1}}{\partial q}\hat{q} \quad \text{where } I_{N-1} \text{ is the image gradient at the } N-1 \text{ order} \tag{3.2}$$

The gradient image features are extracted by means of a sliding window process with window size equal to *dim_p* and *dim_q*, respectively. Each window is described by a feature vector of dimension $M = dim\_p \times dim\_q$, which includes the norm of the (vertical and horizontal) gradients at all considered orders and for each pixel of the window. The dimensions of the sliding window have to be adapted to the expected dimensions of the considered object and to the resolution of the acquired images. For instance, assuming that the objects of interest are cars and that a standard length and width of a car is equal to 4.5 [m] and 1.8 [m], respectively, and a sensor resolution of 2 [cm], a suitable window size could be around 200×200 pixels to include the object car in all possible orientations.

3.2.3. Filter Model

Filtering has always been one of the cornerstones in the image processing field. The theory behind 2D filtering techniques for image processing applications usually derives from the 1D signal processing theory. The nature of the filter often depends on the task that one would like to accomplish and on the characteristics of the analyzed data. Generally, to identify a specific target inside an image one may prefer to use a linear filter for its simplicity. For instance, the Sobel filter is a linear filter exploited to identify edges inside images. Let $\underline{x_i} \in \Re^M$ be a vector of M extracted image gradient features and y_i be the target value which represents the probability of presence of the considered object inside the analyzed window, the goal of the filter design is to find a function F such that:

$$y_i = F(\underline{x_i}) \tag{3.3}$$

If one considers a whole class of objects, the definition of a proper f is not trivial. Indeed, the relationship between the input space (*i.e.*, image gradient features) and the output space (*i.e.*, the probability of presence of the investigated object in the window) is expected not to be linear. Therefore, one needs to resort to a nonlinear filter. The design of such a filter could be done by exploiting methods based on parametric (e.g., polynomial) or nonparametric approaches. In absence of any prior knowledge about f, the nonparametric approach is preferred despite its higher computational complexity. Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs) are good nonparametric candidates [75]- [42].

In the last years, the use of Gaussian Process (GP) nonparametric regression models has gained popularity. GP regression, after an opportune training phase, provides probabilistic prediction estimates for the analyzed samples. GP regression models present some interesting characteristics that make them preferable over other methods. Firstly, they can be learned from small training sets. This property is important especially in remote sensing tasks where the number of training samples is typically not large. And secondly, it is possible to tune the free parameters of the model within a Bayesian framework in a completely automatic way instead of making use of a traditional empirical parameter estimation as typically done for ANNs and SVMs.

After the extraction of the features by means of the sliding window process, every patch is processed with the GP regression model in order to obtain a predictive estimate and to isolate the windows that potentially contain the sought object.

3.2.4. Window Size and Dimensionality Reduction

As pointed out previously, one of the main factors that lead to the determination of the sizes of the sliding window is the expected dimensions of the investigated objects. In EHR images, the objects are often represented by thousands of pixels, which may force us to use large sliding windows. The dimension of the sliding windows impacts directly on the number of generated features. Indeed, the dimension of the feature vectors which describe the windows is M. In case of large windows, the number of features M is extremely high, incurring in a risk of the curse of dimensionality. A common procedure to solve this potential issue is dimensionality reduction for passing from M to m $(m \ll M)$ to [76]. Usually, the reduction of the number of features under consideration could be done in two different ways, namely feature selection or feature extraction. The first approach consists in the identification of a subset of relevant features while the second approach transforms the original high-dimensional space into a subspace of smaller dimension. In this work, we will make use of two different and simple techniques based on the above two approaches, respectively. In particular, the first method is based on the selection of the features through a regular sampling of the window. The second one relies on the Principal Component Analysis (PCA). PCA is an orthogonal linear transformation which projects the data on the so-called principal components, ranked according to their variance (*i.e.,* the quantity of information they convey). The first principal component is the one with the largest possible variance. The first m principal components are chosen for performing the projection (and thus the feature reduction). The value of m in this case could be inferred by keeping the principal components which together account for a certain fraction of the total variance.

3.3. Gaussian Process Regression

3.3.1. Gaussian Processes

Let $G = \{(\underline{x_i}, y_i), i = 1,2, \dots, T\}$ be a set of training samples where each $\underline{x_i} \in \Re^m$ represents a vector of m (selected/extracted) image gradient features and $y_i \in \Re$ is the associated target value (*i.e.*, the percentage of presence of the considered object in the analyzed window). Let $\underline{\underline{X}}$ be the $m \times T$ feature matrix that conveys all the $\underline{x_i}'s (i = 1,2, \dots, T)$ and $\underline{y}$ be the $T \times 1$ matrix (vector) that includes the target values. Accordingly, $G = \{\underline{\underline{X}}, \underline{y}\}$. The aim of the process is to infer from the set of samples G the function $F(\cdot)$so that $y = F(\underline{x})$.

Under a Gaussian process learning framework, the observed $\underline{y}$ of the function to model are supposed to be the sum of a latent function $\underline{f}$ and a noise component $\underline{\varepsilon}$, where:

$$f \sim GP\{\underline{0}, \underline{\underline{K}}(\underline{\underline{X}}, \underline{\underline{X}})\} \tag{3.4}$$

$$\underline{\varepsilon} \sim N(\underline{0}, \sigma_n^2 \underline{\underline{I}}) \tag{3.5}$$

Equation (3.4) means that a Gaussian process $GP\{.,.\}$ is assumed over $\underline{f}$, i.e., this last is a collection of random variables, any finite number of which follows a joint Gaussian distribution with mean $\underline{0}$ and covariance matrix $\underline{\underline{K}}(\underline{\underline{X}}, \underline{\underline{X}})$. This matrix is built by means of a covariance (kernel) function $k(\underline{x}, \underline{x}')$

computed between all the training sample pairs. Equation (3.5) says that a Gaussian distribution $\mathrm{N}(.,.)$ with zero mean and variance σ_n^2 is supposed for the entries of the noise vector $\underline{\varepsilon}$ with each entry drawn independently from the others (I in this case represents the identity matrix). Because of the statistical independence between the latent function $\underline{f}$ and the noise component $\underline{\varepsilon}$, also the noisy observations $\underline{y}$ are modeled with a Gaussian process, *i.e.*:

$$\begin{gathered}\underline{y} \sim GP\left(\underline{0}, \underline{\underline{K}}\left(\underline{\underline{X}}, \underline{\underline{X}}\right) + \sigma_n^2 \underline{I}\right) \\ \Updownarrow \\ p\left(\underline{y} \,\middle|\, \underline{\underline{X}}\right) = N(\underline{0}, \underline{\underline{K}}\left(\underline{\underline{X}}, \underline{\underline{X}}\right) + \sigma_n^2 \underline{I})\end{gathered} \tag{3.6}$$

3.3.2. GP Regression

A way to derive a GP regression model consists in formulating the Bayesian estimation problem directly in the function space (the so-called function space view). According to this formulation, given the set of samples *G*, the best estimation of the output value f_* associated with an unknown sample $\underline{x}_*$ is represented by the expectation of the desired output quantity conditioned to G and $\underline{x}_*$, *i.e.:*

$$\hat{f}_* \mid \underline{\underline{X}}, \underline{\underline{y}}, \underline{x}_* \sim E\left\{f_* \middle| \underline{\underline{X}}, \underline{\underline{y}}, \underline{x}_*\right\} = \int f_* p(f_* \mid \underline{\underline{X}}, \underline{\underline{y}}, \underline{x}_*)\, df_* \tag{3.7}$$

In order to determine the output value estimate, it is necessary to compute the predictive distribution $p(f_* \mid \underline{\underline{X}}, \underline{\underline{y}}, \underline{x}_*)$. For this purpose, we will first consider the joint distribution of the known observations $\underline{y}$ and the desired function value f_* of percentage of presence of the considered object in the investigated window. Thanks to the assumption of a GP over $\underline{y}$ (Eq. 3.6) and to the marginalization property of Gaussian processes, the joint distribution is Gaussian. The desired predictive distribution can be derived simply by conditioning the joint one to the noisy observations $\underline{y}$ and takes the following expression (for more details we refer the reader to [73]):

$$p\left(f_* \middle| \underline{\underline{X}}, \underline{\underline{y}}, \underline{x}_*\right) \sim N(\mu_*, \sigma_*^2) \tag{3.8}$$

where:

$$\mu_* = \underline{k}_*^T \cdot \left[\underline{\underline{K}}\left(\underline{\underline{X}}, \underline{\underline{X}}\right) + \sigma_n^2 \underline{I}\right]^{-1} \cdot \underline{y} \tag{3.9}$$

$$\mathrm{\sigma}_*^2 = \mathrm{k}\left(\underline{\mathrm{x}}_*, \underline{\mathrm{x}}_*\right) - \underline{\mathrm{k}}_*^{\mathrm{T}} \cdot \left[\underline{\underline{\mathrm{K}}}\left(\underline{\underline{\mathrm{X}}}, \underline{\underline{\mathrm{X}}}\right) + \mathrm{\sigma}_{\mathrm{n}}^2 \underline{\mathrm{I}}\right]^{-1} \cdot \underline{\mathrm{k}}_* \tag{3.10}$$

The vector $\underline{k}_*$ denotes the covariance values between the training samples $\underline{\underline{X}}$ and the sample $\underline{x}_*$ whose prediction is looked for. These are the key equations in the GP regression approach. From these equations it is possible to draw two important considerations: 1) μ_* expresses the best output value estimate for the investigated sample according to Eq. 3.6 and depends on the covariance matrix $\underline{\underline{K}}\left(\underline{\underline{X}}, \underline{\underline{X}}\right)$, the kernel distances between training and test samples $\underline{k}_*$, the noise variance σ_n^2, and the training observations; and 2) the variance σ_*^2 represents the confidence measure associated by the model to the output. Inside the GP regression model the covariance function $k(\underline{x},\underline{x}')$ assumes great importance as it embeds the geometrical structure of the training samples. In some sense, it conveys the prior knowledge about the output function $F(\cdot)$. A common choice for the covariance function is the squared exponential function [73]:

$$k_{SE}(\underline{x},\underline{x}') = \sigma_f^2\ exp\left(-\frac{\left|\underline{x}-\underline{x}'\right|^2}{2l^2}\right) \tag{3.11}$$

The two hyperparameters σ_f^2 and *l* are called process (signal) variance and length scale, respectively.

3.3.3. Model Selection Issue

GP regression models need the determination of the hyperparameters σ_f^2, l, and σ_n^2 which we will group in the vector $\underline{\theta}$. The process which leads the tuning of these parameters is called model selection. It is of crucial importance because if affects the prediction accuracy of the system. The model selection is formulated within a Bayesian framework and it relies on the idea to maximize the posterior probability distribution defined over the vector of parameters $\underline{\theta}$ [73]:

$$p\left(\underline{\theta}\,\middle|\,\underline{\underline{X}},\underline{y}\right) = \frac{p\left(\underline{y}\,\middle|\,\underline{\underline{X}},\underline{\theta}\right)\cdot p(\underline{\theta})}{p\left(\underline{y}\,\middle|\,\underline{\underline{X}},\right)} \tag{3.12}$$

Often, the evaluation of the denominator in Eq. 3.12 is analytically intractable. As a solution, one may resort to the type II maximum likelihood (ML-II) estimation procedure. It consists in the maximization of the marginal likelihood (evidence), that is the integral of the likelihood times the prior:

$$p\left(\underline{y}\,\middle|\,\underline{\underline{X}},\underline{\theta}\right) = \int p\left(\underline{y}\,\middle|\,\underline{f},\underline{\underline{X}},\underline{\theta}\right)\cdot p\left(\underline{f}\,\middle|\,\underline{\underline{X}},\underline{\theta}\right) d\underline{f} \tag{3.13}$$

with the marginalization over the latent function *f*. Under a GP regression modeling, both the prior and the likelihood follow Gaussian distributions. After some manipulation, it is possible to show that the log marginal likelihood can be written as:

$$\begin{aligned} log\,p\left(\underline{y}\,\middle|\,\underline{\underline{X}},\underline{\theta}\right) = \quad & -\frac{1}{2}\underline{y}^T\cdot\left(\underline{\underline{K}}\left(\underline{\underline{X}},\underline{\underline{X}}\right)+\sigma_n^2\underline{\underline{I}}\right)^{-1}\cdot\underline{y}\,+ \\ & -\frac{1}{2}log\left|\underline{\underline{K}}\left(\underline{\underline{X}},\underline{\underline{X}}\right)+\sigma_n^2\underline{\underline{I}}\right|+ \\ & -\frac{n}{2}log(2\pi) \end{aligned} \tag{3.14}$$

This last Eq. 3.14 can be seen as the sum of three terms which represent the capability of the model to fit the data, the model complexity penalty and a normalization constant, respectively. From an implementation viewpoint, this maximization problem can easily be solved by a gradient-based search routine [73]

3.4. Experimental Results

In the experimental activity, we performed two separate experiments, each aiming at assessing the proposed technique to detect a given class of objects very common in urban scenarios, namely vehicles and solar panels, respectively. The sets of images were acquired by means of a UAV equipped with imaging sensors spanning the visible range. The images were taken over different areas in or near the city of Trento, Italy, and at different times. All the acquisitions were performed with a picture camera Canon EOS 550D characterized by a CMOS APS-C sensor with 18 megapixels. The images georeferenced and are characterized by a spatial resolution of approximately 2 [cm] and by three channels (RGB). The size of the acquired images is of 5184 x 3456 with 8 bit of radiometric resolution.

3.4.1 Experiments 1 – Vehicle Detection: Dataset Description and Setups

To test the potentiality of the presented filter to detect vehicles in urban areas, several acquisitions over different places have been performed. From the whole datasets, thirteen images have been selected and divided in three groups:

1) *Training group*. It is composed of three images which represent three parking lots and where it is possible to identify several cars parked with different orientations. By using manually created masks (i.e., car masks), we constructed the training set of samples representing cars and background used to train the GP regression model. For such purpose, by using a sliding window process over the training images we assigned to each window a target value which represents the percentage of presence of a vehicle inside the considered window. From the training images, by using a step size of the sliding window of 40, we identified 984 and 1215 car and background windows, respectively. The training of a GP regression model by using the original size of the sliding window would be too demanding. Thus, the two simple feature reduction techniques described above have been applied. In case of regular sampling, we decided to keep a window sample every 16 in order to pass from a window size of 40000 (200×200 pixels) to 2500 for each image gradient feature order. Regarding the PCA technique, we decided to keep 90% of total information which allowed us to pass from 40000 to 1650 features for each gradient order.

2*) Validation group*. To this group belong 2 images that are used to estimate empirically the discrimination threshold on the estimates of the GP regression model. The images belonging to this group show two parking lots in which the cars (6 in the first image and17 in the second one) are easily recognizable. The identified best threshold value is the one that generates the maximum ratio between the number of cars correctly identified and the number of false alarms. This step has great importance because it allows setting a threshold value generalizable to other UAV flights at similar acquisition conditions.

3) *Test group*. To this group belong the images on which we assessed the accuracy of the proposed methodology. It includes eight different images, acquired over different areas and at different times in order to test the methodology in different conditions. We selected two images representing two big parking lots in which it is possible to identify 278 and 149 cars respectively, an image showing a medium-size parking lot (i.e., 56 cars) of a shopping center, four images representing standard urban areas in which it is possible to detect 19, 15, 51 and 31 cars, respectively and finally an image with only three cars to assess our technique in a situation where the presence of cars is rare.

In the case of the detection of cars in urban scenario, it is possible to assume that the vehicles lie only on asphalted areas such as roads or parking lots. Therefore, a screening operation can be performed by obtaining the information about road maps possibly available from Geographic Information Systems (GIS) [71]. By knowing a priori the area of interest one could restrict the investigated areas only to these regions obtaining two significant advantages: 1) improve the velocity of detection by limiting the areas to analyze; and 2) reduce the number of false alarms. In the following, we will experiment our car detector with and without screening.

The experimental assessment has been organized in two phases. In the first stage, a GPR model was trained and the best threshold value was estimated by using the training and validation groups, respectively. The second phase consists in the assessment of the results on the test images. At the end of the validation step, we found that the best threshold value to apply on the estimates of the GP regression model is 0.4 in the presence of screening (see Fig. 3.3). By contrast, without screening, the threshold has been raised to 0.5. This increment is necessary to limit the number of false alarms that otherwise would increase drastically.

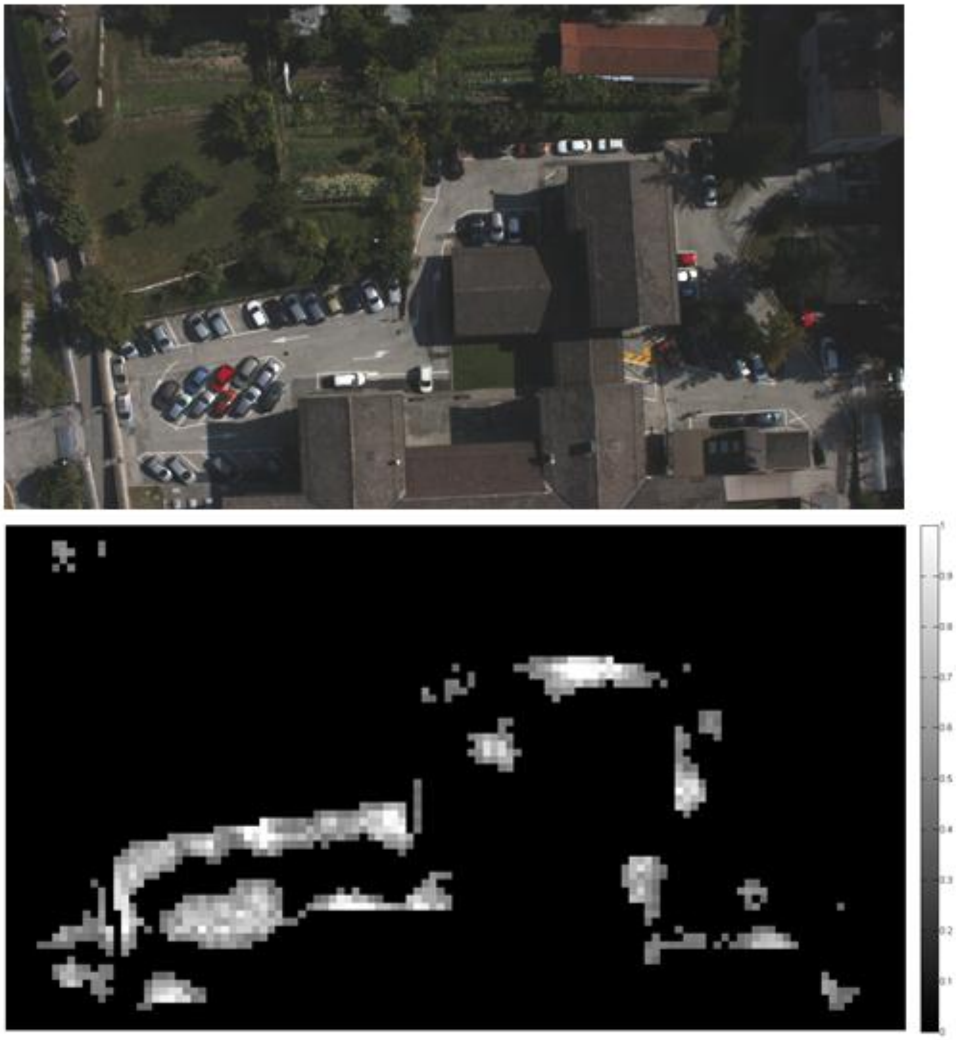


Figure 3.3. Illustration of the output provided by the detector (trained to identify cars) before thresholding.

It is noteworthy that our method does not provide directly an estimate of the number of cars. This last can however be inferred in a straightforward way, that is by dividing the total area of the regions identified by the detector as regions of cars to the expected area covered by a single car. In our case, given that the spatial resolution is equal to 2 [cm] and assuming that a standard car length and width is equal to 4.5 [m] and 1.8 [m], respectively, it may be reasonably assumed that on an average a single car covers around 21000 pixels.

The quantitative results are expressed in terms of producer's accuracy versus user's accuracy. Producer's accuracy (*i.e.*, *Pacc*) shows the number of correct cars with respect to the real number of cars. Instead, the user's accuracy (*i.e., Uacc*) compares the number of correct cars with the number of identified cars:

$$Pacc = \frac{TP}{M} \tag{3.15}$$

$$Uacc = \frac{TP}{TP + FP} \tag{3.16}$$

where *TP* is the number of cars correctly identified, *FP* is the number of false alarms and *M* represents the number of cars really present in the scene. We provide also the average of these two accuracies (*i.e.*, *Acc*). Finally, all the experiments were conducted in Matlab R2013b on an Intel Core i5-2400 CPU @ 3.10 GHz with 4 GB RAM.

3.4.2. Experiments 1 – Vehicle Detection: Results

The results of the proposed methodology applied to the detection of vehicles are detailed in the following paragraphs. We report the results obtained by using both previously described feature reduction techniques, namely regular sampling and PCA. The results in Tables 3.I-3.II show the performances (accuracies and processing times) obtained without the screening operation, instead those in Table 3.III-3.IV represent the performances with screening. A first look at the results makes clear (as expected) the importance of the screening operation. Indeed, with both feature reduction techniques, the final results increase of about 20% thanks to a considerable reduction of the number of false alarms. In all the situations, the Producer's accuracy, i.e., the one that gives information about the detection capability of the methodology, remains constant, while the User's accuracy, which is linked to the number of false alarms, changes substantially if one applies or not a screening operation.

TABLE 3.I. CAR DETECTION ACCURACIES IN PERCENT ACHIEVED ON THE TEST IMAGES USING REGULAR SAMPLING AND WITHOUT SCREENING OPERATION. TIMES EXPRESS THE AVERAGE TIME NEEDED TO PROCESS AN IMAGE (OF 5184 × 3456 PIXELS).

Gradient Order	TP (true positive)	FP (false positive)	M (cars present)	User's Accuracy	Producer's Accuracy	Total Accuracy	Average Time (s)
1	65	353	598	15.55	10.87	**13.21**	**22.5**
2	476	770	598	38.20	79.6	**58.91**	**30.6**
3	473	676	598	41.17	79.1	**60.14**	**38.7**
4	483	764	598	38.73	80.77	**59.51**	**48.1**

TABLE 3.II. CAR DETECTION ACCURACIES IN PERCENT ACHIEVED ON THE TEST IMAGES USING PCA REDUCTION AND WITHOUT SCREENING OPERATION.

Gradient Order	TP (true positive)	FP (false positive)	M (cars present)	User's Accuracy	Producer's Accuracy	Total Accuracy	Average Time (s)
1	440	971	598	31.18	73.58	**52.38**	**175.7**
2	465	956	598	32.72	77.76	**55.24**	**346.3**
3	474	908	598	34.29	79.26	**56.78**	**504.9**
4	480	835	598	36.50	80.26	**58.38**	**732.2**

TABLE3.III. CAR DETECTION ACCURACIES IN PERCENT ACHIEVED ON THE TEST IMAGES USING REGULAR SAMPLING AND WITH SCREENING OPERATION.

Gradient Order	TP (true positive)	FP (false positive)	M (cars present)	User's Accuracy	Producer's Accuracy	Total Accuracy	Average Time (s)
1	233	68	598	77.41	38.96	**58.18**	**10.6**
2	511	156	598	76.61	85.45	**81.03**	**14.1**
3	527	137	598	79.37	88.13	**83.74**	**19.3**
4	517	136	598	79.17	86.6	**82.81**	**24.4**

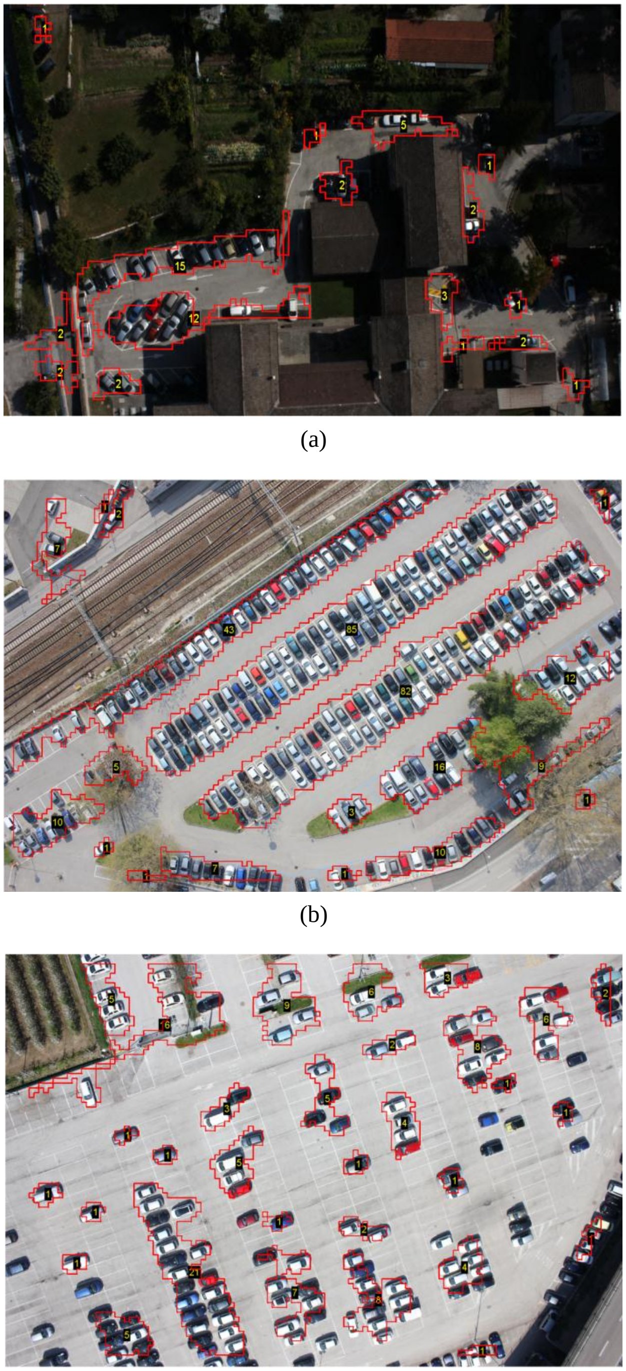
TABLE3.IV. CAR DETECTION ACCURACIES IN PERCENT ACHIEVED ON THE TEST IMAGES USING PCA REDUCTION AND WITH SCREENING OPERATION.

Gradient Order	TP (true positive)	FP (false positive)	M (cars present)	User's Accuracy	Producer's Accuracy	Total Accuracy	Average Time (s)
1	480	113	598	80.94	80.26	**80.61**	**82.1**
2	487	120	598	80.23	81.44	**80.83**	**141.0**
3	500	115	598	81.30	83.61	**82.54**	**202.6**
4	516	114	598	81.91	86.28	**84.09**	**259.6**

As we initially supposed, the exploitation of image gradient features of high order could bring some benefit for a better detection of the objects, in this case, of cars. In Tables 3.I-3.IV, we show the results yielded by accumulating gradually the gradients from order 1 to order 4. With regular sampling as feature reduction means, the accuracy improvement is sharp from the first order to the second one. After the second order, the accuracy improves slightly or remains stable. In both situations, with and without screening, the percentage differences from the first to the second order are about 30% which represents an increment of correct detections of 411 and 278 vehicles, respectively. The number of false alarms increases of 417 without screening and of 19 in the other case. When PCA is used to reduce the number of features, the gaps between the first two orders are less marked. The total increment of true positives from the first to the fourth order is of 40 and 36 cars, respectively, which means 4% of improvement of the total accuracy. Comparing regular sampling and PCA, the final results look very similar but PCA exhibits a better stability. This can be explained by the fact that PCA projects in a subspace while maintaining most of the information content (in our case 90%), which is not the case with a trivial regular sampling.

From the point of view of the average computation time per image, regular sampling is definitely better. This huge difference (up to 700 seconds in the case of extraction of four orders of image gradients – see Tables 3.I-3.II) is due to the matrix operations needed by PCA for performing the projections.

By analyzing the qualitative results (see Figure 3.4), it is possible to see how the proposed methodology works very well to detect clusters of cars (parked near each others), but it shows some limitations in the detection of isolated cars. A smaller value of the threshold on the estimates of the GP regression model makes it possible to detect also these cars but, at the cost of a higher number of false alarms. Outside parking lots in urban areas, the objects that generate false alarms are pedestrian lines and sidewalks which have rectangular shapes very similar to vehicles.



(a)

(b)

(c)

Figure 3.4. Final results related to the detection of cars obtained on three test images (a), (b) and (c). For each detected region (surrounded by a red contour), a number is provided in a small black square, which indicates the estimated number of cars in the corresponding region.

3.4.3 Experiments 1 – Vehicle Detection: Comparison with Reference Work

In order to further assess the goodness of the proposed detector, we compared it with a recently developed method specifically developed for detecting cars and described in [71]. This method detects and counts cars in urban scenarios by exploiting similarity measures and a Support Vector Machine (SVM) classifier. Filtering operations in the horizontal and vertical directions are performed to extract histogram-of-gradient features and to assign a signature based on a similarity measure to each window. The SVM classifier is used to achieve the discrimination. In the last part of the detection procedure, a merging step is implemented to group the windows which refer to the same car. At the end, the user can obtain information about the number of cars present in the scene, as well as the exact position and orientation of each detected car.

TABLE3.V. COMPARISON OF THE DETECTION PERFORMANCES OBTAINED BY THE PROPOSED OBJECT DETECTOR (HG-GPR) AND THE CAR DETECTION METHOD IN **[71]**. BOTH WERE APPLIED WITH SCREENING OPERATION.

	M (cars present)	**TP (true positive)**	**FP (false positive)**	**Total Accuracy**	**Average time (s)**
HG-GPR with regular sampling	598	517	136	**82.81**	**48**
HG-GPR with PCA reduction	598	516	114	**84.09**	**732**
Method [71]	598	466	133	**77.86**	**12600**

A numerical comparison between the two methods is provided in Table 3.V. The improvements yielded by the method proposed in this work (HG-GPR detector) are significant. Indeed, the presented methodology correctly detects 50 cars more and, at the same time, it is less affected by false alarms, namely 114 against 133. The overall accuracy gain is more than 5%. Moreover, the computational time is substantially reduced and passes from 3.5 hours to just 48 seconds for processing a single image. However, the drawback of the HG-GPR detector is that it does not provide information about the exact location and orientation of each vehicle, but just yields an indication of the regions containing cars and an estimate of the number of cars in each region.

3.4.4. Experiments 2 – Solar Panel Detection: Data Set Description and Setups

The other class of objects investigated in this work is the solar panels, which are little by little invading urban areas. Also in this set of experiments, several acquisitions have been performed to simulate different conditions and different scenarios. Among the acquired images, ten have been selected to compose the three groups used to train the GP regression model and to test the whole detection procedure.

1) *Training group.* This group is composed by 4 images which represent 3 urban areas in which it is possible to identify several buildings with solar panels and an image that shows a big building which has tens of solar panels on the roof. The GP regression model is trained, by using a step size of the sliding window of 20, with 576 and 750 solar panel and background windows, respectively. Also in the case of solar panels, since the sliding windows size is fixed to 100×100 pixels, feature reduction has been applied so as to

reduce the number of features from 10000 to 625 and 729 for each gradient order by using the regular sampling and the PCA, respectively.

2) *Validation group*. It includes one image representing an urban area (i.e., 63 windows containing solar panels). As done in Experiments 1, the validation group is used to calibrate the threshold on the estimates of the GP regression model.

3) *Test group.* We used two images representing two big buildings with hundreds of solar panels on the roof (more exactly 2711 and 1132 windows representing solar panels, respectively). Two more images representing urban areas where it is possible to find some houses with solar panels on the roofs (i.e., 450 and 340 windows, respectively) as well as a fifth image with no solar panel (to test the methodology in absence of the investigated object) were considered. Usually, solar panels have different appearances and are arranged in different ways depending on the nature of the roof or on the available space. Therefore, it is not possible to compute a priori the expected area covered by a single solar panel, making it in this case not possible to estimate the number of solar panels as previously done for the cars. Accordingly, the accuracy of the detector will be computed with respect to a ground-truth manually created.

3.4.5. Experiments 2 – Solar Panel Detection: Results

By running the methodology on the validation image, the best value of the discrimination threshold was found equal to 0.4 for both feature reduction techniques. The numerical results achieved on the test images are listed in Tables 3.VI-3.VII. From these results, it is possible to find several analogies with what found for the detection of vehicles. The performances of both feature reduction techniques are similar. With the regular sampling, the addition of gradient features of increasing orders helps in improving the detection. Especially the difference between the first and the second order is marked, accompanied by a strong reduction of the number of false alarms. Instead, PCA provides a more stable behavior showing that for objects with very regular structures like solar panels one can afford the cost of the use of high-order gradients.

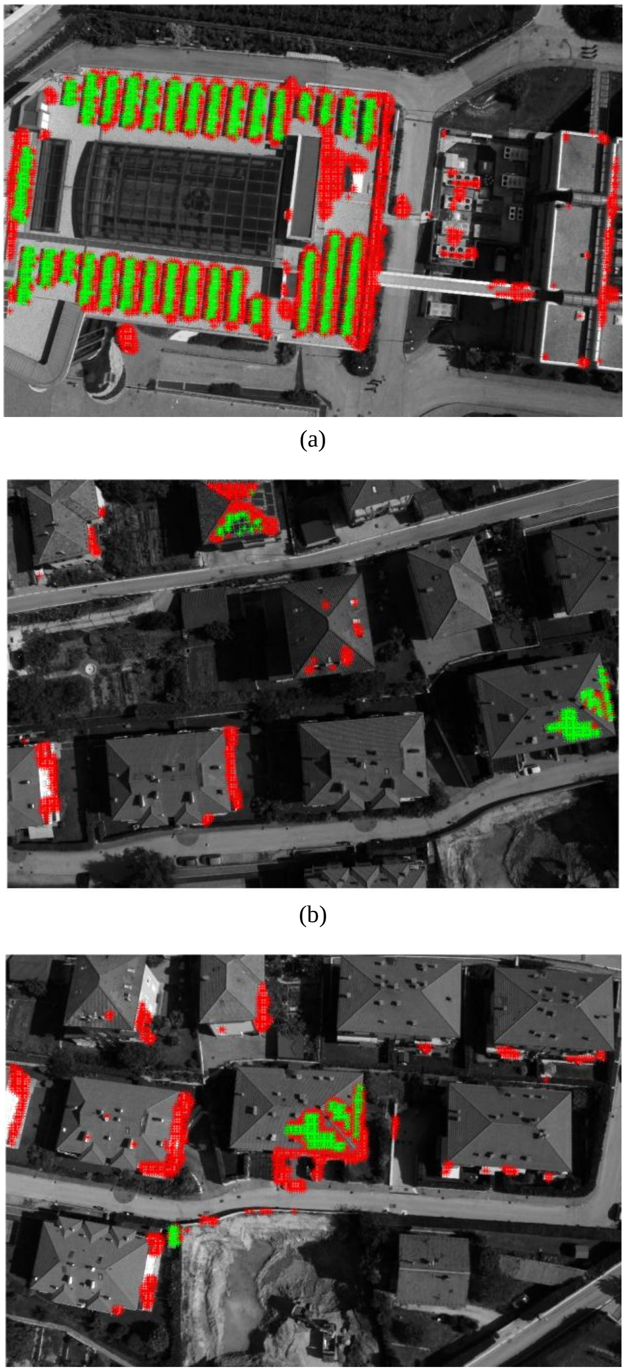
TABLE 3.VI. SOLAR PANEL DETECTION ACCURACIES IN PERCENT ACHIEVED ON THE TEST IMAGES USING REGULAR SAMPLING.

Gradient Order	User's Accuracy	Producer's Accuracy	Producer's Accuracy (No BA)	Total Accuracy	Total Accuracy (No BA)	Average Time (s)
1	94.04	14.09	14.24	**54.07**	**54.64**	**57.62**
2	79.58	35.08	40.52	**57.33**	**60.05**	**68.74**
3	77.29	40.38	45.70	**58.83**	**61.49**	**70.53**
4	89.49	34.47	40.11	**61.98**	**64.79**	**78.54**

TABLE 3.VII. SOLAR PANEL DETECTION ACCURACIES IN PERCENT ACHIEVED ON THE TEST IMAGES USING PCA REDUCTION.

Gradient Order	User's Accuracy	Producer's Accuracy	Producer's Accuracy (No BA)	Total Accuracy	Total Accuracy (No BA)	Average Time (s)
1	80.57	44.69	52.14	**62.63**	**66.36**	**95.08**
2	69.27	49.81	57.00	**59.54**	**63.13**	**196.73**
3	77.32	46.64	54.77	**61.98**	**66.04**	**267.88**
4	81.68	40.43	47.27	**61.01**	**64.48**	**370.25**

For a better reading of the results, we introduce the concept of border alarms. A border alarm is an alarm which happens when the center of the analyzed window is not over a solar panel but this last is anyway covered partially by the window. This partial covering (of the object) stimulates enough the detector to make it generate a wrong positive detection. A border alarm cannot be viewed as a true alarm since this last occurs



(a)

(b)

(c)

Figure 3.5. Final results related to the detection of solar panels obtained on three test images (a),(b) and (c). Green crosses indicate correct positive detections, while red ones represent false positives (including border alarms).

when the detector generates a wrong positive detection while the window is completely object-free. In Tables 3.VI-3.VII, we have integrated the results with accuracies excluding the border alarms (no BA). In general, if we exclude the border alarms, the final accuracy raises of about 3-4% and in the best case reaches 66.36%, suggesting that border alarms impact significantly in the accuracy assessment.

The qualitative results (Fig. 3.5) show the capability of the proposed methodology to localize satisfactorily the areas containing solar panels. False alarms are either border alarms (at the borders of solar panels) or restricted to objects having a structure similar to that of solar panels, such as windows on the facades or on the roof of buildings, or small rectangular-shaped balconies. It is noteworthy that in most of the cases these objects are difficult to discriminate from solar panels even by visual inspection.

Regarding computation times, the results are comparable with those obtained with the detection of vehicles even if times are slightly shorter because of the smaller dimension of the analyzed windows.

3.5 Conclusions

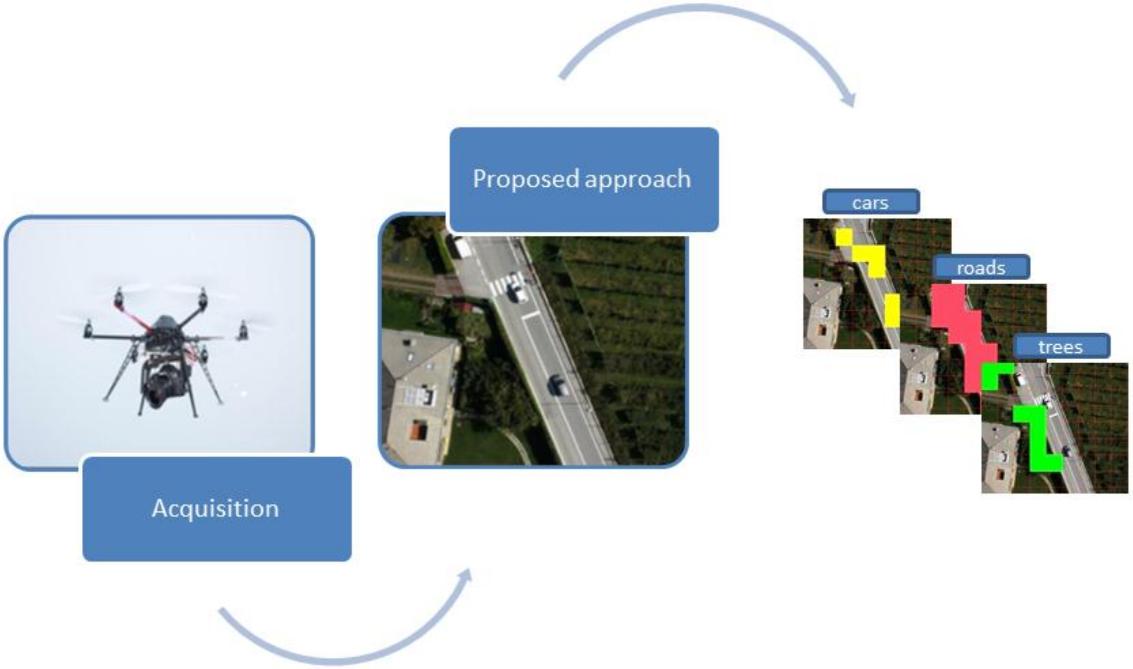
In this work, we have proposed a new methodology for the detection of objects in urban areas from images acquired by means of UAV. This methodology revolves around two main ingredients: 1) object description with gradients of high order; 2) nonlinear filtering model based on GP regression. Image gradients extracted at different orders can help in better describing the structure of objects to detect, in particular complex objects. This is experimentally confirmed for the detection of vehicles, for which a substantial boost of the detection accuracy was achieved by passing from the first to the fourth order of the gradient. The GP model fed with high-order gradients permits to yield an estimate of the presence of the object of interest for a given position of the sliding window within the image. Such estimate is then binarized through a simple thresholding operation. The choice of the GP modeling is motivated by the fact that is fast and accurate. Moreover, it gives the possibility to automatically tune its free parameters.

From the experimental results, it emerges that the proposed detector works satisfactorily well. The final accuracies are higher than 80% in case of vehicles and higher that 60% for solar panels. Its fast processing capability makes it particularly interesting. It can exhibit a computation time per image of the order of one minute for images of very large dimensions. Its main limitation is that it does not detect objects singularly but captures the areas where the objects lie. A counting of the objects is not directly provided but needs to be a posteriori inferred.

Regarding future developments, two directions deserve to be explored to further improve the proposed detector: 1) reducing the border alarms through for instance a post-processing based on morphological filtering or a Markovian binarization; and 2) handling better the scale invariance issue through a multiresolution image analysis.

4. A Multiclass Tile-based Analysis Method for UAV Imagery

Abstract *– This chapter presents a novel method to "coarsely" describe extremely high resolution (EHR) images acquired by means of unmanned aerial vehicles (UAV) over urban areas. It consists first in the subdivision of the original UAV image in a grid of tiles. Then, each tile is compared with a library of training tiles to inherit the binary multilabel vector of the most similar training tile. This vector conveys a list of classes likely present in the considered tile. Our multiclass tile-based approach needs the definition of two main ingredients: 1) a suitable tile representation strategy; and 2) a tile-to-tile matching operation. Various tile representation and matching strategies are investigated. In particular, we present three global representation strategies which process each tile as a whole and two point-based strategies which exploit points of interest within the considered tile. Regarding the matching strategies, two simple measures of distance, namely the Euclidean and the chi-squared histogram distances are explored. Interesting experimental results conducted on a rich set of real UAV images acquired over an urban area are reported and discussed.*



4.1. Introduction

Unmanned Aerial Vehicles (UAVs) have been opening a large assortment of remote sensing applications. UAVs are rapid, efficient and flexible acquisition systems. Nowadays, they represent a valid alternative or a complementary solution to satellite or airborne devices especially for small coverage or inaccessible areas. Thanks to their timely and extremely rich data acquisition capacity with respect to other acquisition systems, UAVs are emerging as innovative and cost-effective devices to perform urban and environmental survey tasks. Originally, the exploitation of UAVs was restricted to military purposes but, in the last years, they have invaded civilian application fields. For instance, they are gaining a remarkable success for agricultural and environmental analyses. Indeed, the use of different sensors, such as thermal and multispectral sensors, combined with the extremely high resolution (EHR) of UAV imagery allows providing suitable crop analysis solutions. In [6], the authors offer an interesting overview about the use of UAVs for precision agriculture purposes. In particular, they show the different applications and the range of available sensors and platforms suitable for this task. Gonzalez-Dugo *et al.* [8] present a work in which they describe the spatial variability of crop water status in thermal imagery of a commercial orchard where five different fruit tree species are cultivated.

Also in urban scenarios, UAVs have been used with promising results in various applications. In [72], the authors present a method based on the combination of a support vector machine (SVM) classifier and invariant features (i.e., SIFT) to detect vehicles. Since extremely high resolution images contain a large quantity of information, they need to be handled with suitable strategies to exploit all their potential. In this context, the use of point-based representations to detect cars has proven particularly interesting. Another attractive method to detect and count vehicles in urban scenarios is presented by Moranduzzo and Melgani [71]. It exploits a beforehand built catalogue, in which histogram of gradient (HoG) features of positive and negative patches are stored, and the similarity measure is performed through SVM modelling. UAVs have also demonstrated to be helpful in urban emergency situations. For instance, in [14], an interesting study in which several methods to automatically detect people lying on the ground is presented and discussed. In the archaeological context, it is also shown how UAV technology offers practical and inexpensive solutions to support archaeological analyses [16]. Moreover, Saleri *et al.* [17] describe a complete archaeological survey starting from the collection of the data to the creation of accurate 3D models.

Image classification and analysis represent one of the most active research areas within the remote sensing community. The possibility to detect the presence of specific classes of objects from very high resolution (VHR) remote sensing images is certainly one of the reasons motivating the widespread exploitation of remote sensing images. In particular, the classification problem in satellite and airborne images has been deeply investigated. For instance, in [77], the authors propose a fuzzy decision tree-SVM classifier for the classification of high-dimensional images. Xu *et al.* [78] introduce a representation of VHR aerial images based on a bag-of-visual words strategy and a combination of spectral and texture features. Li *et al*. [79] expose a technique for the classification of hyperspectral imagery which exploits a combination of a one-against-one strategy with a kernel-based discriminant analysis. The multiclass problem is decomposed in binary classification problems and the final decision is taken on the basis of several decision-fusion rules. Shiyong and Datcu [80] present a patch-based method for the multi-temporal analysis of high resolution images. In particular, they present two methods devoted to the change detection and evolution pattern analysis of multi-temporal images.

Despite the fact that they can achieve very good accuracies for satellite or airborne imagery, the current classification approaches cannot be directly applied to UAV images. As mentioned before, UAV images are characterized by an extremely high spatial resolution (up to few centimeters). If in satellite images an object is described by a few pixels, the same object in a UAV image is represented by thousands of pixels. Moreover, the jump from satellite/airborne to UAV images involves, if not a complete redefinition of the classes, at least a sharp qualitative and quantitative change in the definition of classes characterizing a given area.

In this chapter, we present a multiclass analysis approach for the description of images specifically acquired with UAVs. The extremely high spatial resolution of UAV images makes the aspect of objects which belong to the same class very different. Accordingly, the identification of a detection strategy which works simultaneously well for numerous classes of objects is not trivial. To overcome this problem, we propose an innovative method which aims at "coarsely" describing a given UAV image. It starts with the subdivision of the image into a grid of tiles, and then associates a list of objects present in each tile thanks to an opportune tile-based matching process. In [81], an unsupervised semantic labelling framework to monitor nuclear proliferation in multi-spectral satellite images is presented. The method extracts several kinds of feature from 128×128 pixels non-overlapping tiles, which are used to train a latent Dirichlet allocation model. Differently from satellite images, UAV images cover much smaller areas, but due to their extremely high resolution the information amount they convey is huge. Our tiling approach for UAV images allows exploiting all the information conveyed in each tile. In particular, different tile representation and matching strategies are proposed and discussed.

This chapter is organized as follows. Section 4.2 depicts the main concepts behind the proposed multiclass tile-based analysis methodology. Section 4.3 and Section 4.4 expose the investigated tile representation and matching strategies. In Section 4.5, we illustrate the experimental results. Finally, Section 4.6 is devoted to the conclusions of the work and to future developments.

4.2. Methodological Overview

Let us consider an extremely high resolution (EHR) image $I(x, y)$ (where (x, y) represent the pixel coordinates in image I) acquired in the visible range (RGB channels) by means of an UAV over a given area of interest. As mentioned above, EHR images are characterized by the fact that even very small objects may be described by thousands of pixels in the image. A so huge amount of information per single object calls for new image analysis strategies.

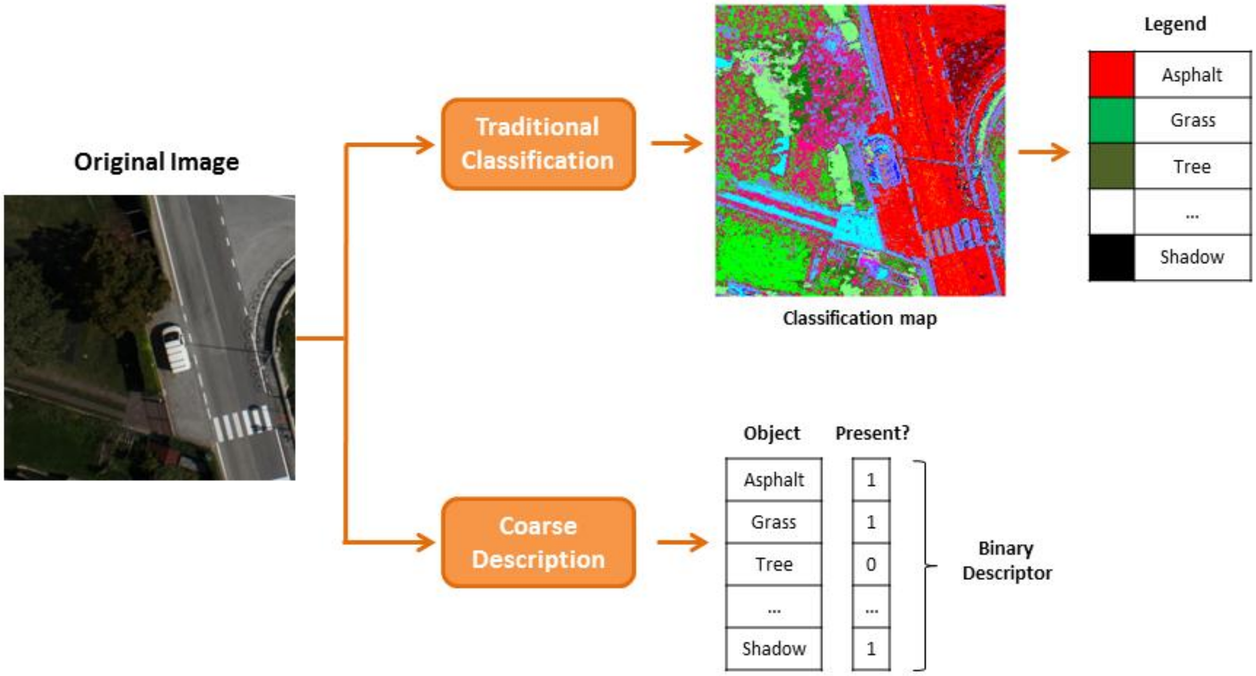


Figure 4.1. Illustration of the conceptual difference between traditional image classification and coarse image description for a UAV image.

The goal of this work is to develop a method that automatically describes the content of a given UAV image which may convey numerous classes of objects. A standard classification of the image (*e.g.*, pixel-based) can reveal ineffective for the intrinsic variability of the classes. For instance, the class of cars is so variable in the color space that single pixels cannot be used for discriminating cars from other classes. An alternative could be to adopt point-based or segment-based descriptors. When dealing simultaneously with numerous classes, this raises however the problem of the spatial merging of the descriptors besides the dramatic increase of the computational needs. An alternative and innovative solution we propose in this work consists in, what we will call, a “coarse” description of the image. Differently from standard classification techniques, a “coarse” description does not aim at assigning a label to each single pixel or descriptor, but simply draws a list of possible classes of objects present in the image (Fig. 4.1). Coarse description thus handles the image as a unique entity and associates to this entity multiple labels. This new way of approaching the image analysis problem brings advantages but also drawbacks. The main advantage is that it simplifies substantially the analysis process, while its most important drawback is that information about the number of objects of the same class as well as their spatial location in the image is lost. Actually, the user may not always be interested in getting such precise information but just in knowing if some classes are present or not in a given image (e.g., in image archiving). In order to attenuate this drawback of the coarse image description and since UAV images convey intrinsic redundancy (due to their spatial resolution), we propose to marry the coarse description idea with a tile-based analysis of the image. This means that the considered image is first divided into a grid of N equal tiles P and then each tile is coarsely described (Fig.

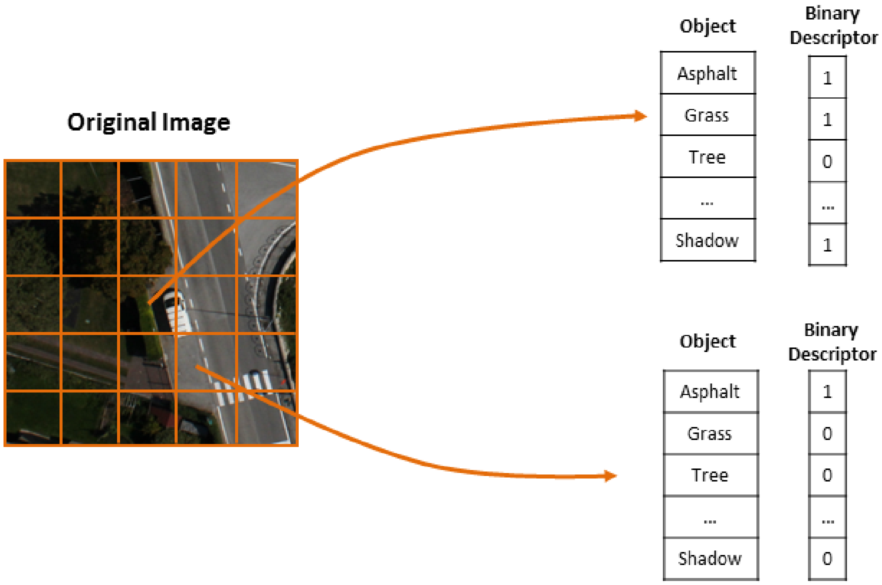


Figure 4.2. Tailing and coarse description of an original UAV image

4.2). The size of the tiles could be inferred from the relationship between the expected object sizes and the image resolution.

Assuming a library of tiles has been beforehand built so as to be exploited as training tiles each multilabeled with a binary descriptor, our multiclass tile-based approach needs the definition of two main ingredients: 1) a suitable tile representation strategy; and 2) a tile-to-tile matching operation (Fig. 4.3). All training tiles are associated with a binary vector that indicates which of the predefined classes of objects are present. Each unlabeled tile *P* is labeled with the same binary vector of the most similar tile present in the training library. Indeed, if two analyzed tiles are very close in the representation space, it is likely that they own analogous information content, i.e. they convey the same classes of objects.

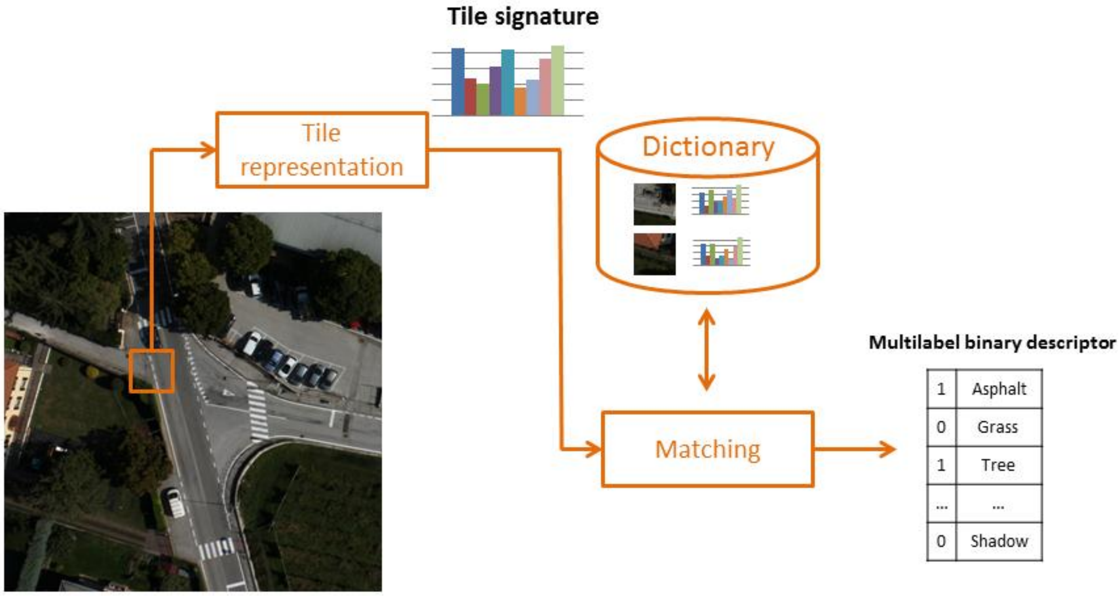


Figure 4.3. Flow chart of the proposed coarse description procedure

Both the investigated tile-representation and matching solutions are described in the following two sections, respectively.

4.3. Tile Representation Strategies

As mentioned before, the way how the tiles are represented is of key importance. For such purpose, in this section, we explore different representation strategies (Table 4.I). First, we present two global strategies that assign to each tile a unique and compact signature (*i.e.*, color and HoG representations [54] integrated in a bag of words (BOW) model). A third strategy consists in an opportune fusion of the previous two strategies. Then, in the second part of this section, we explore two additional point-based strategies (*i.e.*, SIFT [25] and SIFT plus local color representation integrated in a bag of words (BOW) strategy) which, instead of providing a unique signature as the previous strategies, associate to each tile a set of points of interest described with their local properties.

TABLE 4.I. LIST OF INVESTIGATED REPRESENTATION STRATEGIES

Name	Acronym	Features
Color-Representation	CR-B	RGB
HoG-Representation	HR-B	HoG
HoG-Color-Representation	HCR-B	HoG and RGB
SIFT Representation	SR	SIFT
SIFT-Color Representation	SCR-B	SIFT-RGB

4.3.1. Global Representation Strategies

In the context of EHR images, the quantity of information conveyed by each tile is huge, thus calling for opportune and compact representation forms. The underlying idea of the proposed strategies is to derive signatures which describe the appearance and the structural information of the corresponding tiles.

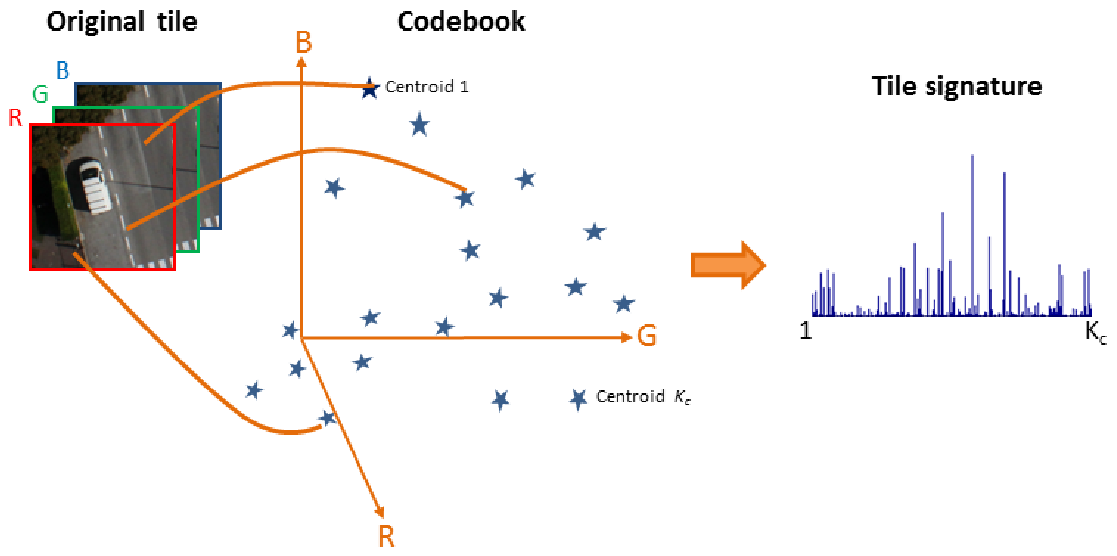


Figure 4.4 Tile signature extraction according to the color-based representation coupled with a bag-of-words modeling (CR-B strategy).

The first approach (*i.e.*, CR-B) consists in the extraction of a signature S_C which encodes the colors and, consequently, the aspect of the scene. For each tile, the simple use of all pixel values (*i.e.*, RGB values) would create signatures too large (curse of dimensionality) and unsuitable in terms of processing time. A formulation of the tile representation problem under a bag of words (BOW) model could represent an interesting solution to overcome these issues. Essentially, BOW maps the set of extracted feature vectors (in our case, the RGB vectors) into a fixed size histogram of visual words [82]. In greater detail, first, let us consider a set of training tiles, each characterized by a binary multilabeling vector. These binary vectors actually will not be exploited during the representation phase, but later in the matching phase (described in the next section). All the pixels of all training tiles are projected in the RGB space and the K-means clustering algorithm, [83], is applied on the resulting clouds of points in order to generate a so-called codebook, namely a set of K_c words each referring to a cluster centroid. Every tile of the considered image (including training tiles) is switched from a RGB to a BOW representation. This is done by assigning each pixel of the tile to the closest centroid (word) and incrementing the counter of this last. At the end, a compact histogram (signature) S_c encoding the number of occurrences of each word in the tile is obtained (Fig. 4.4).

The second proposed strategy (HR-B) aims at extracting signatures which describe the tile structures (geometric shapes of objects). To fulfill this purpose, we will resort to histogram of gradients (HoG) features, particularly suited to highlight the shapes of objects within images. In more details, HoG features are obtained by dividing the investigated tile into a set of overlapping cells and by extracting from each cell 8-bin histograms of gradient directions. The histograms are concatenated to form the full descriptor, which is then normalized to make it invariant to illumination and shadowing changes. The original HoG formulation exhibits the advantage of expressing very well the shapes within the tile but the drawback of not being enough compact. For this reason, we will exploit again a BOW-like model with K_h centroids defined this time on a one-dimensional space. Indeed, since for each pixel of the tile, just one value is associated that is the gradient value, we will subdivide (quantize) the gradient scale into K_h centroids regularly spaced (Fig. 4.5). Once the centroids are defined, for each tile, the retrieval of the shape signature S_H is carried out in the same way as for the color signature S_c.

A third strategy (HRC-B) consists in exploiting the synergy between the color and shape signatures, through a simple fusion operation which will be described in Section 4.3.4.

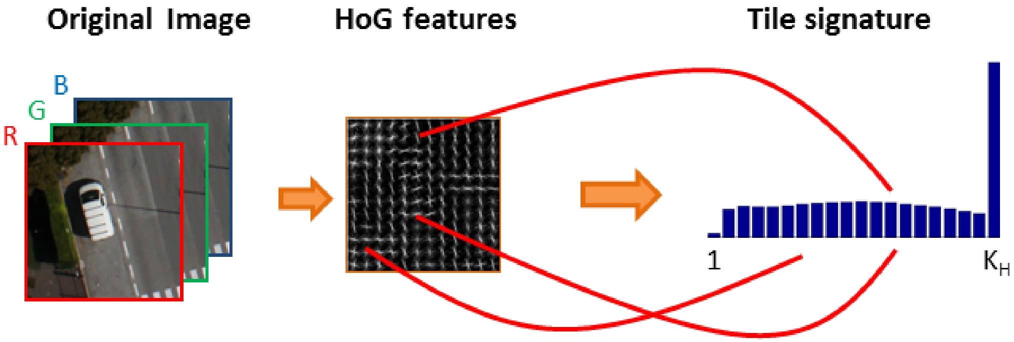


Figure 4.5. Tile signature extraction with the shape-based representation (HR-B strategy).

4.3.2. Point-Based Representation Strategies

Differently from the previous strategies, the following ones do not analyze each tile as a single entity but describe it according to its points of interest. This way to represent images has demonstrated to be very effective especially in the computer vision field where image resolution is typically extremely high. In particular, in this work, the scale invariant feature transform (SIFT) algorithm is investigated to extract and describe the points of interest within tiles. SIFT detects invariant points inside an image and describes them with descriptors of 128 features by exploiting a four-stage algorithm. In greater detail, from a scale space created by convolving the original image with a cascade of Gaussian functions (*i.e.*, difference of Gaussian DoG), a set of invariant points, that represent the maxima and the minima, is extracted. After a refinement process which eliminates the points poorly localized along the edges or with low contrast because considered unstable, the remaining points are described by a feature vector. The features are extracted from a 16×16 pixel mask created around each keypoint and oriented in accordance with the main local orientation. In the mask, the gradient orientations are represented in 8-bin histograms. To obtain vectors invariant to illumination, a normalization operation is performed. In the context of this work, each extracted tile will be represented by K_s points described by vectors of length 128. It is noteworthy that the number of points K_s is not fixed and varies from tile to tile depending on the information content. This strategy, which takes the acronym SR, associates a variable length and multidimensional (since each element is a vector of size 128) signature S_S.

A potential issue of the SR strategy is that it does not exploit color information but just local shape information. An alternative point-based strategy (SCR-B) consists in extracting and adding to the descriptor of each point belonging to S_S color information from its local context, *i.e.*, the 16×16 pixels window. In order to reduce the dimension of this extra color information, a BOW model is adopted. The number of words for representing the color information is also in this case K_c (as for the CR-B strategy). At the end, in the SCR-B representation, the tiles are described with a signature S_{SC} composed by the same points of interest as in SR but with larger feature vectors (concatenation of the 128 SIFT features and the K_c-bin color histogram).

4.4. Matching

Once all the tiles of an image I are extracted and represented, the next step consists in the comparison between the tiles of I and those in the training library to perform the multilabeling process. In particular, given a tile P of image I, it is associated with the binary descriptor of the tile with the most similar signature present in the training library. In this work, various similarity measures are compared. For what concerns the global representation strategies, the similarity is computed by exploiting histogram distance measures. In particular, the Euclidean distance (E) and the Chi-squared histogram distance (χ^2) are considered. The first distance measure is very popular. It however tends to magnify large bin differences and neglect small ones [84]. The χ^2 somewhat solves this issue by equalizing these differences. Given a test tile P and a training tile P_T, the Euclidean and the Chi-squared distances between their generic signatures S and S_T (these signatures can be any of those introduced in Section 4.3.1), both of length L, could be computed as follows, respectively:

$$\mathrm{E}(\mathrm{S},\mathrm{S_T}) = \sum_{\mathrm{i}=1}^{\mathrm{L}} \sqrt{(\mathrm{S(i)} - \mathrm{S_T(i)})^2} \tag{4.1}$$

$$\chi^2(\mathrm{S},\mathrm{S_T}) = \frac{1}{2}\sum_{\mathrm{i}=1}^{\mathrm{L}} \frac{(\mathrm{S(i)} - \mathrm{S_T(i)})^2}{\mathrm{S(i)} + \mathrm{S_T(i)}} \tag{4.2}$$

When the HCR-B strategy is adopted, a binary descriptor is assigned to the considered tile on the basis of the distances between, on the one side, its color (S_C) and shape (S_H) signatures and, on the other side, the color training S_{TC} and shape training S_{TH} signatures of the training tiles, respectively. In particular, we derive the following linearly combined distances:

$$\mathrm{E}^2(\mathrm{S},\mathrm{S_T}) = \alpha \times \mathrm{E}^2(\mathrm{S_C},\mathrm{S_{TC}}) + (1-\alpha) \times \mathrm{E}^2(\mathrm{S_H},\mathrm{S_{TH}}) \tag{4.3}$$

$$\chi^2(\mathrm{S},\mathrm{S_T}) = \alpha \times \chi^2(\mathrm{S_C},\mathrm{S_{TC}}) + (1-\alpha) \times \chi^2(\mathrm{S_H},\mathrm{S_{TH}}) \tag{4.4}$$

where α represents a parameter which controls the weight of the color and the shape information in the distance computation.

The matching approach used for the point-based representation strategies is slightly different. Indeed, since each tile is represented by a set of points of interest, the similarity between a given tile and a training tile is defined as the number of points that are similar in accordance to the matching definition proposed by Lowe [25]. In other words, a point p_1 is considered similar to p_2 if their Euclidean distance multiplied by a threshold *Th* is not greater than the Euclidean distances between p_1 (or p_2) and all the other points of interest. Thus, in SR and SCR-B strategies, to a given test tile, it is associated the binary descriptor of the training tile with the highest number of matching points.

At the end of the matching process, each tile is “coarsely” described by a subset of classes present in it and encoded in the binary vector of the closest training tile.

4.5. Experimental Results

4.5.1. Dataset Description and Experimental Setup

The images used in this work have been selected from a set of images acquired by using an UAV equipped with imaging sensors spanning the visible range. The images were taken over the Faculty of Science of the University of Trento (Italy) on the 3rd October 2011 at 12:00 am. Nadir acquisitions were performed with a picture camera Canon EOS 550D characterized by a CMOS APS-C sensor with 18 megapixels. The images have three channels (RGB) and a spatial resolution of approximately 2 cm. All the acquired images have sizes of 5184×3456 pixels with 8 bits of radiometric resolution. For the experimental validation, we used 10 images, subdivided into three groups:

a) *Training Group*. 2 images were selected to create the training library, which is composed by 6000 tiles randomly extracted from both images by cropping them in random positions and under angles chosen randomly among four values {0°, 90°, 180° and 270°}. The size of the tiles was fixed to 250×250 pixels for all experiments. Given the image resolution, such size makes that each tile covers an area of 5×5 meters, which represents a good compromise between detail of provided thematic information and processing time. The choice of the two training images was done so as to cover as most as possible the predefined classes of objects, which are common in urban scenarios. They are: ‘Asphalt’, ‘Grass’, ‘Tree’, ‘Vineyard’, ‘Pedestrian Crossing’, ‘Person’, ‘Car’, ‘Roof 1’, ‘Roof 2’, ‘Solar Panel’, ‘Building Facade’, ‘Soil’ and ‘Shadow’. In the area we investigated, the roofs have a too large variability to be grouped in just one class. We thus split the roof class in two classes, namely bright and dark roofs. The two training images were used to create a color codebook for BOW modeling. The parameter *Kc* (*i.e.*, number of color words) was fixed to 200 in order to satisfactorily represent the large color variations in EHR images. The value of *Kh*, which represents the quantization level of the HoG features, was set to 20. According to [54], HoG feature values are saturated to 0.2, involving in our case a distance of 0.01 between quantization levels.

b) *Validation Group*. Just one image belongs to this group. It was used to calibrate the values of α and Th exploited in the matching stage. A preliminary analysis on the validation image showed us that the best results are those obtained with α equal to 0.6 and *Th* equal to 2. A value of α equal to 0.6 means that color information is slightly favored over shape information (60% versus 40%, respectively) in the search for the best match.

c) *Test Group*. It is made up of 7 images. It is the group on which we assessed the accuracy of the proposed coarse description method. The images that belong to this group represent different classification scenarios. For instance, some images mainly represent agricultural areas, thus the dominant classes are grass, tree or vineyard, while other images are centered over urban areas and therefore the leading classes are roofs and asphalt.

To evaluate the accuracy of the proposed methodology, we computed two well-known accuracy measures, namely the sensitivity (*SENS*) and the specificity (*SPEC*) defined as follows:

$$SENS = \frac{TP}{TP + FN} \tag{4.5}$$

$$SPEC = \frac{TN}{TN + FP} \tag{4.6}$$

where *TP*, *FP*, *TN* and *FN* stand for ‘true positives’, ‘false positives’, ‘true negatives’ and ‘false negatives’, respectively. These values are computed by comparing the estimated binary descriptor assigned to each tile with the true ‘multilabel’ vector derived by tiling the ground-truth manually created.

4.5.2. Final Results

In the following paragraphs, the results of the proposed method are reported and discussed. In particular, we will first look at the global accuracies computed for all the combinations of tile representation and matching strategies on the seven test images. We will also present detailed results (accuracies of each single class and classification map) for two different test images. The first results will give us a general overview about the accuracy of the method whereas the second results will allow us to draw more specific evaluations for the single classes. All the experiments were conducted on an Intel Core i5-2400 CPU @ 3.10 GHz with 4 GB RAM, GPU Intel(R) HD Graphic Family and on a Matlab 2013b platform.

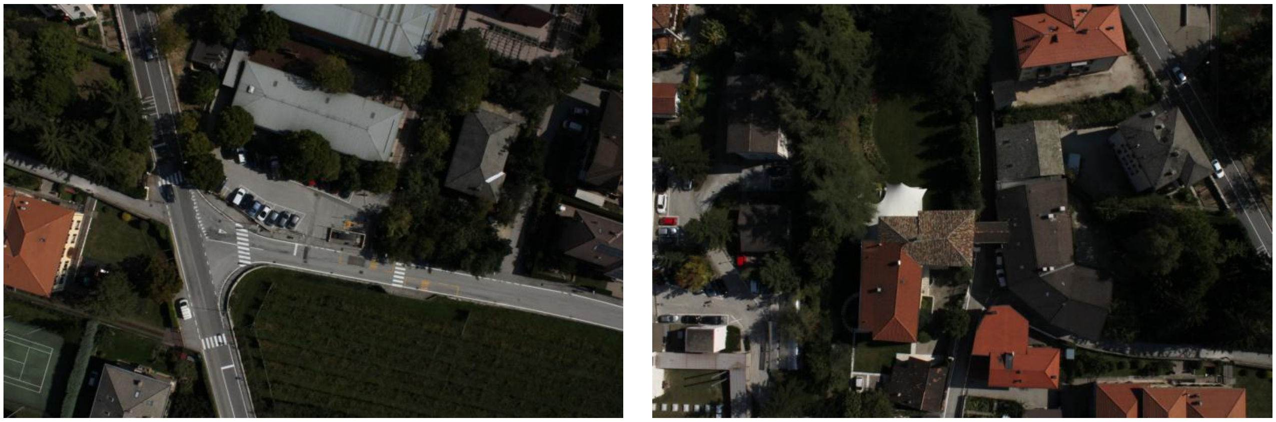
TABLE 4.II. OVERALL ACCURACIES IN TERMS OF SENSITIVITY (SENS) AND SPECIFICITY (SPEC) OBTAINED BY THE PROPOSED TECHNIQUES. FURTHERMORE, THE COMPUTATIONAL TIME PER IMAGE IS REPORTED FOR EACH STRATEGY.

	Global Representation						**Point-Based Representation**	
	Euclidean Distance			**χ^2 Distance**			**Euclidean Distance**	
	HR-B	**CR-B**	**HCR-B**	**HR-B**	**CR-B**	**HCR-B**	**SR**	**SCR-B**
SENS	48.2	59.66	59.9	48.5	61.41	65.37	34.74	37.5
SPEC	88.77	91.26	91	88.7	83.75	92.68	88.31	88.7
TIME (sec.)	22.44	44.32	73.66	48.35	66.99	121.20	6832.8	19685.25

By analyzing Table 2, one can notice that the best results are those which exploit the HCR-B representation approach and the χ^2 distance as similarity measure. Indeed, this combination of representation and matching strategies achieves 65.4% and 92.7% of sensitivity and specificity, respectively. If we consider that the images taken into consideration are 7 and the classes of objects that we would like to identify are 13, the results are encouraging. The training library and the codebooks have been built by exploiting just two training images. Consequently, it is almost impossible to cover all possible aspect variations that the objects belonging to the 13 classes could assume. If we compare the results obtained by this representation strategy coupled with both matching measures, it is possible to see an increment (of around 5%) for the sensitivity achieved with the χ^2 distance with respect to that obtained with the Euclidean distance. On the other hand, the specificity improves but in a less marked way. This suggests that the χ^2 distance is a more appropriate measure thanks to the normalization mechanism it embeds. Another point worth to highlight is the difference between the accuracies obtained with CR-B and those achieved with HR-B. It emerges that separately color works better than shape for analyzing UAV images. Independently from the matching strategy, CR-B allows to obtain values of sensitivity around 60% while those of HR-B are around 48%. The large variability of the shapes of the numerous objects analyzed in our experiments renders shape representation less discriminative than color representation. However, as seen above, their combination makes it particularly interesting because of their synergic properties.

When a point-based representation strategy is adopted, the results are not satisfactory. For both explored strategies, without and with the integration of color information, the final sensitivity accuracies do not exceed 37%. Despite such a poor performance, we can draw interesting observations. First, the combination of the original SIFT descriptor with color information allows increasing the performance of about 3%, confirming what observed for the global representation of tiles, that is the combination of color and shape is advantageous in UAV imagery. Second, global tile representation works better than point-based representation both in terms of accuracy and computation time. The reason can be found in the fact that point-based representation can be useful to detect a particular class of objects. When moving to the simultaneous discrimination of numerous classes of objects (like in these experiments), the variability of the descriptors becomes so high that it calls for more sophisticated classification methods making the simple matching strategies explored in this work not suitable for point-based representations.

In order to enrich our experimental analysis, we report a detailed assessment on two different test images (Fig. 4.6). In particular, sensitivity and specificity for single classes (Tables 4.III-4.IV) as well as classification maps (Fig. 4.7-4.8) are provided. A first aspect, which could be stressed, is the level of details that "coarse" thematic maps provide. Indeed, such maps do not report the exact spatial position of each class but just a rough spatial distribution of the classes in the considered UAV image. It is interesting to observe how similar classes, such as tree and vineyard or roof1 and roof2, are correctly discriminated. Despite the fact that a “coarse” description of the tiles has been performed, the areas in which the classes are present are in general satisfactorily identified.



(a) (b)

Figure 4.6. Example of two test images used in this work

4.III-4.IV confirm the general conclusions drawn before, namely HCR-B representation and χ^2 distance measure work better than the other combinations. In greater detail, the accuracies are good for classes which present small intra-class variability such as ‘asphalt’, ‘grass’, ‘tree’, ‘roofs’, ‘vineyard’, ‘ground’ and ‘shadows’, while for the other classes the accuracy is not high. The reasons behind the poor performance regarding these classes could be ascribed to two main factors: 1) ‘pedestrian crossing’, ‘car’, ‘people’, ‘solar panel’ and ‘building facades’ are all classes composed by objects with very different aspects incurring in a large intra-class variability, 2) some objects belonging to these classes have too small size with respect to the dimension of the tile to contribute in the creation of the signature. The reduction of the tile dimension could be a suitable solution to identify classes composed by small objects but this would decrease the detection accuracy of other classes. A small tile can have not enough information to discriminate between big objects. Moreover, small tiles increase the processing time.

The fact that for all combinations of tile representation and matching strategies and for all considered classes the specificity is higher than the sensitivity deserves to be explained. This situation is reasonable because a priori the probability to get true negatives is higher than the probability to have true positives.

Comparing the processing time per image (Table 4.II), we can note three main aspects that affects the performance: 1) distance measure; 2) signature length; and 3) number of signatures per tile. Indeed, among all the strategies, the fastest one is the HR-B strategy combined with the Euclidean distance while the slower one is the SCR-B strategy. Indeed, HR-B describes the tiles with just 20 features, while in SCR-B the tiles are represented by tens of points of interest and each of them is described with a feature vector of length 328 features (128 related to the SIFT features plus 200 for color information). The HCR-B, which is the best strategy in terms of final accuracies, takes about 2 minutes per image to complete its “coarse” description. It is about 100 seconds slower than the fastest strategy but at the same time increases the performance of about 17%.

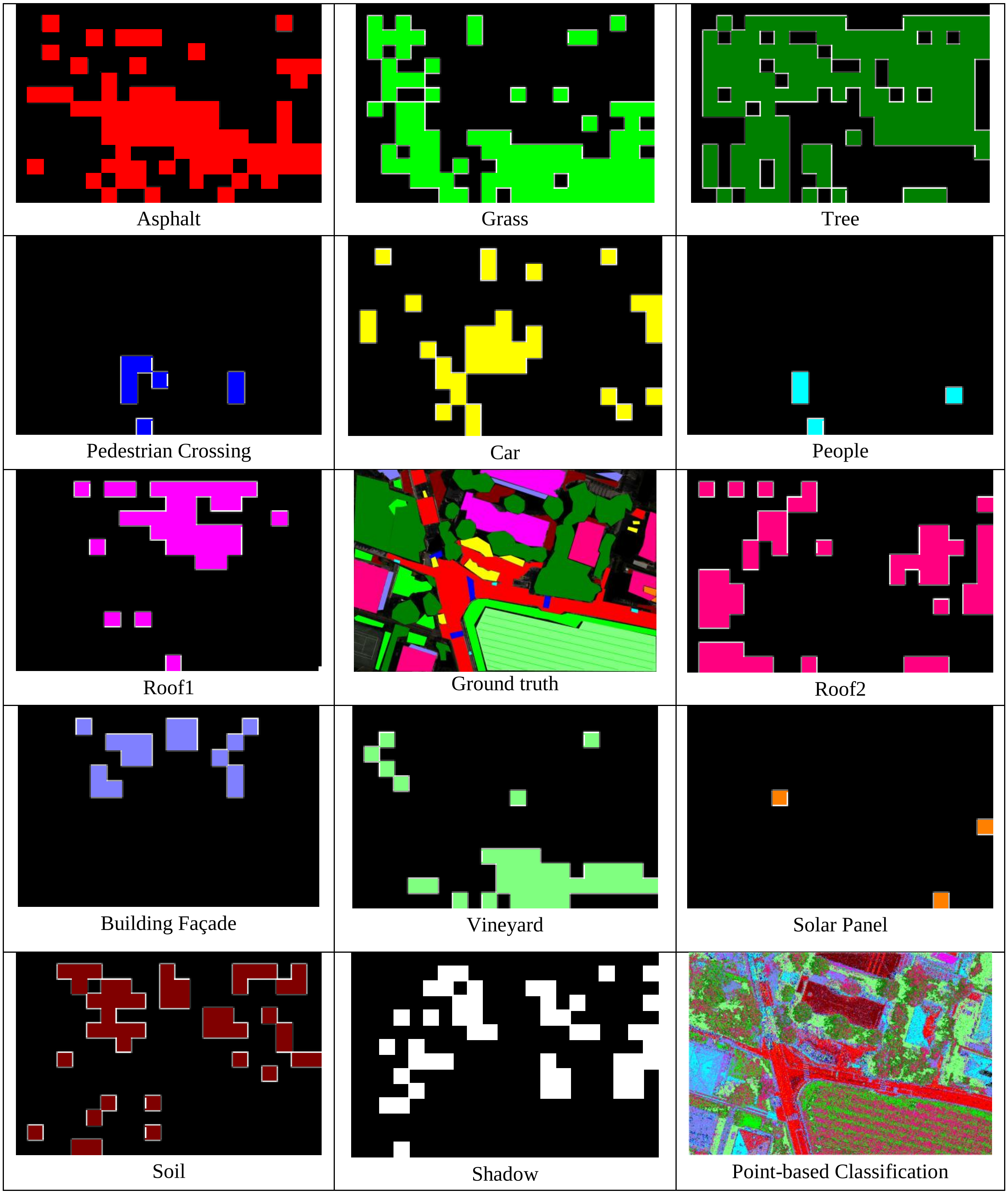


Figure 4.7. Thematic maps of all the investigated classes obtained by HCR-B technique on a first test image.

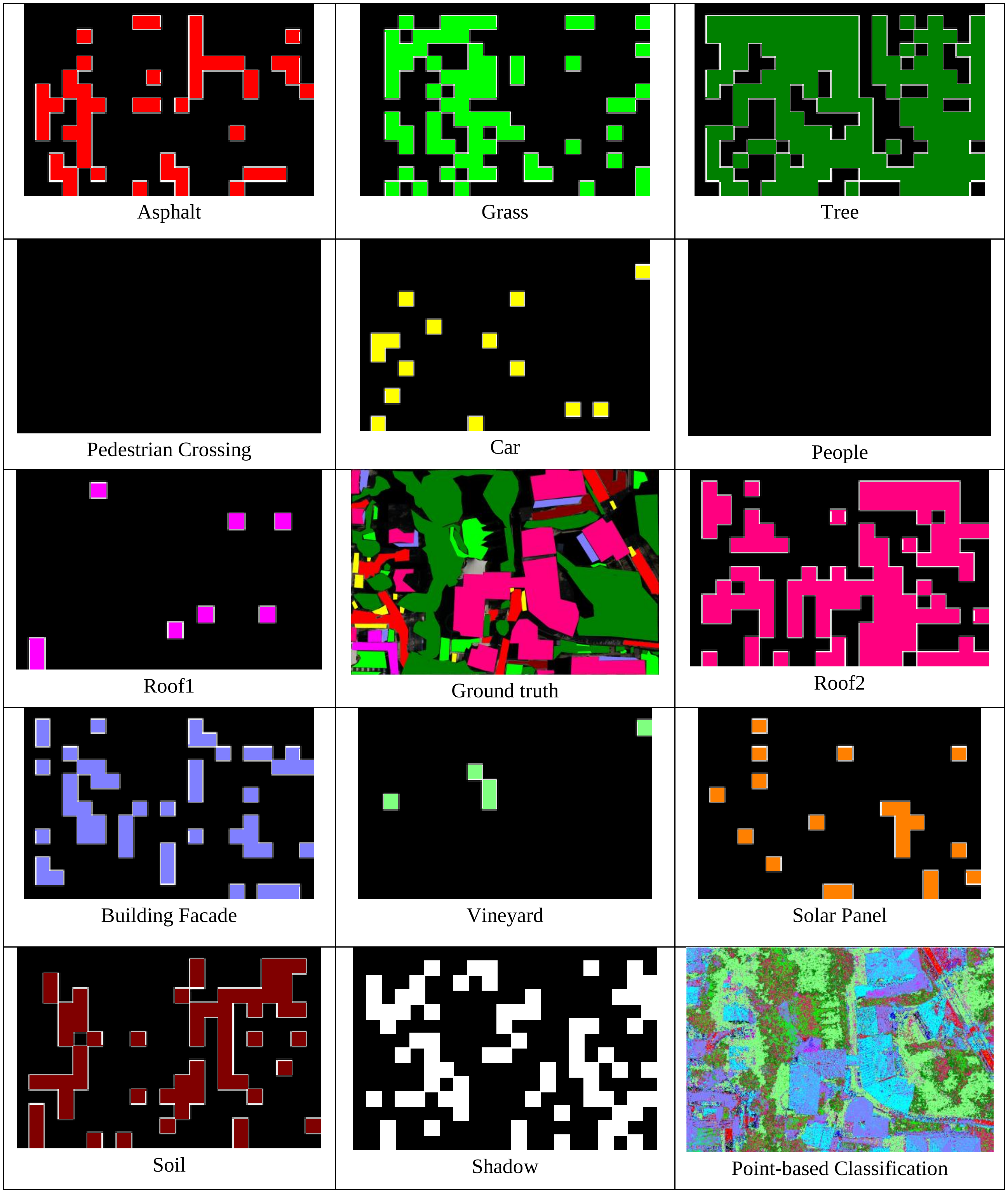


Figure 4.8. Thematic maps of all the investigated classes obtained by HCR-B technique on a second test image.

TABLE 4.III. SENSITIVITY (SENS) AND SPECIFICITY (SPEC) ACCURACIES FOR ALL THE CLASSES ACHIEVED BY THE PROPOSED TECHNIQUES ON A FIRST TEST IMAGE.

				Asphalt	**Grass**	**Tree**	**Pede. Crossin g**	**Car**	**People**	**Roof1**	**Roof2**	**Building Facade**	**Vineyard**	**Solar Panels**	**Soil**	**Shadow**	**TOT**
Global R.	**Euclidean D.**	**HR-B**	**SENS**	35.5	47.0	71.5	16.7	18.5	0.0	48.5	45.5	9.1	33.3	0.0	38.5	55.9	45.9
			SPEC	85.9	57.6	61.5	97.2	91.0	**100.**	92.1	80.2	86.6	90.0	98.8	77.8	82.7	87.5
		CR-B	**SENS**	76.3	62.7	**90.0**	**58.3**	59.3	**16.7**	81.8	84.9	22.7	71.8	50.0	56.4	52.9	70.9
			SPEC	90.2	**83.1**	55.4	98.0	92.3	98.0	95.6	88.1	94.1	**97.3**	**99.6**	87.8	87.2	92.0
		HCR-B	**SENS**	73.7	**67.5**	86.9	33.3	51.9	**16.7**	78.8	75.8	27.3	82.1	**100**	59.0	44.1	69.6
			SPEC	89.1	80.8	60.0	98.4	91.4	96.9	96.0	**90.3**	**95.8**	96.4	98.8	91.0	**88.9**	92.4
	χ2 D.	**HR-B**	**SENS**	39.5	50.6	72.3	33.3	22.2	0.0	51.5	45.5	13.6	35.9	0.0	46.2	47.1	48.3
			SPEC	86.4	59.3	64.6	96.4	92.7	99.6	93.4	78.9	85.7	94.1	97.7	80.5	84.1	88.2
		CR-B	**SENS**	76.3	**67.5**	83.1	50.0	59.3	16.7	**87.9**	**87.9**	31.8	**89.7**	50.0	53.9	55.9	**72.0**
			SPEC	92.4	81.9	67.7	97.2	94.9	97.6	98.7	**90.3**	94.1	97.7	**99.6**	90.5	86.7	92.4
		HCR-B	**SENS**	**77.6**	63.9	83.8	41.7	**63.0**	**16.7**	75.8	81.8	**36.4**	**89.7**	100.	**61.5**	50.0	71.3
			SPEC	91.3	80.8	70.8	98.0	94.0	98.4	**98.2**	**90.3**	95.0	95.9	**99.6**	**91.4**	88.5	**93.4**
Point-Based R.	**Euclidean D.**	**SR**	**SENS**	43.4	50.6	56.9	33.3	37.0	0.0	36.4	84.9	18.2	30.8	50.0	41.0	**58.8**	47.8
			SPEC	92.4	68.9	**76.2**	**99.6**	94.9	98.4	97.4	70.9	95.0	95.0	98.8	78.7	70.4	89.4
		SCR-B	**SENS**	42.1	59.0	58.5	33.3	40.7	0.0	36.4	75.8	13.6	46.2	50.0	46.2	58.8	50.2
			SPEC	**94.6**	66.7	70.8	**99.6**	**95.3**	98.4	96.9	72.7	94.5	95.9	99.2	81.5	72.1	89.7

TABLE 4.IV. SENSITIVITY (SENS) AND SPECIFICITY (SPEC) ACCURACIES FOR ALL THE CLASSES ACHIEVED BY THE PROPOSED TECHNIQUES ON A SECOND TEST IMAGE. THE SENSITIVITY VALUES MARKED WITH "-" INDICATE THAT THE CLASS IS NOT PRESENT IN THE IMAGE.

				Asphalt	**Grass**	**Tree**	**Pede. Crossing**	**Car**	**People**	**Roof1**	**Roof2**	**Building Facade**	**Vineyard**	**Solar Panels**	**Soil**	**Shadow**	**TOT**
Global R.	**Euclidean D.**	**HR-B**	**SENS**	28.6	43.9	66.9	-	29.2	--	**20.0**	38.2	10.0	-	-	21.1	37.0	43.1
			SPEC	86.3	60.3	46.0	98.8	92.8	99.6	94.4	76.0	88.3	89.6	96.9	80.1	81.9	87.7
		CR-B	**SENS**	30.6	41.5	66.3	-	**50.0**	-	**20.0**	41.8	13.3	-	-	21.1	34.0	44.2
			SPEC	85.8	63.5	44.0	97.7	92.4	99.6	94.8	80.7	89.6	90.8	95.4	74.3	84.4	87.7
		HCR-B	**SENS**	24.5	48.8	77.5	-	20.8	-	10.0	**72.7**	**40.0**	-	-	**68.4**	35.0	55.6
			SPEC	85.3	80.8	42.0	**100**	96.6	**100**	96.4	74.0	80.4	**97.7**	95.0	73.4	81.9	88.9
	χ2 D.	**HR-B**	**SENS**	**32.7**	61.0	78.1	-	25.0	-	10.0	61.8	36.7	-	-	63.2	41.0	56.2
			SPEC	88.6	**81.3**	46.0	**100**	**98.7**	**100**	95.2	**84.0**	87.8	**97.7**	94.6	78.0	85.0	90.9
		CR-B	**SENS**	28.6	48.8	77.5	-	29.2	-	**20.0**	69.1	33.3	-	-	63.2	41.0	56.4
			SPEC	82.5	75.8	45.0	**100**	95.8	**100**	98.4	74.0	85.2	97.3	92.7	75.9	80.6	88.8
		HCR-B	**SENS**	30.6	65.9	**81.9**	-	25.0	-	**20.0**	61.8	26.7	-	-	57.9	46.0	**57.8**
			SPEC	87.7	80.4	53.0	**100**	98.3	**100**	**99.2**	81.3	90.9	97.3	93.1	82.2	**85.6**	**91.5**
Point-Based R.	**Euclidean D.**	**SR**	**SENS**	4.1	53.7	40.6	-	4.2	-	**20.0**	52.7	6.7	-	-	5.3	**54.0**	38.1
			SPEC	93.4	52.5	**70.0**	99.6	97.5	99.6	96.8	60.0	95.2	96.9	98.1	82.6	50.0	88.1
		SCR-B	**SENS**	4.1	**68.3**	42.5	-	4.2	-	10.0	44.6	0.0	-	-	10.5	42.0	35.5
			SPEC	**96.2**	45.7	64.0	**100**	97.5	**100**	94.8	65.3	**97.0**	96.9	**98.9**	**86.3**	44.4	87.9

4.6. Comparison with a pixel-based classifier

In order to complete our experimental assessment, we compared our method with a pixel-based classifier which exploits a SVM classifier. SVMs are intrinsically binary classifiers but they can be adapted to multiclass problems by adopting binary decomposition tricks such as the one-against-one method [42]- [43]. Shortly, a binary SVM begins by assuming a training set defined in a N-dimensional feature space where each sample z (in this work, each training sample is described by its three color features) is properly labeled {-1, +1}. During its training, a SVM classifier finds the hyperplane characterized by a weight vector w and a bias b in a kernel-induced feature space $(\Phi(z) \in R^{N'}, N' > N)$ which best separates the two considered classes. From this hyperplane, it is possible to derive a discriminant function $f(z)$ useful to take the class assignment decision:

$$f(z) = w^{*}\emptyset(z) + b^{*} \tag{4.7}$$

In our experiments, for the training of the SVM classifier, 400 samples per class were randomly extracted from the same training images used before. Hence, in total, 5200 training samples were used. From Fig. 4.7-4.8, it is possible to see how a pixel-based classifier ran on EHR images generates various classification problems though, as expected, the whole image structure is better maintained with respect to the coarse classification. The discrimination between objects with similar colors is extremely difficult. For instance, pixels which belong to dark vehicles appear very similar to those belonging to the shadow class. One can also notice the confusion in the discrimination between white cars and pedestrian crossroads. Indeed, all white cars are labeled with the same value of pedestrian crossroad class. The final accuracies for the two test images are 17.4% and 11.2%, respectively. These last results are very poor and confirm that pixel-based classifiers are not suited to EHR images acquired with UAVs because they consider separately each single pixel instead of exploiting contextual information lying not only in the direct vicinity but also in the far neighborhood.

4.7. Conclusions

In this work, an innovative method to "coarsely" describe UAV images has been presented. UAV images convey a huge quantity of information, which makes the single pixel and even its direct vicinity not sufficiently expressive for distinguishing between numerous classes of objects. Such images to be adequately analyzed require to exploit information in a large context. For this reason, we propose a method which first subdivides the input image in tiles, whose size depends on expected size of objects and image resolution, and then assigns to each tile a subset of classes present in it. The assignment requires two ingredients, *i.e.,* tile representation and matching. Several strategies for tile representation are explored and discussed. They are divided in two groups depending on the way they create the signatures to assign to the tiles. The global representation strategies consider the tile as a whole and create the signatures on the basis of color or shape information, whereas point-based representation strategies describe the tiles with sets of points of interest, each with its own descriptor. The matching aims at comparing each tile of the considered image with a training library and assigning it to the closest training tile. In this work, the simple Euclidian and χ^2 distances have been explored.

The experimental results unveil that, among the five proposed tile representation strategies, HCR-B (combining both color and shape representation under a BOW model) generates the best results in terms of accuracy. Regarding the matching strategy, the χ^2 distance allows obtaining a better discrimination capability than the Euclidean distance. The computation times are short, in the order of few minutes per image. It is also shown that traditional classification is not suited for UAV imagery both in terms of accuracy and computation time. The classification maps in Figures 4.7-4.8 illustrate well the outcome of the coarse image classification, that is a coarse spatial distribution of each class within the image. It is noteworthy that the smaller the size of the tiles, the finer the spatial distribution, but the larger the computation time and the smaller the class discrimination capability. Coarse classification maps can be useful for different purposes such as image archiving and fast screening operations for feeding customized object detection algorithms.

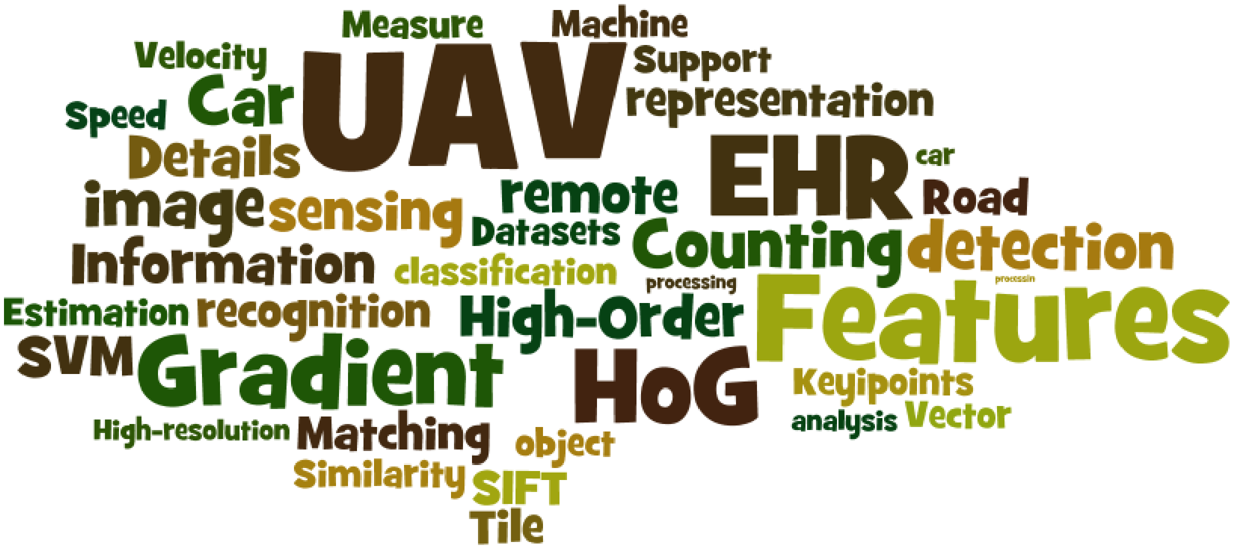
This new image analysis method opens the way to future developments along different directions, in particular toward exploring more sophisticated tile representation and matching strategies. In this work, point-based representation strategies proved unsatisfactory for multiclass coarse image analysis. A different tile representation with these descriptors based for instance on syntactic pattern recognition could be an interesting path to follow.

5. Acknowledgments

The Authors would like to thank C.-C. Chang and C.-J. Lin for supplying the software LIBSVM [85], A. Vedaldi and B. Fulkerson for making available the software VLFeat [86] and C. E. Rasmussen, K. I. Williams and H. Nickisch for supplying the GP toolbox used in the context of this paper [87].

6. Conclusions

***Abstract** – The aims of this chapter are to report the methodological and experimental developments achieved by the present thesis, to draw the conclusions and to describe possible future developments. The reader is referred to the previous single chapters for more detailed discussions about the different proposed methods.*



In this thesis, various detection and analysis methods for Unmanned Aerial Vehicle images have been investigated. UAV images are characterized by extremely high spatial resolution which makes the analysis particularly challenging. Since the objects are described by thousands of pixels, similar targets may assume different aspects. The definition of a model which considers the large intra-class variability and allows the detection of a particular class of objects or the multi-class analysis of the considered image is not easy. In the context of this thesis, we proposed different solutions for the detection of specific classes of objects and for the description of images representing urban areas. In the following, we will summarize the proposed detection and analysis strategies, the related experimental results and conclusions. We refer the reader to the single chapters for more details.

In Chapter 2, (*Traffic Monitoring Strategies in UAV images)*, the problems concerning the detection of vehicles and the determination of their speeds were faced. In particular, in the first part of this chapter, we described two methods for the detection and the counting of cars in urban areas, whereas in the second part we exposed an approach which detects the moving vehicles and estimates their speed. The high level of detail which is typical of UAV images calls for appropriate image representation approaches that highlight the aspect of the objects. The first car detection strategy transforms the investigated image into a set of keypoints properly described and aims at detecting the presence of cars starting from such image representation. The use of points of interest described by features related to the local properties of the objects and which are invariant to image scale, rotation and translation has demonstrated to be adapted to the characterization of UAV images. This strategy, before performing the extraction of the SIFT keypoints, limits the space to investigate to the asphalted areas in order to reduce the computational time and to obtain more accurate results. Then, to increase the discriminative power, the original SIFT descriptors are integrated with further spectral and morphological information. The presence of vehicles is estimated by a properly trained classifier which is able to distinguish the keypoints corresponding to the car class from all the others. Due to EHR images, the number of keypoints extracted is massive and consequently it may happen that a single vehicle is identified by tens of keypoints. To solve this problem and to achieve a univocal identification of the cars and to allow the counting, a keypoint-merging step based on spatial clustering concludes the procedure. The second car detection approach is based on a sliding window filtering process and assigns to each spatial position HOG features. These features are known to be appropriate for the description of the shape of the objects. Also this second strategy, before performing the feature extraction step, undergoes a screening stage to isolate the asphalted areas. This approach performs the detection of the vehicle by comparing the features extracted for each spatial position with those of a previously built catalogue of cars exploiting different similarity measures. As for the first strategy, a merging step has been implemented to allow the counting of cars. Differently from the previous approach, this methodology in addition to the number of detected car provides information related to the spatial position and to the orientation of each vehicles. The results obtained by the two car detection strategies are encouraging; indeed the overall accuracies are 76.61% and 80.94% for the first and for the second approach, respectively. The proposed techniques, despite the promising accuracy, have to process a huge amount of pixels which makes them not suitable for real applications because of the computational time. Therefore possible improvements may focus on the development of fast feature extraction and description techniques. The second part of this chapter describes an approach to detect moving vehicles and compute their speed. The idea which leads to this strategy is that two images, acquired by means of UAV over the same area and with a small time lag, are very similar and, if they are compared, the changes are only due to the moving objects. After an automatic image registration step, a refinement stage isolates the moving objects in the sequence of images and then a matching stage deduces the shifts of position of the vehicles. Also in this case the EHR of the images requires a merging step in order to obtain univocal matches of the changes (vehicles). In the end, thanks to the knowledge of the resolution of the image and of the spatial shift of each moving object, the estimation of the speeds is derived. The proposed monitoring strategy has been evaluated on two different datasets and provides interesting results. The accuracy of the registration and matching steps affects the final performance of the proposed monitoring strategy; therefore, the developments of more robust registration and matching stages could improve the final performances and could allow to deal with more complex scenarios.

In Chapter 3, (*Filter-Based Object Detector for UAV Images)*, a new methodology to detect classes of objects was presented. The technique relies on high-order gradient features and on a Gaussian Process (GP) regression model. High-order gradient features aim at extracting more detailed geometric information than traditional image gradient features improving the discrimination power. The amount of information provided by high-order gradient features may make its managements very challenging and time consuming. For that reason, in this work we developed a filter based on a GP regression model. Typically, GP regression models provide results in short time and are able to tune its free parameters in an automatic way. These characteristics make GP regression models very desirable in our context. The proposed method was tested for the detection of two specific classes of objects very common in urban areas, namely cars and solar panels. The achieved results are very intriguing and allow deducing interesting conclusions. High-order gradient features better describe the structure of the object. Indeed, final accuracies strongly increase by passing from the first to the fourth order of gradient. Moreover, the computational times justify positively the use of GP model. Despite the good results, the proposed technique presents some drawbacks. In particular, the poor precision in the singular detection of the objects and the high sensibility to changes of resolution of the image are two problems that need to be taken into consideration.

Chapter 4, (*Multiclass Tile-based Analysis Method for UAV Imagery)*, describes a novel method to "coarsely" describe UAV images. The proposed approach subdivides the original image in small tiles, and then suggests which of the possible classes are present in each tile. UAV images characterized by EHR cannot be completely classified with traditional pixel-based approaches because pixels considered separately from the global context are not informative, or with object-based approaches because it is extremely difficult to find a model simultaneously adaptable to numerous classes of objects. In this work, several tile representation techniques and distance measures have been explored and compared. From the experimental results, it is possible to see that the HCR-B (HoG-Color representation under a BoW model) and χ^2 as representation strategy and distance measure, respectively, are those that allow to achieve the best results. This innovative work could be the starting point to future developments such as the identification of other tile representations and distance measures.

The contributions provided in this thesis have been focused on the development of novel detection and analysis methods for UAV images. The final results are in general promising, encourage the development of future improvements and could open the ways to new trends. For instance, the multiclass description approach proposed in Chapter 4 has room for improvements. In particular, new image representation and matching strategies could be investigated to make the approach faster and more robust. In the proposed method, the detection of some classes (*e.g.,* people, car) is more complicated that the detection of other classes, therefore the use of image representation and matching strategies that enhance the performance of all classes is desirable. Furthermore, instead of using fixed-size tiles, one could consider the use of tiles adapted to the shape of the objects. In this sense, the exploitation of a segmentation approach to extract the tiles could be useful. This approach could provide additional information to discriminate the presence of the several classes.

A common problem of most of the proposed works regards the computational time needed to compare the samples extracted from the images with those belonging to a dictionary. The dictionaries, due to the resolution of the images, have to include hundreds of samples to cover all possible aspect variations of the objects slowing down significantly the execution times. A potential solution could be envisioned in the dimensionality reduction of the dictionary without losing information content by for instance exploiting compressive sensing-based strategies. These last are able to recover an unknown sparse signal from a small set of linear projections. In this way, instead of working with dictionaries that contain thousands of samples one could exploit a small set of basic elements.

As said in the previous sections, the use of UAVs for civilian applications is continuously growing. In this context, it would be particularly interesting to develop techniques able to monitor the changes generated by urban growth, deforestation or by natural disasters (*e.g.,* earthquakes). The timely and extremely detailed capacity of acquiring information of UAVs is suitable to these purposes. EHR images may be a great advantage as they would allow a detailed description of the scenes, but, at the same time, they would need appropriate image analysis techniques to exploit and adapt all the potential to these aims. Instead of considering standard change detection pixel-based approaches, that may result inaccurate due to the high resolution of the images and extremely time consuming, one could exploit points of interest change detection techniques. For instance, by extracting two sets of points of interest (*e.g.,* SIFT points) from the considered couple of images and by matching such points one could infer an estimation about the percentage of changes. Furthermore, the isolation of the points that do not match could provide information about the areas of changes.

Moreover, a wide range of agriculture applications could be faced thanks to the use of hyperspectral or thermal sensors mounted on UAV platforms. These sensors acquire information that is extremely useful to assess the health of vegetation, but it needs to be properly managed. For instance, the field of view of common thermal sensors is very small, therefore to monitor wide areas it is necessary the acquisition of tens of images. To have a complete overview of the investigated areas, it would be essential the development of suitable image mosaicking techniques. Hyperspectral cameras collect information across many bands of the electromagnetic spectrum, besides having a very high spatial resolution we could have also a very high spectral resolution. The amount of information would grow exponentially but it would allow obtaining an extremely detailed description of the objects. In this context, the development of classification techniques which consider both spectral and spatial information would be fascinating. To face the huge amount of information, the use of appropriate feature selection techniques which reduce the spectral information but at the same time allow enhancing the differences between the several classes of objects could be useful. As said before, in hyperspectal images to each pixel a very large signature, which describe the spectral properties, is associated. Due to the high resolution of the images, it is likely that similar pixels present small differences in their respective spectral signatures. Therefore, the identification of the sets of features (*i.e.,* spectral bands) which highlights the spectral distinctive traits of a specific class of objects could be suitable. Alternatively, the use of points of interest which consider both the spatial and the spectral information in the extraction and in the description stages may reduce the amount of data keeping almost unvaried the informative content.

7. References

[1] T. Lillesand, R. Kiefer and J. Chipman, Remote sensing and image interpretation, John Wiley & Sons Ltd, 2004.

[2] R. A. and Schowengerdt, Remote sensing: Models and methods for image processing, Academic Press, 2006.

[3] M. A. Wulder, J. C. White, S. N. Goward, J. G. Masek, J. R. Irons, M. Herold, W. B. Cohen, T. R. Loveland and C. E. Woodcock, "Landsat continuity: Issues and opportunities for land cover monitoring.," *Remote Sensing of Environment,* vol. 112, no. 3, pp. 955-969, 2008.

[4] B. S. Studies., Landsat and Beyond: Sustaining and Enhancing the Nation's Land Imaging Program, National Academies Press., 2013.

[5] L. Lorenzi, F. Melgani, G. Mercier and Y. & Bazi, "Assessing the Reconstructability of Shadow Areas in VHR Images," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 51, no. 5, pp. 2863-2873, 2013.

[6] C. Zhang and J. M. Kovacs, "The application of small unmanned aerial systems for precision agriculture: a review," *Precision agriculture,* vol. 13, no. 6, pp. 693-712, 2012.

[7] P. J. Zarco-Tejada, V. González-Dugo and J. Berni, "Fluorescence, temperature and narrow-band indices acquired from a UAV platform for water stress detection using a micro-hyperspectral imager and a thermal camera," *Remote Sensing of Environment,* vol. 117, pp. 322-337, 2012.

[8] V. Gonzalez-Dugo, P. Zarco-Tejada, E. Nicolás, P. Nortes, J. Alarcón, D. Intrigliolo and E. Fereres, "Using high resolution UAV thermal imagery to assess the variability in the water status of five fruit tree species within commercial orchard," *Precision Agriculture,* vol. 14, no. 6, pp. 660-678, 2013.

[9] F. G. Costa, J. Ueyama, T. Braun, G. Pessin, F. S. Osório and P. A. Vargas, "The use of unmanned aerial vehicles and wireless sensor network in agricultural applications," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Munich, 2012.

[10] C. A. P. Delacourt, M. Jaud, P. Grandjean, A. Deschamps, J. Ammann, V. Cuq and S. Suanez, "DRELIO: An unmanned helicopter for imaging coastal areas," *Journal of Coastal research ,* pp. 1489-1493, 2009.

[11] M. Israel, "A UAV-based roe deer fawn detection system," in *Proceedings of the International Conference on Unmanned Aerial Vehicle in Geomatics (UAV-g)*, Zurich, 2011.

[12] D. W. Casbeer, R. W. Beard, T. W. McLain, S. M. Li and R. K. Mehra, "Forest fire monitoring with multiple small UAVs.," in *IEEE Procededings in American Control Conference* , Portland, 2005.

[13] K. Choi, I. Lee, J. Hong, T. Oh and S. W. Shin, "Developing a UAV-based rapid mapping system for emergency response," *SPIE Defense, Security, and Sensing. International Society for Optics and Photonics,* pp. 733209-733209-12, 2009.

[14] M. Andriluka, P. Schnitzspan, J. Meyer, S. Kohlbrecher, K. Petersen, O. Von Stryk, S. Roth and B. Schiele, "Vision based victim detection from unmanned aerial vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, 2010.

[15] A. Gaszczak, T. P. Breckon and J. Han, "Real-time peopl eand vehicle detection from UAV imagery," *IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics,* 2011.

[16] F. Rinaudo, F. Chiabrando, A. Lingua and A. Spanó, "Archaeological site monitoring: UAV photogrammetry can be an answer," in *he International archives of the photogrammetry, Remote sensing and spatial information sciences*, Melbourne, 2012.

[17] R. Saleri, M. Perrot-Deseilligny, E. Bardiere, V. Cappellini, N. Nony, L. De Luca and M. Campi, "UAV Photogrammetry for archaeological survey: The Theaters area of Pompeii," in *Digital Heritage International Congress*, Marseille, 2013.

[18] H. Z. L. Eisenbeiss, "Comparison of DSMs generated from mini UAV imagery and terrestrial laser scanner in a cultural heritage application.," in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVI-5*, 2006.

[19] T. Blaschke and J. Strobl, "What's wrong with pixels? Some recent developments interfacing remote sensing and GIS," *GeoBIT/GIS,* vol. 6, no. 01, pp. 12-17, 2001.

[20] T. Blaschke, "Object based image analysis for remote sensing," *ISPRS journal of photogrammetry and remote sensing,* vol. 65, no. 1, pp. 2-16, 2010.

[21] A. S. Laliberte and A. Rango, "Texture and scale in object-based analysis of subdecimeter resolution unmanned aerial vehicle (UAV) imagery.," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 47, no. 3, pp. 761-770, 2009.

[22] R. Klette, Concise Computer Vision., Springer London, 2014.

[23] T. Morris, Computer Vision and Image Processing, Palgrave Macmillan, 2003.

[24] J. Shia, J. Wang and Y. Xu, "Object-based change detection using georeferenced UAV images," in *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2011.

[25] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision,* pp. 91-110, 2004.

[26] X. Li and N. Aouf, "SIFT and SURF feature analysis in visible and infrared imaging for UAVs," in *IEEE International Conference on Cybernetic Intelligent Systems (CIS)*, 2012.

[27] H. Bay, T. Tuytelaars and L. Van Gool, "Surf: Speeded Up Robust Features," *Computer Vision-ECCV,* pp. 404-417, 2005.

[28] J. Leitloff, S. Hinz and U. Stilla, "Vehicle Detection in Very High Resolution Satellite Images of City Areas," *IEEE Trans. on Geoscience and Remote Sensing,* vol. 48, no. 7, pp. 2795-2806, 2010.

[29] B. Salehi, Y. Zhang and M. Zhong, "Automatic Moving Vehicles Information Extraction from Single-Pass Worldview-2 Imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 5, no. 1, pp. 135-145, 2012.

[30] W. Yao and U. Stilla, "Comparison of Two Methods for Vehicle Extraction From Airborne Lidar Data Toward Motion Analysis," *IEEE Geoscience and Remote Sensing Letters,* vol. 8, no. 4, pp. 607-611, 2011.

[31] S. Hinz, "Detection and Counting of Cars in Aerial Images," *IEEE International Conference on Image Processing,* pp. 997-1000, 2003.

[32] T. Zhao and R. Nevatia, "Car Detection in Low Resolution Aerial Images," *IEEE International Conference on Computer Vision,* pp. 710-717, 2001.

[33] H. Moon, A. Rosenfeld and R. Chellapa, "Performance Analysis of a Simple Vehicle Detection Algorithm," *Image and Vision Computing,* vol. 20, no. 1, pp. 1-13, 2002.

[34] W. Shao, W. Yang, G. Liu and L. J., "Car Detection from High-Resolution Aerial Imagery Using Multiple Features," in *In Geoscience and Remote Sensing Symposium (IGARSS)*, 2012.

[35] H. Grabner, T. T. Nguyen, B. Gruber and H. & Bischof, "On-line boosting-based car detection from aerial images," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 3, pp. 382-396, 2006.

[36] S. Kluckner, G. Pacher, H. Grabner and H. B. J. Bishof, "A 3D Teacher for Car Detection in Aerial Images," in *ICCV*, Rio de Janeiro, Brazil, 2007.

[37] S. Tuermer, F. Kurz, P. Reinartz and U. Stilla, "Airborne Vehicle Detection in Dense Urban Areas Using HoG Features and Disparity Maps," *IEEE journal of Selected Topics in Applied Earth Observation and Remote Sensing,* vol. 6, no. 6, pp. 2327-2337, 2013.

[38] J. Gleason, A. Nefian, X. Bouyssounousse, T. Fong and G. Bebis, "Vehicle Detection from Aerial Imagery," *IEEE International Conference on Robotic and Automation,* 2011.

[39] F. Yamazaki, W. Liu and T. T. Vu, "Vehicle Extraction and Speed Detection from Digital Aerial Images," *IEEE International Geoscience and Remote Sensing Symposium,* vol. 3, pp. III-1334, 2008.

[40] C. Cortes and V. Vapnik, "Support Vector Networks," *Machine Learning,* pp. 273-297, 1995.

[41] N. Cristianini and J. Taylor, "An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods," *Cambridge University,* 2000.

[42] N. Ghoggali, F. Melgani and Y. Bazi, "A Multiobjective Genetic SVM-Approach for Classification Problem With Limited Training Samples," *IEEE Trans. On Geoscience and Remote Sensing,* vol. 47, no. 6, 2009.

[43] N. Ghoggali and F. Melgani, "Genetic SVM-Approach to Semisupervised Multitemporal Classification," *IEEE on Geoscience and Remote Sensing Letters,* vol. 5, no. 2, 2008.

[44] F. Melgani and L. Bruzzone, "Classification of Hyperspectral Remote Sensing Images with Support Vector Machines," *IEEE Transaction on Geoscience and Remote Sensing,* vol. 42, no. 8, pp. 1778-1790, 2004.

[45] P. Maragos and R. W. Schafer, "Morphological Filters-Part I: Their Set-Theoretic Analysis and Relations to Linear Shift-Invariant Filters," *IEEE Transaction On Acoustics, Speech and Signal Processing,* vol. 35, no. 8, pp. 1153-1169, 1987.

[46] S. Belongie, J. Malik and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *EEE Transaction on Pattern Analysis and Machine Intelligence,* vol. 24, no. 4, pp. 509-522, 2002.

[47] S. Lazebnik, C. Schmid and J. Ponce, "Sparse Texture Representation Using Affine-Invariant Neighborhoods," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[48] W. Freeman and E. Adelson, "The Design and Use of Steerable Filters," *IEEE Transaction on Pattern Analysis and Machine Intelligence,* vol. 13, no. 9, pp. 891-906, 1991.

[49] J. Koenderink and A. Van Doorn, "Representation of Local Geometry in the Visual System," *Biological Cybernetics,* Vols. 367-375, no. 6, p. 55, 1987.

[50] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Transaction on Pattern Analysis and Machine Intelligence,* vol. 27, 2005.

[51] W. Burger and M. Burge, Digital Image Processing - An Algorithmic Introduction Using Java, London: Springer, 2007.

[52] J. Benediktsson, M. Pesaresi and K. Amason, "Classification and Feature Extraction for Rremote Sensing Images from Urban Areas Based on Morphological Transformations," *IEEE Transaction on Geoscience and Remote Sensing,* vol. 41, no. 9, pp. 1940-1949, 2003.

[53] D. Tuia, F. Pacifici, K. Kanevski and W. Emery, ""Classification of Very High Spatial Resolution Imagery Using Mathematical Morphology and Support Vector Machines," *IEEE Transaction on Geoscience and*

Remote Sensing, vol. 47, no. 11, pp. 3866-3879, 2009.

[54] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *IEEE computer Society Conference on Computer Vision and Pattern Recognition,* vol. 1, pp. 886-893, 2005.

[55] T. Cover and T. J.A., Elements of Information Theory, New York: Wiley, 1991.

[56] I. Vajda, Theory of Statistical Inference and Information, Dordrecht, Netherlands: Kluwer, 1989.

[57] J. Pluim, A. Maintz and M. A. Viergever, "Mutual-Information-Based Registration of Medical Images: A Survey," *IEEE Trans. on Medical Imaging,* vol. 22, no. 8, 2003.

[58] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers,* 1999.

[59] S. Sun and Z. Zeng, "UAV image mosaic based on adaptive SIFT algorithm," in *IEEE international Conference in Geoinformatics*, 2013.

[60] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica,* vol. 11, pp. 23-27, 1975.

[61] L. Duriex, E. Lagarbrielle and A. Nelson, "A method for monitoring building construction in urban sprawl areas using object-based analysis of Spot 5 images and existing GIS data," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 63, no. 4, pp. 399-408, 2003.

[62] C. Corbane, L. Najman, E. Pecoul and L. Demagistri, "A Complete Processing Chain for Ship Detection Using Optical Satellite Imagery," *International Journal of Remote Sensing,* vol. 21, no. 22, pp. 5837-5854, 2010.

[63] D. Manolakis, E. Truslow, M. Pieper, T. Cooley and M. Brueggeman, "Detection Algorithms in Hyperspectral Imaging Systems: An Overview of Practical Algorithms," *IEEE Signal Processing Magazine,* vol. 31, no. 1, pp. 24-33, 2014.

[64] N. Nasrabadi, "Hyperspectral Target Detection: An Overview of Current and Future Challenges," *IEEE Signal Processing Magazine,* vol. 31, no. 1, pp. 34-44, 2014.

[65] S. Xu, T. Deng, D. LI and S. Wang, "Object Classification of Aerial Images With Bag-of-Visual Words," *IEEE GEOSCIENCE AND REMOTE SENSING LETTERS,* vol. 7, no. 2, pp. 366-370, 2010.

[66] A. Selim, I. Z. Yalniz and K. Tasdemir, "Automatic Detection and Segmentation of Orchards Using Very High Resolution Imagery," *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING,* vol. 50, no. 8, pp. 3117-3131, 2012.

[67] B. Sirmacek and C. Unsalan, "Urban-Area and Building Detection Using SIFT Keypoints and Graph Theory," *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING,* vol. 47, no. 4, pp. 1156-1167, 2009.

[68] P. Felzenszwalb, R. Girshick, D. McAllester and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Trans. On Pattern Analysis and Machine Intelligence,* vol. 32, no. 9, pp. 1627-1645, 2010.

[69] L. Merino, F. Caballero, J. Martinez-de Dios and A. Ollero, "Cooperative Fire Detection using Unmanned Aerial Vehicles," in *International Conference on Robotics and Automation*, Barcelona, Spain, 2005.

[70] S. Adams, C. Friedland and M. Levitan, "Unmanned Aerial Vehicle Data Acquisition for Damage Assessment in Hurricane Events," in *International Workshop on Remote Sensing for Disaster Management*, Tokyo, Japan, 2010.

[71] T. Moranduzzo, F. Melgani, "Detecting Cars in UAV Images with a Catalog-Based Approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 10, pp. 6356-6367, Oct. 2014.

[72] T. Moranduzzo, F. Melgani, "An automatic car counting method for unmanned aerial vehicle images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 3, pp. 1635-1647, Mar. 2014.

[73] C. E. Rasmussen and C. K. I. Williams, Gaussian Process for Machine Learning, Cambridge, Massachussets: The MIT press, 2006.

[74] L. Pasolli, F. Melgani and E. Blanzieri, "Gaussian Process Regression for Estimating Chlorophyll Concentration in Subsurface Waters from Remote Sensing Data," *IEEE Geoscience and Remote Sensing Letters,* vol. 7, no. 3, pp. 464-468, 2010.

[75] B. Yegnanarayana, Artificial Neural Networks, PHI Learning Pvt. Ltd., 2009.

[76] S. T. Roweis and K. L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science,* vol. 290, no. 2000, pp. 2323-2326.

[77] S. Moustakidis, G. Mallinis, N. Koutsias, J. Theocharis and V. Petridis, "SVM-based fuzzy decision trees for classification of high spatial resolution remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 50, no. 1, pp. 149-169, 2012.

[78] S. Xu, T. Fang, D. Li and S. Wang, "Object classification of aerial images with bag-of-visual words," *IEEE Geoscience and Remote Sensing Letters,* vol. 7, no. 2, pp. 366-370, 2010.

[79] W. Li, S. Prasad and E. Fowler, "Decision Fusion in Kernel-Induced Spaces for Hyperspectral Image Classification," *IEEE Transaction on Geoscience and Remote Sensing,* vol. 52, no. 6, pp. 3399-2014, 2014.

[80] C. Shiyong and M. Datcu, "Coarse to fine patches-based multitemporal analysis of very high resolution satellite images," in *nternational Workshop on the Analysis of Multi-temporal Remote Sensing Images (Multi-Temp)*, 2011.

[81] R. R. Vatsavai, A. Cheriyadat and S. Gleason, "Unsupervised semantic labeling framework for identification of complex facilities in high-resolution remote sensing images," in *IEEE International Conference on Data Mining Workshops (ICDMW)*, 2010.

[82] Y. Zhang, J. Rong and Z. Zhi-Hua, "Understanding bag-of-words model: a statistical framework," *International Journal of Machine Learning and Cybernetics,* vol. 1, no. 1-4, pp. 43-52, 2010.

[83] J. Wu, Advances in K-means Clustering: A Data Mining Thinking, Springer, 2012.

[84] O. Pele and M. Werman, "The quadratic-chi histogram distance family," in *Computer Vision–ECCV 2010*, Berling, Heidelberg, Springer, 2010, pp. 749-762.

[85] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines.," *ACM Transactions on Intelligent Systems and Technology (TIST),* 2011.

[86] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms.".

[87] C. Rasmussen and H. Nickish, "http://gaussianprocess.org/gpml/code," [Online].

8. List of Related Publications

8.1. Published Journal Papers

[J.1] T. Moranduzzo, F. Melgani. An automatic car counting method for unmanned aerial vehicle images. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 3, pp. 1635-1647, Mar. 2014.

[J.2] T. Moranduzzo, F. Melgani. Detecting Cars in UAV Images with a Catalog-Based Approach. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 10, pp. 6356-6367, Oct. 2014.

[J.3] T. Moranduzzo, F. Melgani, Y. Bazi, N. Alajlan. A Fast Object Detector Based on High-Order Gradients and Gaussian Process Regression for UAV Images. *International Journal of Remote Sensing, in press.*

8.2. Conference Proceedings

[C.1] T. Moranduzzo, F. Melgani. A SIFT-SVM method for detecting cars in UAV images. *IEEE International Geoscience and Remote Sensing Symposium* 2012, pages: 6868-6871, Munich, Germany, 2012.

[C.2] T. Moranduzzo, F. Melgani, A. Daamouche. An object detection technique for very high resolution remote sensing images. *IEEE International Workshop on Systems, Signal Processing and their Applications (WoSSPA)* 2013, pages: 79-83, Algiers, Algeria, 2013

[C.3] T. Moranduzzo, F. Melgani. Comparison of different feature detectors and descriptors for car classification in UAV images. *IEEE International Geoscience and Remote Sensing Symposium* 2013. pages: 204-207, Melbourne, Australia, 2013.

[C.4] T. Moranduzzo, F. Melgani. Car Speed Detection in UAV Images. *IEEE International Geoscience and Remote Sensing Symposium 2014*, pp. 4943-4945, Quebec City, Canada, 2014.

[C.5] T. Moranduzzo, F. Melgani. Monitoring Structural Damages in Big Industrial Plants With UAV Images. *IEEE International Geoscience and Remote Sensing Symposium 2014*, pp. 4950-4953, Quebec City, Canada, 2014.