POLITECNICO DI MILANO
DIPARTIMENTO DI ELETTRONICA, INFORMAZIONE E BIOINGEGNERIA
DOCTORAL PROGRAMME IN INFORMATION TECHNOLOGY
COMPUTER SCIENCE AND ENGINEERING

EXPLOITING PUBLIC WEB CONTENT TO ENHANCE ENVIRONMENTAL MONITORING

Doctoral Dissertation of:
Roman Fedorov

Advisor:
Prof. Piero Fraternali

Tutor:
Prof. Andrea Castelletti

Chair of the Doctoral Program:
Prof. Andrea Bonarini

2017 – XXIX

Алексею и Ирине.

Acknowledgments

It is difficult to do justice with few lines of text to the importance that these people had in my PhD adventure and to the gratitude I feel.

To Piero Fraternali, for all of this. For introducing me to the research world, for his mentorship and effort in getting me here, for truly caring for me and the success of this work.

To Marco Tagliasacchi, for his constant support, expertise and guidance, for being an example to follow. Also, for not letting me die somewhere in the middle of Tanzania.

To Andrea Castelletti and Matteo Giuliani, for their precious environmental modeling contributions that greatly increased the quality of this work.

To Chiara Pasini, for making me get stuff done and for the enormous help in writing this thesis.

To Patrice Carbonneau and Alexis Joly, for their expertise and for taking time to provide incredibly detailed and useful reviews.

To Alexander Varshavsky, for the priceless opportunity and being a wonderful supervisor.

To David J. Crandall, for making me feel a hotshot researcher.

To the folks, too many to be listed here, for their important contributions to this work and for best part of all this: coffee breaks and small talks.

Heartfelt thanks.

Abstract

THE quantity of public content, generated by users or by sensors, available on the web nowadays is reaching unprecedented volumes. This massive collection of data contains an enormous amount of latent knowledge, which can be used for a variety of purposes, such as event detection and predictive modeling. The goal of the research presented in this thesis is to explore the methods for extracting such knowledge and building useful applications using low-cost, publicly available, multimedia web content, with reference to the field of environmental monitoring, which often suffers from the lack of significant and exhaustive input data. This objective requires addressing such challenges as content acquisition, content normalization and fusion, analysis and correlation, and model building and training. Specifically, the focus is set on monitoring snow cover in mountainous regions, that is, the spatial extent of earth surface covered by snow. The effort exploits visual data, terrestrial photography crawled from the public image sharing websites and publicly available webcams. We present algorithms for retrieving and analyzing such data, and prove its usefulness thanks to a data-driven environmental model. The experiments prove that, in the described use case, the virtual snow indexes obtained from a single public webcam are able to replace the original authoritative snow measurements without a performance drop and improve the performance if complemented to the authoritative snow measurements. These results confirm the feasibility of exploiting visual content publicly available on the web in real world environmental monitoring scenarios.

Sommario

La quantità di contenuti pubblici, generati dagli utenti o da sensori, disponibile oggi sul web, sta raggiungendo volumi senza precedenti. Questa considerevole raccolta di dati contiene un'enorme quantità di informazioni nascoste, che possono essere utilizzate per svariati scopi, come l'individuazione degli eventi e il predictive modeling. L'obiettivo della ricerca presentato in questa tesi è quello di esplorare i metodi per estrarre tali informazioni e costruire applicazioni utili utilizzando contenuti web a basso costo e pubblicamente accessibili, con riferimento al campo del monitoraggio ambientale, che spesso soffre dalla mancanza di dati di input significativi ed esaustivi. Questo obiettivo richiede di affrontare sfide come l'acquisizione dei contenuti, la loro normalizzazione e fusione, l'analisi e la correlazione, la creazione di modelli e addestramento. In particolare, l'attenzione è posta sul monitoraggio della neve in regioni montuose, ovvero l'estensione spaziale della superficie terrestre coperta dalla neve. Il lavoro sfrutta i dati visuali: fotografie estratte dai siti web di condivisione di immagini pubbliche e webcams pubblicamente disponibili. Vengono presentati gli algoritmi per il recupero e l'analisi di tali dati e viene dimostrata la loro utilità grazie ad un modello ambientale basato su di essi. Gli esperimenti dimostrano che, nel caso d'uso descritto, gli indici della neve virtuale ottenuti da una singola webcam pubblica sono in grado di sostituire le misurazioni ufficiali delle neve mantendo lo stesso livello di prestazioni, mentre le prestazioni migliorano se gli indici della neve virtuale vengono integrati alle misurazioni ufficiali. Questi risultati confermano la fattibilità di sfruttare contenuti visuali pubblicamente disponibili sul web in scenari di monitoraggio ambientale del mondo reale.

Contents

CHAPTER *1*

Introduction

Over the past two decades, the scientific and technological advances in the web, mobile hardware and software fields fostered a phenomenon known as the *social web*. The social web can be seen as a global network wherein millions of human sensors across the globe capture different aspects of the spatio-temporal processes, which can be aggregated into a holistic view [164]. The unprecedented amount of content publicly available on the social web poses unique opportunities for extracting valuable information: human-sensors can be used to observe phenomena that are impossible, hard or costly to observe by hardware alone. This social media knowledge (also known as *collective intelligence* or *wisdom of the crowd*) has been successfully used in different fields, such as urban monitoring and planning [82], social interactions analysis [58], marketing [41] and even politics [25].

One of the fields that has historically been based on expert observations and hardware sensors is *environmental monitoring*. Environmental monitoring is used to observe conditions of the environment in order to study environmental changes, particularly those arising from human activities. Understanding and predicting the environment evolution is something that humanity has been interested in since prehistoric times [146], and the progress made in this area in the last decades thanks to scientific and hardware breakthroughs is outstanding [75]. Environmental monitoring requires collecting measurements of a very diversified range of physical quantities, which are then fed to models aimed at understanding past observations (e.g., climate change), detecting critical events in real-time (e.g., bush fires) and making predictions for the future (e.g., availability of water resources). Traditionally, such measurements are obtained by means of specialized instrumentation that is designed, installed and managed by researchers and professionals interested in their analysis.

The advance of the social web, however, offers unique opportunities for novel environmental monitoring approaches. The public content available on the social web can be used to enrich traditional measurements by increasing coverage along both the spatial and temporal dimension. Specifically, thanks to the diffusion of web-connected devices, millions of **publicly available** photographs and videos are uploaded daily to the web [95]. These photographs are generated by humans using personal mobile devices as well as by Internet of Things (IoT) devices (e.g., webcams). The uploaded photographs contain often a geotag, timestamp and depict outdoor scenarios. As such, they carry an incredible amount of environmental observations that is waiting to be found, extracted and used in environmental applications.

1.1 Problem Statement

The goal of this thesis is to study the feasibility of exploiting visual content publicly available on the social web for environmental monitoring purposes. The thesis aims to answer this question by illustrating a use case in which the problem has been successfully addressed. This includes the complete path from retrieving and analyzing the data up to using it for environmental purposes and proving its usefulness.

Problem Statement:

Given a massive amount of unstructured and non-authoritative visual content publicly available on the social web, devise techniques for the automatic analysis of such content, so as to extract environmentally relevant spatio-temporal data and objectively prove the utility of such data.

We specifically focus on monitoring snow cover in mountainous regions, that is, the spatial extent of terrain surface covered by snow. Snow processes are traditionally observed by means of ground measurements stations, which can either be manned or fully automated. In both cases, measurements are accurate and capture different aspects, including the snow depth and density (possibly at different altitudes). However, the number of ground measurement stations is limited (for example, only 46 stations are currently deployed over an area of 10 500 km^2 covering the Italian Alps in the region of Lombardy), thus enabling only a sparse sampling of the snow cover over large areas. Moreover, the high variability of snow processes, which depend on temperature, elevation, exposure, slope, winds, etc., is such that it is difficult to extrapolate snow depth and density at different locations. An alternative source of measurements is represented by remote sensing, which relies on satellite [81] or airborne [135] imagery, synthetic aperture radar interferometry [161], laser scanner altimetry [145]. These methods can provide a very high spatial coverage at moderate spatial resolution, but can be insufficient for applications that require high temporal density (e.g. daily), since observations might not be available due to cloud cover and limited temporal frequency of satellite imagery.

We chose public photographs taken in mountain regions as the use case of visual content. A large fraction of them contains the skyline defined by mountain peaks, slopes, ridges, crests, etc., both as main subject and as background. Such photographs implicitly contain precious information related to snow cover phenomena, which can complement the traditional measurements and has not been fully exploited so far.

1.2 Research questions

In this section, we formulate the research questions that motivate the work of this thesis; we list the questions following the logical order in which they should be answered to in order to successfully address the problem.

Research Question 1. *How to acquire relevant public visual content from the social web?*

Crawling data from the social platforms and public photo-sharing websites is a trivial task that requires low implementation and research effort. However, automatically estimating if a certain piece of content is *relevant* and *suitable* for the desired application requires non-trivial content processing.

Research Question 2. *Is the amount of the available content suitable for environmental analysis?*

Although the statistics on the amount of available web data sound impressive, it is not obvious whether the relevant data is actually enough to provide a sufficient spatial and temporal resolution, enabling effective environment process analysis.

Research Question 3. *How to automatically extract environmentally relevant spatio-temporal data from the visual content at scale?*

This requires high-performance and high-accuracy image processing and computer vision algorithms to extract data from uncontrolled collections of photographs acquired from non-authoritative sources.

Research Question 4. *Can the extracted environmental data be objectively proved to be useful for real environmental applications?*

The existing works in the literature usually prove the fact that the obtained data is correlated w.r.t. to a certain phenomena/process, which can support the assumption that the data can be potentially useful, but do not prove the fact that the data is practically useful in a specific environmental application that already uses environmental measurements obtained from traditional sources.

Research Question 5. *Can active crowdsourcing be leveraged to engage users into providing more relevant data and improving the existing data analysis?*

Crowdsourcing has been proved successful when accompanying automatic data analysis pipelines in different applications, such as content processing, query processing and relevance feedback processing [61]. We investigate if the voluntary citizen contribution can benefit also the approaches that analyze social media for environmental purposes.

1.3 Contributions

In this thesis we explore the feasibility of using social web content to monitor snow cover. The objective is not to replace the use of ground-, satellite- or airborne-based measurements. We argue that social web content might represent an additional source

that can complement and enrich the traditional ones, due to its unique characteristics in terms of spatio-temporal coverage resolution and cost. We focus on *visual* content and its accompanying metadata, which can be obtained from two different sources: user-generated photographs posted on social media websites and image feeds from outdoor webcams.
The contributions of this thesis, schematized in Figure 1.1, are as follows:

- We describe an acquisition pipeline that continuously retrieves new images containing mountain slopes from photo-sharing platforms and public webcams. The pipeline deals with all the necessary aspects in order to output only relevant geolocated photographs (this answers to Research Question 1 and Research Question 2). The specific contributions can be summarized as follows:
  - we design a binary classifier that estimates whether a photograph does or does not contain a relevant mountain view;
  - we design an algorithm that filters out webcam images affected by bad weather conditions, aggregates all daily images into a single combined image and also mitigates the webcam shaking;
  - we describe the implementation of the social network photograph crawler that acquired from Flickr more than 600 k relevant user-generated photographs taken in an extended Alpine area during the last 7 years, and the webcam crawler that acquired more than 100 M images from $\sim$ 2 k public webcams in the whole Alpine region over the last 2 years.
- We describe a set of algorithms for mountain image geo-registration that, given a geolocated photograph as input, infer photograph geographical properties (this answers partially to Research Question 3). Specifically, the algorithms estimate:
  - photograph direction, i.e. the orientation of the camera during the shot;
  - on-screen photograph coordinates of the corresponding visible mountain peaks;
  - for every photograph pixel, if it corresponds to the sky or to the terrain and in the latter case, terrain-specific properties, such as GPS position, extent, altitude and distance from the observer.
- We realize novel approach for snow/non-snow pixel level classification and propose several virtual snow cover measures (this answers partially to Research Question 3), including:
  - *physical* measures that describe a specific real-world physical measure, such as the snow line altitude;
  - *non-physical* measures that do not carry a specific meaning in terms of a physical measure, but are correlated with an environmentally relevant trend that can be used by a data-driven model.
- In collaboration with environmental researchers [23, 71], we present a supervised learning data-driven water management model that, among other inputs, relies on the authoritative snow measurements provided by the Italian Environmental Protection Agency (Agenzia Regionale per la Protezione dell'Ambiente, ARPA).

We define the performance metric of the model and test how the performance varies if we complement or replace the authoritative snow measurements with the virtual snow indexes computed by the proposed system (this answers to Research Question 4). We argue that the data acquired from one single touristic webcam is able to:

  - *replace* the authoritative snow measurements without a performance drop;
  - *complement* the authoritative snow measurements, improving the performance.

- Finally, we discuss the potential of the crowdsourcing in systems that exploit unstructured content for environmental monitoring (this answers to Research Question 5) with two use cases:

  - a web portal that allows users to explore the acquired photographs, contribute their own content and help the geo-registration process by correcting the errors made by automated tools;
  - a real-time augmented reality mobile application that identifies mountain peaks, engaging the users to contribute with their photographs. To this end, we also describe a variant of the photograph geo-registration approach executable real-time on low power devices: this allows us to run the algorithms directly on consumer mobile phones and engage users with an entertaining experience.

1.4 Structure of the thesis

The structure of the thesis follows the logical flow depicted in Figure 1.1:

Chapter 2 discusses the background of the work contained in this thesis. It describes the evolution of the environmental monitoring techniques that adopt social web content, and illustrates how such content is acquired, processed and validated in the state-of-the-art.

Chapter 3 describes the mountain image acquisition pipeline and the underlying algorithms.

Chapter 4 describes the devised approach for mountain photograph geo-registration that enriches the photograph with metadata regarding the position of the photograph w.r.t. the terrain. It also describes how this approach can be adapted to be used in real-time on mobile devices, enabling an augmented reality experience.

Chapter 5 narrows the discussion to a specific environmental use case, which is snow cover. It presents novel algorithms for image snow cover identification and proposes several virtual snow cover measures.

Chapter 6 introduces the water management model and reports how the performance of the model responds when the virtual snow cover measures are fed in input.

Chapter 7 provides two use cases of crowdsourcing techniques applied to enhance the automatic analysis processing.

Finally, **Chapter 8** concludes the thesis, discusses the current challenges and proposes the future direction of the research in this area.

This thesis includes the material from the following publications, co-authored by the candidate:

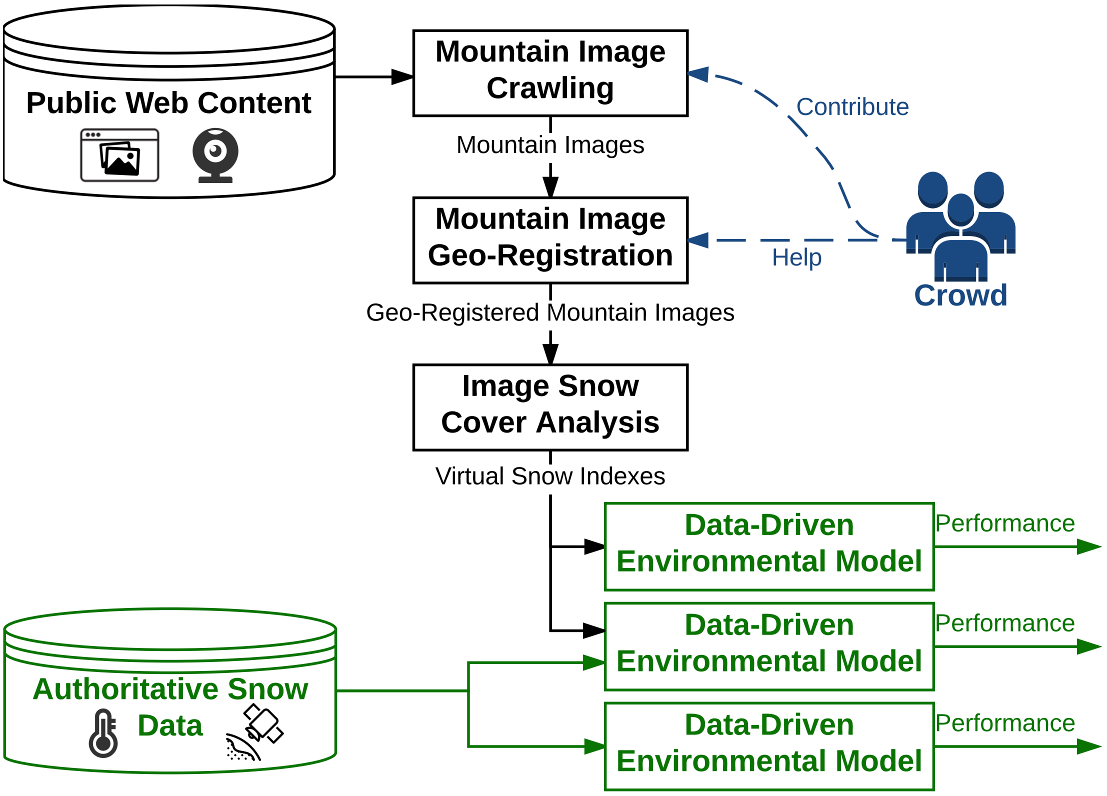

Public Web Content
Mountain Image Crawling
Contribute
Mountain Images
Mountain Image Geo-Registration
Help
Crowd
Geo-Registered Mountain Images
Image Snow Cover Analysis
Virtual Snow Indexes
Data-Driven Environmental Model
Performance
Data-Driven Environmental Model
Performance
Authoritative Snow Data
Data-Driven Environmental Model
Performance

Figure 1.1: *Schematic overview of the thesis contributions and structure. Black items represent the automatic data processing pipelines, green items represent environmental models and applications, blue items represent the crowdsourcing approaches.*

- Roman Fedorov, Piero Fraternali, and Marco Tagliasacchi. "Mountain peak identification in visual content based on coarse digital elevation models" [54].
- Roman Fedorov, Alessandro Camerada, Piero Fraternali, and Marco Tagliasacchi. "Estimating snow cover from publicly available images" [50].
- Roman Fedorov, Piero Fraternali, and Chiara Pasini. "SnowWatch: a multi-modal citizen science application" [53].
- Roman Fedorov, Darian Frajberg, and Piero Fraternali. "A framework for outdoor mobile augmented reality and its application to mountain peak detection" [52].
- Andrea Castelletti, Roman Fedorov, Piero Fraternali, and Matteo Giuliani. "Multimedia on the mountaintop: Using public snow images to improve water systems operation" [23].
- Matteo Giuliani, Andrea Castelletti, Roman Fedorov, and Piero Fraternali. "Using crowdsourced web content for informing water systems operations in snow-dominated catchments" [71].

CHAPTER 2

Background

The idea of using content acquired from the social web in environmental monitoring scenarios is not new. In this chapter we provide an overview of the scientific literature that addresses this problem, highlighting the novel contributions of this thesis w.r.t. the current state of the art.

The chapter is structured as follows: Section 2.1 provides a brief historical perspective on the adoption of the social web in environmental monitoring, Section 2.2 overviews the acquisition of the data from the social web while Section 2.3 deals with the processing of such data. Finally, Section 2.4 describes the evaluations that asses the value of the obtained environmental information.

In order to provide a complete picture, we present a generic literature review that covers the full range of environmental monitoring applications and different content types. Nevertheless, for the sake of a fair description of the novelty of the algorithms proposed in this thesis, we also properly narrow the background discussion to specific use cases in Section 2.3.4, describing the state-or-the-art techniques for mountain image analysis and image snow cover estimation.

2.1 Web and Environmental Monitoring Evolution

In this section we walk through the events that led to the advent of the social web and illustrate how these events gradually introduced new environmental monitoring approaches that use non-authoritative observations. Although providing a historical perspective is not the primary goal of this chapter, a general overview of this evolution is crucial to understand the dynamics that brought this research area to its current state-of-the-art form.

2.1.1 Citizen Science

The first deviation of the environmental monitoring from the use of the sole authoritative data occurred with the birth of *citizen science*. *Citizen science* evokes a science that assists the needs and concerns of citizens, and at the same time implies a form of science developed and enacted by citizens themselves [91]. Citizen science can also be described as "a process where concerned citizens, government agencies, industry, academia, community groups, and local institutions collaborate to monitor, track and respond to issues of common community [environmental] concern" [180]. In other words, citizen science shifted the paradigm "*only scientists participate to environmental monitoring*" to "*also citizens participate to environmental monitoring*".

One of the oldest documented environmental citizen science entities, *Earthwatch Institute*[1], was founded in 1971 to offer volunteers the opportunity to join research teams through the collection of field data in the areas of rainforest ecology, wildlife conservation and marine science. Nowadays, citizen science communities cover the full range of environmental and ecological monitoring areas, from water monitoring (*Waterkeeper Alliance*[2], *URI Watershed Watch* [79], *Florida Lakewatch* [19]), air quality monitoring (*The Bucket Brigade*[3]), plant monitoring (*Pl@ntNet*[4], *Project Budburst*[5], *iNaturalist*[6], *iSpot*[7]) up to animal species monitoring (*Reef Environmental Education Foundation*[8], *FrogWatch*[9], *Celebrate Urban Birds*[10], *eBird*[11]).

The benefits of citizen science are an increasing environmental democracy (sharing of information), volunteer engagement, data provided at no- or low-cost to governments and early warning/detection systems. The challenges, on the other hand, generally include the lack of volunteer interest (due to, among others, the lack of networking opportunities) and the inability to access appropriate information or expertise [30]. The birth of citizen science occured before the adoption of the Internet, thus, it is easy to understand why the challenge of inaccessible information was hard to overcome.

The citizen science concept is pivotal inside the scope of this work. In the strict sense, every technique described in this thesis is a citizen science approach, since we collect data, directly or indirectly, thanks to the citizens. For the sake of the clarity, however, in the rest of the thesis we use the term *citizen science* for works that involve direct engagement of the citizens. We adopt *data crawling* and *passive acquisition* terms, instead, in scenarios that involve automatic data collection from social networks or from physical web-connected devices.

The citizen science literature contains a huge number of terms with overlapping definitions. Citizen science applied to environmental monitoring is based on spatio-temporal aggregations and volunteer communities, for the sake of this discussion we consider the following terms as the synonyms of the environmental citizen science: *Community-Based Monitoring (CBM)* [180], *Volunteered Geographical Information (VGI)* [17],

[1] http://earthwatch.org

[2] http://waterkeeper.org

[3] http://labucketbrigade.org

[4] http://identify.plantnet-project.org/

[5] http://budburst.org

[6] https://www.inaturalist.org

[7] https://www.ispotnature.org/

[8] http://www.reef.org

[9] http://www.aza.org/frogwatch

[10] http://celebrateurbanbirds.org

[11] http://www.birds.cornell.edu

Participatory GIS, Public Participatory GIS (PPGIS) [17, 136], *Citizens as Sensors* and *Citizens as Voluntary Sensors* [73, 74], *Participatory Sensing* [147].

2.1.2 Web

In 1989, English scientist Tim Berners-Lee invented the World Wide Web, an information space that allows documents and other resources to be accessed through the Internet. Although this invention was going to deeply change the world, the real impact of the web on the society must be postponed by nearly a decade. 1994 was characterized by thousands of notable websites, while the beginning of the web commercialization and its exponential growth is estimated to be between 1996 and 1998.

The citizen science communities are based on the engagement and the extent of their communication campaigns: unsurprisingly, the web had a great impact on these communities. The novel channel for the communication and public engagement rapidly spread across the existing citizen science projects, and encouraged the development of the new ones [162]. Not by chance, Earthwatch Institute unveiled its first website in 1994 and hosted the first live educational web broadcast in 1996.

As a matter of fact, the consensus between the researches states that partnering with existing organizations, such as civic groups, neighborhood organizations, non-profit environmental protection groups is an effective way to reach target communities, and providing constant support through email list-servers and online discussion boards is foundamental to retain participants [31].

Although the late nineties were characterized by an improved one-way web communication (from scientists to citizens), the real potential of the web applied to the environmental science remained latent for several other years, until the bidirectional data flow was finally unlocked thanks to the *social web*.

2.1.3 Social Web

Social web is the term that is usually used to define the set of platforms that allow users to communicate on the web through the social media tools. The term is often associated with *web 2.0*, which (even if coined in 1999) gained popularity in late 2004. The social web refers to the websites that emphasize user-generated content. The increased popularity of social media articles and microblogging systems changed the way the online information is produced: users switched from being only content consumers to being both content publishers and content consumers.

The citizen science advent revolutionized the way the environmental monitoring is performed, while the social web, in turn, revolutionized the way the citizen science campaigns are performed, with novel methods for information dissemination, user engagement and feedback collection. In fact, according to [107], citizen science moved from the traditional "*scientists using citizens as data collectors*" concept to "*citizens as scientists*" thanks to the social web.

The novel approaches for the user engagement, however, are just half the story: ever since the rise of the social web, users are creating a *massive amount* of publicly available content. This content contains knowledge, both intentionally provided (e.g. a timestampted photograph of a user in front of the Eiffel Tower states that the user was in Paris in that moment) and unintentionally provided (e.g. the same photograph acts as an observation of the meteorological conditions in Paris in that moment).

Being public, though, this content can be used for purposes that it was not originally intended for: the unprecedented availability of the user-generated data on the social web poses unique opportunities for extracting valuable environmental measurements from such data. These can be used to enrich traditional measurements by increasing coverage along both the spatial and temporal dimension.
Beside high data volume and easiness of access advantages, the adoption of the user-generated content has several drawbacks, which mainly converge to the roughness of the data. This content is usually intended to be consumed by other human social users, so it is poorly structured and tends to require a significant processing effort. In particular, if we consider every user-generated datum as a single virtual observation of a real world event, we must face the problem of understanding: what did happen, where did it happen and when did it happen. Working with user-generated content revokes the possibility of asking these questions directly to the user, forcing the researchers to devise automatic data processing techniques.

2.1.4 Webcams

In 1991 the first webcam ever was pointed at a coffee pot in Cambridge University, allowing the department personnel to check the coffee availability without leaving their desks. Since then, webcam technology evolved significantly and the range of webcam use cases expanded far beyond coffee monitoring, but the key concept of webcam adoption remained unaltered: provide a group of people an easy and universal web access to a visual real-time snapshot of reality. The idea of the low cost real-time image stream provided through the web was so successful that nowadays, twenty years after the first commercial webcam launch [29], public webcams densely cover the world [92].
The reason webcams gained such success and popularity lays in the simplicity of providing information that would require significant processing effort otherwise. The human brain is an extremely powerful processor and webcams largely exploit this: the scholars from Cambridge University could have installed a sophisticated physical coffee level hardware sensor inside the pot and transmit the coffee level over the web, but a simpler solution was to point a camera at the pot, transmit the raw data (images) over the web and let the brain of each individual to process the data and estimate the coffee level.
Webcams became universal, real-time application-independent sensors, which can provide a large quantity of relevant information. Without any sophisticated hardware, software, communication tools or constant human intervention, nowadays, one can easily:

- check a mountain hut webcam and decide whether it is a good day for hiking;
- check a seafront webcam and evaluate how much is the beach crowded;
- check webcams placed on different roads and decide to avoid routes affected by heavy traffic.

Although the webcam industry found a commercial niche in the security and surveillance business, one of the most popular webcam use cases is providing uncontrolled public access to the data. The low cost of the devices and the easiness of the deployment (a power and internet connections are sufficient) contributed to the growth of the network of the publicly available webcams. Numerous public webcams are placed outdoors and capture snapshots of the environment. There are several reasons a private citizen or an entity would install a public outdoor webcam for, among which:

- **Commercial and touristic**: a strategically placed webcam can provide information regarding the touristic attractions and points of interest, reveal the potential beauty of the place and attract tourists [173].
- **Meteorological**: visual feed from a webcam can provide information regarding the current meteorological conditions, such as the presence of fog, rain or clouds [128].
- **Environmental**: webcams can be also used by scientists in environmental use cases, providing information regarding the vegetation, snow and water phenomena [14].
- **Ecological**: webcams are an efficient non invasive tool for ecological and wild life monitoring [176].

Although webcams appeared on the market before the advent of the social web, we argue that, nowadays, public webcams actually represent the social web paradigm: thanks to the decreased costs, common users deploy the webcams and thus become providers of the web content. Furthermore, the idea of webcams as providers of data only "for human consultation" is recently being abandoned, thanks to the advances in image processing and computer vision fields.

2.1.5 Mobile Web

The *mobile web*, i.e. the set of web-based services on mobile devices, was developed since 2007 with the raise of the consumer large multi-touch smartphones. Although the formal definition of the term implies that the applications should be browser-based, nowadays, the distinction between browser and native applications can be inappreciable. In this discussion we treat all mobile web-based applications equally.

While the social web changed the way the citizen science connects to the environmental monitoring, the advent of the mobile web, in turn, deeply changed the way people interact on the social web. First, thanks to the mobile web people currently carry web-connected devices on daily basis while being outdoor. Second, these devices are equipped with new sensors, such as camera, compass, microphone and GPS sensor.

The reaction to these new portable opportunities was immediate, with the release of numerous mobile applications for the environmental monitoring [57], and even frameworks that enable people without programming skills to build mobile data collection and management tools for the citizen science campaigns [99].

For example, project Budburst aims at gathering information regarding the flowering of native plants to study the climate change [147]: mobile phones are used by volunteers to upload timestamped, geotagged plant photographs. Other examples of using mobile devices for collecting geotagged photographs are: online database for avian surveys with access for citizens and scientists alike [170], water level monitoring [122], noise pollution detection [124] and meteorology monitoring [98].

The mobile web had a strong impact also on the way users generate content on the social web. First, the volume of the public social media increased due to the increased social networking (nowadays, more than half of Facebook users access the service on **mobile only** [134]). Second, the user-generated content became richer thanks to the increased amount of visual content (everyone has a camera at all times). Furthermore, the content became partially real-time and accompanied by geolocation information.

The advent of the mobile web finally succeeded at creating the human-sensor network: every socially-active citizen is carrying a mobile device that has been transformed into a sensor. It can be argued that the mobile web finally enabled successful data crawling from the social web: all of the works referenced in this chapter that passively mine content from the social web for environmental monitoring purposes are subsequent to 2007.

2.2 Data Acquisition

Traditional environmental monitoring approaches collect observations from authoritative sources, such as scientists and hardware sensors. The environmental monitoring approaches that use the social web, instead, focus on non-authoritative sources, i.e. social web users and web-connected devices deployed by social web users.

The data can be collected in two conceptually different ways: directly engaging the users into providing the data or passively collecting the existing data. Due to the distinct characteristics of the two approaches, we discuss them separately.

Section 2.2.1 describes the *Active Data Acquisition* approach, in which the users are engaged and motivated to actively provide environmental observations. We also refer to this approach as *crowdsourcing* or *citizen science*.

Section 2.2.2, instead, describes the *Data Crawling* scenario, in which the users are unaware of the fact that the content they are publishing on the web is being passively collected and treated as environmental observations.

Both active and passive data acquisition techniques are relevant to this thesis. While the major contributions of the thesis are related to the automatic data crawling (described in Chapter 3), the benefits of the direct user engagement and active data acquisition are also studied, as Chapter 7 reports.

Furthermore, a growing number of initiatives is mixing, nowadays, both passive and active data acquisition approaches.

2.2.1 Active Data Acquisition

The most intuitive way to approach the problem of collecting observations from a set of users is to simply ask them to provide the desired observations. As simple as it may sound, however, this involves two major issues. First, at least two communication channels should be established: one that allows the scientists to communicate with the users and one that allows the users to transmit the observations to the scientists. Second, collecting observations requires time and effort, the users must be somehow engaged in doing so and kept motivated to do it in time.

Table 2.1 reports numerous citizen science works in the literature that collect data for environmental monitoring purposes directly from the users. The aforementioned table aggregates the works by the corresponding environmental field and lists which communication channels and engagement techniques they use.

Table 2.1: *Examples of citizen science and crowdsourcing works that use a website for the collection or display of the data (Website), use social networks to collect the data (SN), have a dedicate mobile application for data collection (Mobile), use gamification techniques (Game) or explicitly describe and discuss the adopted engagement strategy (Engage)*

	Water Level			Noise Pollution			Animal Species		Air Quality	Climate Change	Weather
	[39]	[117]	[122]	[125]	[8]	[124]	[37]	[115]	[130]	[143]	[98]
Website	X		X		X	X			X		X
SN							X	X			
Mobile	X		X	X	X	X	X		X	X	X
Game				X						X	
Engage		X	X				X			X	

Communication Channels

There are several communication channels that are commonly adopted in active crowdsourcing campaigns.

Emails and **Email Lists** are among the oldest Web communication methods. As a matter of fact, they have been used since the earliest citizen science projects [87]. However, emails pose several limitations, such as the impossibility of reaching new users and inefficient data collection. While nowadays mailing lists are still being used [117], they are usually adopted as a non-primary communication method, which assists the communication within an already engaged community.

Almost every citizen science project has a dedicated **Website** and several use it both for the communication towards the users and for the data collection. Project websites bring several advantages, such as an increased project visibility, richer content w.r.t. the mailing lists, universal access from almost any web-connected device and wider possibilities in terms of community building (forums, member dedicated areas, etc.). The observations collected through the websites can be highly customized and structured.

While almost every citizen science project makes a heavy use of the **Social Networks** for communication campaigns, several use them also for the observation collection, as Table 2.1 testifies. The main advantage of collecting the observations through the social networks is the integration: no registration, dedicaded websites or tools are involved, the users are just asked to perform the same actions they are used to: use a hashtag, upload a photograph with a tag or publish a post on the project social page [44].

The most recent and, arguably, the most efficient method for observation collection is the use of dedicated **Mobile Applications**. Not only it allows the users to submit the observations in real-time without postponing the task, but it also provides sensors which are usually not available during normal daily outdoor activities: camera, compass, microphone and GPS sensor. In fact, almost all of the recent citizen science projects offer this solution (Table 2.1). Must be highlighted that, with the rise of Rich Internet Applications (RIAs), websites can be developed to be almost indistinguishable from native mobile applications. For the sake of this discussion we consider these RIAs as mobile apps, and keep the term *website* only for traditional non-RIA architectures.

User Engagement

A perfect “recipe” for a successful engagement campaign does not exist: it is a heavily domain- and case-specific challenge. Several works in the literature overview the engagement techniques and study the success stories of citizen science projects. This include both generic citizen science works [38] and those specifically applied to the environmental monitoring [76, 118].

Luckily, environmental monitoring receives more and more public attention, and is proving itself a field that is well suited for the user engagement. However, the difficulty of engaging the crowd must not be underestimated. The general consensus is that the communication is the key for a successful citizen science campaign: the citizens need to share their findings, but they also need to receive feedback from the scientists. The common conclusion is that the existence of feedback mechanisms is a key issue in any public participation support system since it promotes citizens’ commitment to participate. Moreover, volunteers also need to communicate among themselves to support the organization of monitoring activities [76]. Social networks have been found to have a great power to generate public interest in a citizen science project [167].

The users can be also engaged through **Gamification**. Slightly adapting the definition proposed by Zichermann and Cunningham [188] to our scenario, gamification is a process of game-thinking and game mechanics to engage users in collecting environmental observations. Table 2.1 shows citizen science projects that adopt gamification to engage their users.

2.2.2 Data Crawling

The data crawling approach consists in passively acquiring content that has been publicly published prior to its acquisition. In this scenario the data is provided unintentionally, i.e. the original intent for data publishing was not contributing to an environmental monitoring campaign.

The advantages of this approach usually include the low cost of data acquisition and the large volumes of available data. The major drawback, however, is the lack of the control over the format in which the data is acquired. Social data is usually meant to be consumed by humans, as such, it is highly unstructured: the effort of extracting the necessary knowledge from the data tends to be higher than in the active data acquisition approaches.

The claim that crawling the social web requires low effort may sound controversial, since the desired querying conditions are not always supported by different social sources. However, if the desired querying condition is not supported by the content source, we relax the querying condition until it is supported and enforce the excluded conditions a posteriori, as a processing step.

For example, one may want to retrieve only images taken in Arizona, US that contain California Poppy flowers (a task actually approached in [178]). Although it is unlikely that an API supports flower species filtering, the task can be seen as a simple crawling task “*retrieve all images taken in Arizona, US*” followed by an image processing task “*remove all images that do not contain California Poppy flowers*”. In case the API does not provide a geolocation information, we can see the whole process as a crawling task “*retrieve all images*”, followed by a first processing step “*estimate the location of the*

photo, discard if not Arizona, US" and finally "*remove all images that do not contain California Poppy flowers*".
In this section we deal with the mere crawling of the data that can be easily managed through the supported APIs, while the eventual processing steps are discussed in Section 2.3.
The content that is publicly available on the social web can be generated either by human users or by Internet of Things (IoT) devices. In this section we cover both options, illustrating the main social networks the user-generated content can be acquired from and discussing the potential of the data crawling from the IoT devices (specifically outdoor webcams).

User-Generated Content Sources

Social networks and social media platforms are the major sources of the user-generated content. These platforms are reaching unprecedented volumes of data [104]. The type of the content which can be extracted from these web platforms depends on the nature of the platform itself and usually include one or more of the following: text, images, videos and geographical information.
The choice of the best platform to crawl the data from depends on several factors: which types of content the platform provides, what are the volumes of the content (in terms of spatial and temporal density), the public accessibility of the content and its historical availability. Here we provide a brief overview of the ones that are commonly used in the environmental monitoring field.
We do not discuss the implementation aspects of the crawling, as they heavily depend on a single software engineer preferences and the software architecture being used. Generally, all the platforms discussed here provide a solid and mature API.
Twitter is one of the biggest microblogging web services. It focuses on short text messages eventually accompanied by an image or a short video. Twitter is arguably the most common source of textual user-generated content in the social data mining field. There are indeed good reasons behind this: (i) the real-time nature of the content [151]; (ii) the high volume of the content [94]; (iii) the use of short text messages and hashtags that facilitates the analysis, often avoiding Natural Language Processing operations and allows to deal with tweets in different languages.
Some portion of the tweets is created using mobile devices and carry a geotag (estimated to be approximately 0.4 % of the overall tweets [7]). Furtheremore, an estimated 95 % of the data is not restricted to particular set of users and is publicly available [174]).
The limitations of Twitter, on the other hand, include the partial availability of real-time tweets (2 % of the overall stream for unpaid users) and the one week limit on the historical tweet search.
Facebook is the social network that produces the largest amount of user-generated content in the world [175], but is extremely privacy-concerned. Consequentially, the public API access is available only for the content published on public pages (i.e. accounts for businesses, brands and organizations) and not on personal profiles. Users hardly report environment observational data on organization pages, so Facebook is rarely used for environmental data crawling. The rare exceptions include crawling posts from other citizen science Facebook pages where volunteers publish unstructured environmental

observations [44, 115].

Google Plus policies are friendly towards content crawling (95 % of the data is estimated to be public [48]). Unfortunately, the volume of content on Google Plus is very low. For this reason, nowadays, Google Plus is rarely used in the literature.

Although there are several other sources that are commonly used in the social web data crawling, they are not discussed here because their nature and their content type is not suitable for the environmental monitoring. These sources include article based systems (such as Wikipedia, news websites, blogs and Question&Answer platforms). These sources tend to be updated with less frequency, with high quality content, often produced by several authors. It is difficult to imagine thousands of people rushing to edit a Wikipedia article when their home town is hit by a snowfall, while this exact behavior can be observed on microblogging platforms [100].

Visual Content

"*A picture is worth a thousand words*" - the famous idiom applies also to the world of the social web media. An outdoor photograph attached to a user post can often provide more information than the accompanying text could ever do. The advantages of the photographs w.r.t. to the text messages are the objectivity (messages are subjective, while photographs do not lie), universality (no different languages) and the high level of details. The information provided by a photograph can be rich enough to include details the user did not intend to provide and even was not aware of: you hardly remember the amount, size and pattern of the clouds that you observed a week ago, and you do not tweet messages like "*A beautiful day with a bit of stratocumulus lenticularis clouds!*", however, almost any outdoor photograph you publish on the web does have that information available.

Flickr and **Instagram** are two large photo sharing web platforms, having a user base in the order of 120 M and 600 M users respectively [165]. Instagram, due to its viral and social nature is mainly studied in the social research: improving communication experience [179], discovering online popularity and topic of interest [56], tracing cultural visual rhythms [86]. Flickr, on the other hand, is used mainly to publish photographs to a wide community, and is used by researchers to retrieve information that helps to describe the world and its natural phenomena, such as snow cover [51, 177], vegetation cover [186], animal species monitoring [178] and land type estimation [112].

The social networks discussed earlier (Twitter, Facebook and Google Plus) also allow users to attach photographs to their posts. However, Twitter, Facebook and Instagram follow a strict policy of not preserving the photograph original size and erasing the EXIF information. The EXIF container of a photograph carries important information, such as geotag of the photograph, shooting timestamp, camera model and manufacturer. This information is essential to applications that aim at extracting spatio-temporal environmental information [54, 186]. Google Plus has an image-friendly policy (preserving photograph geotag, EXIF and even the original size if the users allows so), but once again, the volumes of the content are very low. Thus, Flickr is de-facto the main source of the photographs used in the social web data crawling applied to environmental monitoring.

Video content is rarely used in the environmental monitoring, mainly due to the processing difficulty. However, few pioneer works provided recent succesfull proof-of-concept studies, e.g., manually processing Youtube videos to improve flood assessment [127,156]. Video processing techniques are more complex than image processing ones, and, at the best of our knowledge, up to today no work automatically analyses web video content for environmental purposes. The amount of the video content on the web, however, is growing [26] and automatic video analysis is an active topic in the scientific community [172]. We believe that analysis of the web video content can become one of the frontiers of the social web environmental monitoring in next years.

Public Webcams

Humans are not the only providers of public social web content. Some of the web-connected devices that collect and exchange data - collectively called *Internet of Things (IoT)* - publicly upload text, images and videos.

While an IoT device can not be engaged in an active crowdsourcing campaign, the content it produces can be crawled in the same manner as the user-generated one. We focus on a particular case of IoT devices, which are broadly used in environmental monitoring research: **publicly available outdoor webcams**. These webcams are usually installed by privates and organizations (ski resorts, hotels, restaurants, touristic agencies, etc.) as an asset for tourists, meteorological scholars and general public. Webcams usually expose an image, which content is periodically changed (usually with frequency of once every 1' - 15'). The use of the webcam imagery brings several important advantages and disadvantages, and can complement [50] or even substitute [46, 132] the use of user-generated photographs.

The main advantage of the webcam imagery is the fact that, generally, webcams have a static frame: their position does not change, niether does their direction. This poses an enormous advantage for all the applications in which object identification and tracking play an important role: one can assume that a non-moving object maintains its position on all the images, and that any variation of the object on the image is due to the movement of the object itself and not the one of the camera. These assumptions do not hold with user-generated photographs, as their different orientation and position must be considered. However, several works, including the one we present in this thesis, state that an outdoor webcam should not be considered completely static, since the shaking of the webcam can introduce non-negligible variations: this shaking can be neutralized with different techniques of image registration using edge-based (Section 3.2.3) or color-based [46] features. Another advantage of the webcam imagery is the high temporal frequency: a single webcam can produce more images per day than thousands of human photographers.

The disadvantages of the webcams, on the other hand, are:

- Bad quality content: while an uploaded photograph implies the fact that the photographer considered the photograph informative enough, a webcam produces images at a regular frequency regardless of the visibility of the framed objects. A large portion of the images acquired from the outdoor webcams can be unsuitable for the defined purpose due to the illumination conditions, weather conditions and camera malfunctioning: we estimate that 67 % of images acquired from webcams placed in mountain regions in daylight hours lack from sufficient visibility due to the fog, rain and snowfall (Section 3.2.2).

- The location and optical properties of the camera are usually unknown. This penalizes the algorithms that need the position of the camera and optical details, e.g., estimating the Field Of View (FOV) of the camera. The position of each webcam, so, must be determined manually [46] or by automatic outdoor image geolocation algorithms [155].
- Lack of a centralized repository. Until the URL of a public webcam is explicitly provided or mentioned, its existence remains unknown.

Luckily, the last two problems are partially resolved thanks to websites that collect outdoor webcam datasets. They store the URL, tentative GPS position and other relevant information available for every webcam, and provide public API for the webcam search. Two of the webcam datasets that must be mentioned are *EarthCam*[12] and *Webcams.travel*[13] (containing more than 52 k outdoor webcams).

2.3 Data Processing

Once acquired, the content must be processed in order to extract environmentally-relevant spatial, temporal, or spatio-temporal measurements. The design of a social web environmental monitoring campaign is a trade-off between the easiness of the content acquisition and the volume of the content on one side, and the required processing effort on the other.

In this section we discuss the processing techniques that are adopted in the state-of-the-art, based on the content type and the desired result. The described content processing is not related to a specific acquisition type (active data acquisition or passive data crawling), it is common, though, for the passively crawled data to require more processing effort as it tends to be less structured.

2.3.1 Naïve and Manual Processing

Several works do not adopt sophisticated processing methods, mainly because the input data is structured enough to directly provide the desired environmental measurements. Examples of these scenarios include using a Twitter query and considering the daily tweet counts as the output measure [36] or considering every tweet which is geotagged and contains a specific keyword to be an environmental observation of a forest fire [40]. The absence of complex processing algorithms does not necessarily imply the bad quality of the work. The "simple is better" principle often holds, and the absence of processing can be a result of a well performed citizen science campaign or an excellent choice of the public content source [11, 100]. Processing can also be unnecessary if the final goal of the work is the creation of a map or a list of the detected events without their aggregation [32]. Furthermore, the need for processing algorithms often disappears in citizen science scenarious when the input data is already formatted as necessary, for example, if the users deliver final water level measures [43].

Another aspect that must be considered is the desired volume of the data: the less data we are ready to settle with, the less processing effort we will likely need. If we consider our problem to be a binary classification (relevant/non-relevant item), the more relevant

[12] `http://www.earthcam.com`

[13] `http://www.webcams.travel`

data we are ready to “sacrifice” - the lower is the recall we are ready to tollerate, consequentially, the precision of the classifier increases. We can obtain high precision with naïve processing if we are ready to tolerate the low recall. To make an example, assume that we want to analyze the photographs from Twitter containing the California Poppy flower. One possible crawling approach could be to search for tweets that contain images and that include the *#CaliforniaPoppy* hashtag. At the date of writing, *#CaliforniaPoppy* hashtag has been used in 15 tweets during the last week, 7 of which contain an image. All the images depict the California Poppy (this is no surprise, the hashtag is very specific). The price of using this approach, though, is the data volume: we end up with approximately one image per day. If this volume is enough for our purpose - we can avoid any filtering. However, if the volume is not satisfactory, we are forced to explore other scenarios: for example, crawling all tweets with images that include *#flower* hashtag (750 tweets in the last week, at the date of wriring) and devise a complex content based image classifier that retains only California Poppy photographs [178].

Manual data processing is also adopted in the literature [35, 89, 156] when the volume of the data is small. These works usually present proof-of-concept studies, analyzing the overall quality of the data. For example, Michelsen et al. [127] analyze Youtube videos taken in the same location in different moments and manually determine the observed water level, thanks to the graffiti on the cave wall.

2.3.2 Text and Natural Language Processing

Blindly trusting keywords and hashtags can induce ambiguity in the processing step. For example, querying by hashtag *#rain* can sound like a safe way to obtain tweets related to the meteorological event, however, one would end up with more than half of the tweets related to the just-married Korean singer known by his stage name “*Rain*”. Although this example holds only at the moment of the writing, and will not hold in few days, it highlights the volatile, dynamic and unpredictable nature of the social media content. Even when the tweet is actually speaking about rain (meteorological event), there are no guarantees that it is actually a positive observation (“*A lot of #rain here in London today*”, a negative observation (“*Luckily no #rain today*”) or a historical one (“*Unbelievable amount of #rain yesterday*”).

In case the ambiguity of the textual data is considered to be high enough to jeopardize the quality of the results, a relevance classification can be performed. Apart from the simple techniques (such as regular expression rules [32]), machine learning approaches are often used, learning the discriminative tags [186] or the entire space of statistical, keyword and word context features [151].

If a mere relevance classification is not enough, and specific concepts and entities must be extracted from the text, Natural Language Processing (NLP) and ontology algorithms can be used as proved by the authors of [44] and [115], that automatically extract places, dates and names of animal species from observational posts of a Facebook citizen science group.

2.3.3 Data Geolocation

The geolocation of the observations tends to be a binding factor in the works that study the spatial dynamics of the concerned environmental phenomena. This information can be provided by the users during citizen science campaigns, by the social networks or by the geotags contained in the photographs. There are, however, numerous scenarios in which the data is not geolocated and the location must be estimated from the content.

Furthermore, several social sources are characterized by a small portion of the geolocated content. If the amount of geolocated data is not enough - the location of the remaining part of the content must be infered. For example, only 0.5 % of the tweets are estimated to have a valid geolocation [7], thus, working with geolocated tweets only is subject to this strong limitation. In fact, limiting the content to its geotagged subset has been reported as the root cause of failed experiments [32].

Geolocation can be estimated thanks to the content metadata information (e.g. user profile [27] and social relationships [116]) or analysing geographical terms and names [5], tags [159] and keywords [28].

Image processing techniques also apply, in fact, content-based geolocation of the photographs is a hot topic in computer vision literature. Examples include the photograph gelocation through image visual attributes and descriptions [33], scene features and reconstructed 3D geometry [113].

2.3.4 Image Processing and Computer Vision

The processing of the visual content requires the adoption of the image processing and computer vision techniques. The visual content is generally richer than the text, however, the price to pay is the increased processing difficulty. The last years have been characterized by impressive advances in image processing and computer vision fields [166], which resulted in a number of tools that help social media researches to extract knowledge from their visual data.

The most common scenario is to perform a binary content-based classification, in order to decide whether a photograph is a valid observation of some environmental phenomena or not. The classification could infer whether a photograph does or does not contain snow covered areas [177], vegetation [186] or specific flower species [178]. Other use cases include also cloud monitoring [132], plant phenology [77] and air quality [130]. In Section 3.1.2 we propose a binary image classifier that infers whether a photograph contains a relevant mountain slope.

The classification is usually performed using supervised machine learning algorithms (e.g., Random Forest [132], Convolutional Neural Networks [103] and SVM [186] with color, shape and texture visual vocabularies [178]). The groundtruth data can be obtained through manual annotation, crowdsourcing campaigns [50] or using remote sensing (satellite) data [177].

In case a binary photograph classification is not sufficient and a quantitative measure must be assigned to a photograph, pixel-wise and segmentation techniques can be adopted. These approaches are often used in snow and vegetation monitoring, with segmentation tasks respectively identifying the amount of snow/vegetation visible in a single photograph.

Specifically, snow cover estimation in mountain images is the primary use case of this thesis, thus, in the remaining part of this subsection we review the background of mountain image processing and image snow cover identification.

Mountain Image Processing

The problem of understanding the position of mountain photographs w.r.t. the terrain has recently attracted the attention of the research community.
Baboud et al. [4] propose an algorithm for photo-to-terrain alignment based on a Digital Elevation Model (DEM). However, the method is not quantitatively evaluated on a large dataset, and qualitative results are provided only for 28 photographs. The examples reported in the paper reveal a very accurate alignment with the terrain, indicating the use of a high-resolution DEM.
Other works approach a related problem, that is, the estimation of the geographical position of mountain photographs in the absence of geotags by means of content based analysis [3]. However, they do not address how to determine the labels of the mountain peaks. In addition, in some of the examples, the sky-to-terrain segmentation is performed manually, before the photograph is processed by the algorithm.
Liu and Su [119] present an image content search method based on the shape of the skyline. The idea is to match two photographs which contain the same peaks, similarly to landmark search in urban environments. However, labeling of mountain peaks is not supported.
Unlike [4], we provide a quantitative evaluation on a significantly larger dataset and introduce different adjustments in the preprocessing and alignment algorithm, needed when coping with photos taken in diverse weather conditions and in the presence of other objects (trees, mountain slopes in the foreground, etc.). In addition, we adopt a coarse resolution DEM, which is publicly available. Conversely, [3] is based on an extremely precise DEM available only for Switzerland ($swissALTI^{3D}$: 2 m spatial resolution), and it is not obvious how similar results can be achieved in a different area.
In addition, some works propose methods in which human assistance is needed to perform photo-to-terrain alignment [49] [42] [84]. Due to these constraint, the aforementioned works are not suitable to the scenario addressed in this section, in which a very large number of images are collected in uncontrolled conditions.

Image Snow Cover Identification

The idea of using visual ground photography for snow monitoring purpose is not new. However, the state-of-the-art works often adopt a single camera or multiple cameras, purposely positioned and calibrated by the authors.
Farinotti et al. [49] combined melt-out patterns extracted from oblique photography with a temperature index melt model and a simple accumulation model to infer the snow accumulation distribution of a small Swiss Alpine catchment. However, the whole image processing pipeline was completely manual. It included choosing the photographs with the best meteorological and visibility conditions, photo-to-terrain alignment and snow covered area identification.
DeBeer et al. [42] examined the spatial variability in areal depletion of the snow cover over a small alpine cirque of the Canadian Rocky Mountains, by observing oblique terrestrial photography. The images, obtained from a single ad-hoc installed digital high-precision camera, were projected on an extremely precise DEM with 1 m resolution. The orientation parameters were found manually for each image. The pixel level snow classification was obtained by means of a fixed threshold. This was possible because images were taken in short range, so that snow and terrain could be easily

distinguished based on brightness alone.

A similar problem was addressed by Hinkler et al. [84], in which the authors derived snow depletion curves by projecting photographs obtained from a single ad-hoc camera onto the DEM. In this case, though, pixel-level labeling of snow was performed automatically exploiting RGB color components.

Other works adopt multiple cameras, which are positioned by the authors to monitor a specific area of interest. Laffly et al. [106] combined oblique view ground-based pictures together with satellite images to produce a high temporal resolution monitoring of snow cover. The experiments were performed in the basin of a small polar glacier in Norway (10 km^2), with 10 digital cameras each producing 3 images per day. The described method required a manual installation of 2 m $\times$ 2 m orange flags on the snow at regularly spaced intervals to provide artificial reference points for photo-to-satellite matching. The identification of snow covered areas on the images was performed manually.

Garvelmann et al. [64] exploited a network of 45 spatially distributed cameras to obtain measurements of snow depth, albedo and interception in a German mountain range. Even if the results are highly correlated with ground-truth data, the proposed approach required the installation of wooden measurement sticks with alternating bars and plastic boards for compensating the different illumination conditions of each camera.

Other works investigated the benefit of using terrestrial photography for both short-range and far-range views [140]. In case of short-range, analysis measurement sticks were installed in front of the cameras, whereas in the far-range the authors did not identify snow, but simply compared the photographs with simulations of snow distribution.

Floyd et al. [59] monitored the snow accumulation during the rain-on-snow events by means of the acquisition of photographs from cameras designed and positioned ad-hoc. This approach required the installation of measurement sticks within the camera field of view. The analysis was performed on a short-range view, so that a fixed pixel intensity threshold was enough to perform pixel-level snow classification.

The problem of automatically detecting the presence of snow at the pixel-level was addressed in just a few works. As mentioned above, both [42] and [59] perform a simple thresholding of brightness values. However, this is applicable only to short-range views. Full color information was exploited in [84], which proposed a snow index based on a normalized difference between RGB components. Similarly, [153] presented a simple algorithm for pixel-level snow classification based on thresholding the blue color component, in which the threshold is determined automatically based on the statistical analysis of the image histogram. The method produced excellent results (precision above 0.99), but was tested in somewhat controlled conditions, with short-range views without shadows and cloud occlusions. More recently, Rüfenacht et al. [149] proposed a method based on Gaussiam-Mixture-Model (GMM) clustering of RGB pixel values, designed to work for long-range images of mountain slopes. All these methods ([84], [153], [149]) are included in the experimental evaluation in Chapter 5.

2.3.5 Data Aggregation

Once the observations are processed, they must be aggregated together in order to produce a homogeneous spatial, temporal or spatio-temporal trend. The aggregation and the interpolation of the data is necessary for several reasons: reaching consensus if more observations refer to the same spatio-temporal item; providing an estimation for missing data; fixing the erroneously predicted data.

Geospatial interpolation is a broad topic with a lot of relevant literature [171]. The techniques adopted in the works that analyze the social web for the environmental monitoring purposes vary from simple voting [186] and Kriging [132] up to applying a Kalman filter [151] (considering every observation as a single sensor) or even training machine learning models, that given a histogram of the confidences of the corresponding observations produce a final decision on the phenomena [177].

Sometimes, instead of a spatio-temporal aggregation, spatial and temporal aggregations are performed separately [40]. Other examples do not perform any aggregation when a simple map or list of all the observations is desired [32].

2.4 Evaluation and Experimental Settings

A solid experimental setting is crucial to any robust scientific work. In this section we describe how the works that extract data from the social web for environmental purposes evaluate their performance. Furthermore, we describe the novel evaluation approach that we propose in this thesis, which goes beyond the assessment of the data correctness and actually assesses the data usefulness.

2.4.1 Input Validation

Prior to the design of the processing steps, the input data should be validated. This involves assessing the fact that the input is available, correct and sufficient. While such assessment is often implicitly took for granted (the fact that the results are good implies that the input was also good), some works focus on this point explicitly.

For example, Hyvärinen and Saltikoff study whether visual social media content can be used for the monitoring of the meteorological events, such as snowfall, rain and hail [89]. The goal is not to prove that the social data is useful, but to prove that it has the potential for being useful. Among other evaluations the authors show two photographs retrieved from Flickr that contain hail on the ground and compare them with official meteorological observations, concluding that Flickr photographs are able to detect hail. While such research question may seem silly to some data scientists (e.g., "*Is it not obvious that geotagged Flickr photograph depicting hail corresponds to an actual hail?*"), the authors answer to several questions that should not be taken for granted, such as whether Flickr photographs are timestamped precisely enough and whether they are geolocated with a sufficient accuracy.

Other examples study whether the available volume of the input data is enough, for example, analyzing the temporal trend of tweets related to particular vegetation species in order to assess the feasibility of using Twitter to monitor these species [36].

Such works are usually specific to a precise use case and act as the pioneers, encouraging the future research.

2.4.2 Processing Accuracy

Section 2.3 describes a wide range of processing techniques that can be adopted. These processing techniques are meant to automatically extract some knowledge from the input social media, that would be hard, costly or impossible to extract manually. The success of the environmental monitoring approach strictly depends on the ability of the underlying processing methods to perform their job, and the performance of these methods should be properly assessed.

These evaluations are agnostic w.r.t. the environmental nature of the overall approach and are objective, since they report the capability of the methods to obtain accurate results, where the accuracy of the result is an objective fact. For example: recognizing which plant species a Facebook post is talking about [115], identifying similar tree species from a set of photographs [110], identifying which photographs [177] or even which pixels of a photograph correspond to snow (Chapter 5) or vegetation [108] scenarios.

However, the fact that the extracted knowledge corresponds faithfully to the one depicted in the social media does not imply that this knowledge is correct in terms of the real world environmental phenomena. For example, Chapter 5 proves that the approach it proposes is able (most of the times) to detect whether a photograph does or does not contain snow. What it does not prove, though, is that the same approach is able to identify snow presence in a certain geographical region at a certain time: this is proved by experiments proposed in Chapter 6.

2.4.3 Result Correctness

The next step in the evaluation ladder is to actually assess the capacity of the proposed methods to obtain specific spatial or temporal environmental knowledge that is objectively correct. For example, while presenting an approach for the continental-scale cloud map identification from public webcams [132], the authors prove that the cloud maps they obtain are *correct*. The correctness is measured against an authoritative source: cloud maps obtained from remote sensing (satellite) data.

The comparison with the remote sensing is very common in the estimation of large geographical scale environmental phenomena. Beside the cloud presence, the satellites provide a wide range of maps that can be used as groundtruth: snow [178] cover, vegetation cover [177] and even fires [11]. Other types of authoritative data to compare the results against include meteorological data, such as maps of weather events [32], snowfall and rainfall statistics [100] or snow depth automatic measurement stations [50].

2.4.4 Result Usefulness

The assessment of the result correctness described in Section 2.4.3 is - de facto - the state-of-the-art evaluation, required for a work to be considered a solid study of the social web content applied to environmental monitoring. However, these evaluations objectively prove the capacity of the proposed techniques to replicate the authoritative data, but do not prove the utility of doing so.

This phenomena puts the researchers in a vicious circle: the environmental monitoring community wants a proof that the proposed data is useful w.r.t. the existing sources, so it must be **novel** data not available from authoritative sources; however, proving that the proposed data is correct requires a comparison with an already existing ground truth.

A mitigation of this problem is one of the major contributions of this thesis, as described in Chapter 6. The key idea is to adopt a data-driven environmental model that, among other inputs, relies on the authoritative (government) inputs. Such model must have a well defined performance metric, and the assessment of the usefulness of the results is performed by testing how the performance of the model varies when the input is complemented with the environmental results obtained by social media processing pipelines.

CHAPTER 3

Mountain Image Acquisition

In this chapter we describe the acquisition of visual content depicting mountain landscapes, which can be obtained from two different sources: user-generated photographs posted on social media and image feeds from outdoor webcams. These sources have complementary characteristics. On the one hand, photographs are taken from different locations, possibly capturing different views of the same mountain peak, but their density varies significantly depending on the location (with higher spatial density near popular touristic destinations) and time of the year (with higher temporal density during holidays). On the other hand, webcams capture the very same view at a high temporal resolution. Although webcams are far more numerous than ground-based stations, they monitor a specific location and do not extensively cover large areas.
Due to the distinct characteristics of photographs and webcams, we address them separately, designing two visual content processing pipelines tailored to the specific challenges posed by each source. To this end, we identify and retain for further analysis only those images depicting a mountainous landscape taken in good weather conditions (i.e., without occlusions due to clouds), for which it is possible to determine the location and the pose of the shot, so as to automatically identify the positioning of the image w.r.t. the terrain.

3.1 User-Generated Photographs

Flickr was selected as the data source for user-generated photographs, because it contains a large number of publicly available images, many of which have an associated geotag (GPS latitude and longitude position saved in the EXIF container of the photograph). Furthermore, differently from the others popular social networks and image hosting platforms (i.e. Facebook, Twitter and Instagram), Flickr conserves the original resolution of the photographs and does not wipe out the information carried in the EXIF container of the file. Both the resolution and the EXIF data (such as focal camera length, model and manufacturer) are necessary for a successful photograph geo-registration as described in Chapter 4.

Specifically, we crawled a 300 km × 160 km region across the Italian and Swiss Alps (in the area of Pennine Alps, Lepontine Alps, Rhaetian Alps and Lombard Prealps, approximately from *45.6N,6.7E* to *47.1N,10.7E*).

3.1.1 Crawling Photographs

The Flickr API allows to query the service using temporal and spatial filters. However, each query is limited to return a maximum of 4 k records. The algorithm is designed to start from the whole region of interest and recursively split it into subregions and then perform separate queries. This is performed until the sub-regions have an image count (information provided by the API) lower than the maximum allowed, so as to retrieve all the publicly available images in the desired area.

The crawler was implemented as a stack containing regions to be processed. The processing of a region consists in either splitting the region in subregions and insert them onto the stack, or downloading the list of all available photographs and scheduling them for the relevance classification. The stack is stored in the persistent memory, so if the process restarts due to a failure - the crawling resumes from the last completed operation. The crawling is perpetual and incremental in time, at each iteration the photographs that were uploaded since the last crawling iteration are retrieved. Even if the system is down for an extended period - the first crawling cycle retrieves all the pending photographs.

To understand the content of the crawled data we performed a study on all photographs with a valid geotag within the described region in the temporal window between January 2010 and July 2014. This resulted in approximately 600 k photographs. The first qualitative analysis of the photographs clearly showed that the portion of the positive photographs (i.e. relevant for our purpose, containing a mountain slope) was extremely low. Performing well (both in terms of precision and recall) on such an unbalanced dataset would have required a classifier to have unrealistically high performance. We observed that the negative (non-mountain) photographs consisting of indoor and short-range outdoor photographs were located mainly in the cities and villages. Thus, based on the intuition that a higher elevation implies higher probability of a photograph to contain mountains, we performed a study on how the terrain elevation of the shooting location influences the ratio between positive and negative photographs.

Several online elevation APIs (such as Google Elevation) exist, however, performing online queries for each potential photograph would significantly increase crawling latency. For this reason the crawler relied on an offline Digital Elevation Model (DEM), namely SRTM3 Global[1], reducing the elevation estimation time to a single access to the main memory. The DEM can be interpreted as a regular grid covering the Earth surface that provides the terrain elevation in each point of such grid.

In order to extract the elevation statistics, 6 940 randomly selected photographs from those taken above 500 m elevation were processed in a crowdsourcing experiment[2],

[1] `www2.jpl.nasa.gov/srtm`

[2] Using Microtask platform - `http://microtask.com`

designed to collect three labels for each photograph. Specifically, each annotator was asked to label each image by answering to the following question: “Does this image contain a meaningful skyline of a mountain landscape?”. Since the concept of a meaningful skyline could be ambiguous, we clarified the expected outcome of the task providing a tutorial with some selected images representing both positive (mountain) and negative (small hills and other no-mountain) samples. The total of 20 820 annotations were collected using an internal unpaid crowd. The labeling task required approximately 1” per image. The aggregated label was then obtained by means of majority voting. The results of the crowdsourcing experiment are reported in Table 3.1. Approximately 23 % (3/3 and 2/3) of the images were classified as positive. Note that in almost 13 % of the cases there was not full agreement among workers, due to the subjective nature of the task. Figure 3.1 illustrates the number of positive/negative images for each elevation range. Approximately 50 % of the images taken above 2000 m represent mountain landscapes and the number of negatives rapidly grows below 600 m. Hence, we kept 600 m as the elevation threshold for the new images - all images under such threshold were discarded. Specifically, 237 k of the originally crawled 600 k images were retained.

In future, we plan to investigate whether, depending on the environmental use case, other environmental variables (e.g. climate or ecological data, soil occupation) can be more suitable for the photograph filtering.

Table 3.1: *Aggregated outcomes of the photograph classification crowdsourcing experiment*

Outcome	Count
3/3 positive	1 184 (17 %)
2/3 positive	422 (6 %)
2/3 negative	483 (7 %)
3/3 negative	4 851 (70 %)

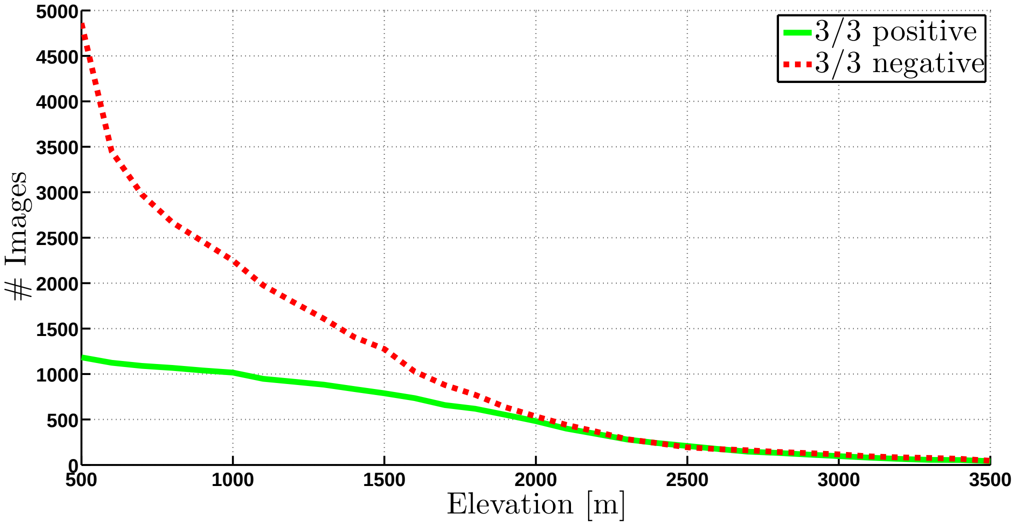


Figure 3.1: *Histogram of the number of positive/negative samples at different elevations.*

3.1.2 Photograph Relevance Classification

Once acquired, the relevance of every photograph must be estimated and the non-relevant photographs discarded. We devised a binary image classifier and studied the performance of a wide set of image descriptors and different feature encoding techniques.

A fixed-dimensional feature vector, which summarizes the visual content, was computed for every image in the dataset. Following the experiments proposed by Xiao et al. [182] we evaluated the classifier performance with several local and global feature descriptors. The feature vector was then fed to a Support Vector Machine (SVM) classifier and its performance was tested. The features descriptors are listed below.

Dense SIFT: we extracted SIFT descriptors from color images by testing different color models at different scales $\{\frac{2}{3}, 1, \frac{4}{3}, \frac{5}{3}\}$, sampled from a uniformly spaced grid with step size equal to 6 × 6 pixels, obtaining around 10^5 descriptors for each image (the exact number depends on the image resolution). The descriptors for RGB, HSV and opponent color models were obtained as the concatenation of the SIFT descriptors of each color channel. The Bag-of-Visual-Word (BoVW) [183] model was adopted, encoding the feature vector as a histogram of visual words with a dictionary determined during an offline training phase. Specifically, $10^2 \cdot V$ SIFT descriptors were randomly sampled from 100 randomly selected images, where V denotes the number of visual words in the dictionary. The dictionary was learned using k-means, with $k = V$.

HOG2x2: as with Dense SIFT, Histogram of Oriented Edges (HOG) descriptors were densely extracted, computing a histogram of oriented gradients in each 8 × 8 pixels cell and normalizing the result using a block-wise pattern (with 2 × 2 square HOG blocks for normalization). We adopted UoCTTI HOG variant [55]. Similarly to Dense SIFT, the BoVW model was adopted.

SSIM: self-similarity descriptors [160] were computed on a regular grid at 5 × 5 pixels step. Each descriptor was obtained as a correlation map of a patch of 5 × 5 in a window with radius equal to 40 pixels, quantified in 3 radial bins and 10 angular bins. Similarly to Dense SIFT, the BoVW model was adopted.

GIST: the GIST descriptor [137] was computed as a wavelet image decomposition (each image location is represented by the output of filters tuned to different orientations and scales). We adopted the parameter setting proposed in [182]. The result was a global image descriptor of 512 dimensions.

CNN: furthermore, we used a pretrained convolutional network for large-scale visual recognition from [163]. The feature vector was defined as a concatenation of the output of the last convolution layer and the vector containing all final label scores (a vector of 1 000 probabilities of the photograph to belong to a certain dataset label, such as *race car*, *volcano*, *mountain tent*, etc.).

We also explored the effects of replacing the BoVW model with the Fisher Vector encoding as described in [154]. We studied the Fisher Vector encoding applied with different number of Gaussians, and with/without Principal Component Analysis (PCA).

In order to capture spatial clues, we adopted the spatial histogram approach proposed by [78] and [109]. In addition to computing a Dense SIFT, HOG2x2 or SSIM V-dimensional histogram for the whole image, we also split the image in three equally sized horizontal tiles, and computed a V-dimensional histogram for each tile. Each of the four histograms (total and three tiles) was L_1-normalized and then stacked to form a $4V$-dimensional vector, which was L_2-normalized. The choice of horizontal tiles was driven by the intuition that a generic mountain image is horizontally-symmetric (no difference between right and left), while vertical information is specific: there is usually the sky in the upper part and the terrain in the lower part of the image. Analogously to spatial histogram approach for local features, GIST descriptors were extracted from the whole image and three images representing equally sized horizontal tiles, then concatenated. Figure 3.2 shows an example of the concatenation of 4 V-dimensional vectors. A similar technique was applied in case of the Fisher Vector, concatenating four encodings (all image features and one for each horizontal tile).

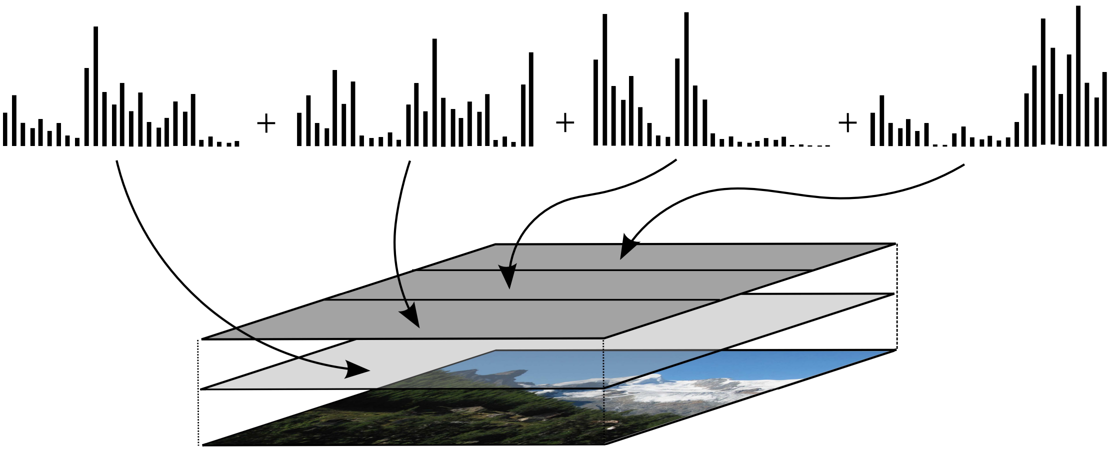


Figure 3.2: *An example of the adopted spatial histogram approach concatenating 4 feature vectors.*

Experiments

The feature vectors were fed to a SVM classifier using a χ^2 kernel. In order to create a balanced dataset, we retained all the 1 184 positive samples (voted 3/3) and randomly selected the same number of negative samples. Then, we used random 1 658 samples ($\sim$70 %) for training and validation and 710 samples ($\sim$30 %) for testing. In order to learn the optimal values of the parameters of the SVM classifier, we adopted k-fold cross validation, with $k = 5$. Thus, the set of labelled samples for training and validation was split in k disjoint sets. At each iteration, one set was used for validation, while the others were used for training. We performed a grid search to seek the optimal hyper-parameters C and γ of the kernel, each parameter in the set $\{0.01, 0.033, 0.066, 0.1, 0.33, \ldots, 10, 33, 66, 100\}$.

Table 3.2 summarizes the results obtained within the test set, by all listed feature extractors, with the best configuration in terms of C and γ of the SVM kernel. Performance was measured using accuracy, defined as the fraction of samples for which the classifier provides the correct label. For completeness, Table 3.2 also shows the values of precision and recall. HOG2x2 obtains similar performance to Dense SIFT; both Dense SIFT and HOG2x2 slightly outperform SSIM. All three local feature descriptors (Dense SIFT, HOG2x2, SSIM) perform better than GIST. The poor performance obtained by the pre-trained CNN can be justified by the fact that the definition of the positive image is very restrictive, indeed, annotators were asked to label as positive only images with "*meaningful skyline of a mountain landscape*", as only such images are relevant in out use case. We believe that the CNN pretrained on one thousand real world classes was not able to distinguish well between scenarios like a far placed mountain skyline (positive in our dataset) or a close view of a hill (negative in our dataset).

Table 3.3 shows the detailed results obtained by Dense SIFT (as the best performing feature) within the test set, for different sizes of the dictionary $V \in \{2\ 500, 5\ 000\}$ and for each color model. In all cases, we obtained very good results, with the highest value of accuracy (above 95 %) achieved by using the RGB color model, regardless of the number of visual words adopted. We also computed the learning curves indicating the accuracy for both the training and the test set, to exclude overfitting and verify that no additional gains could be expected by further increasing the size of the training set. In addition, we investigated the use of different vocabulary sizes (namely $V = 1\ 000$ and $V = 10\ 000$), which did not improve the accuracy. Furthermore, we investigated the effect of the replacement of the BoVW model with the Fisher Vector encoding. We used different number of Gaussians for the Fisher Vector (namely 16 and 128), with and without applying PCA. None of the configurations of the Fisher Vector improved the accuracy.

Finally, the images that are classified as positive are passed to the next step of the pipeline, to register the image w.r.t. the terrain. This phase is described in Section 4.1.

Table 3.2: *Results obtained by different feature extractors for the photograph relevance classification problem (mountain vs. non-mountain).*

Feature	C	γ	Accuracy	Precision	Recall
Dense SIFT	3.3	0.66	**95.1**	**94.0**	**96.3**
HOG2x2	3.3	0.033	94.7	93.9	95.5
SSIM	0.66	0.33	93.0	92.5	93.5
GIST	0.33	1	87.61	82.64	95.21
CNN	3.3	0.1	80.0	72.8	95.8

Table 3.3: *Results obtained by Dense SIFT for the photograph relevance classification problem (mountain vs. non-mountain).*

Color Model	V	C	γ	Accuracy	Precision	Recall
gray	2500	3.3	0.1	93.6	91.9	95.8
gray	5000	1	0.33	94.4	93.6	95.2
RGB	2500	0.33	0.01	**95.1**	**94.7**	95.5
RGB	5000	3.3	0.66	**95.1**	94.0	96.3
HSV	2500	33	0.66	94.2	92.0	**96.9**
HSV	5000	6.6	1	94.1	92.9	95.5
opponent	2500	0.66	0.66	94.0	92.0	96.6
opponent	5000	1	0.33	93.2	90.7	96.3

3.1.3 Crawler Web GUI

We developed an internal web platform to facilitate the debug, qualitative analysis and system health monitoring of the crawler component. The platform allows to analyze the spatial and temporal distributions of the crawled photographs (both classified as positive and negative), provides a heatmap GUI and a timeline histogram with spatial and temporal boundary filters. For every query the resulting images can be visualized. Furthermore, the platform acts also as a proxy for the experimental runs: given a set of parameters (such as the dataset, feature set, SVM, BoVW and Fisher Vector configuration variables) it runs the tests and produces the detailed report that includes the algorithm performance, error metrics, plots and ROC curves. Figure 3.3 shows several screenshots of the platform.

3.2 Public Outdoor Webcams

Outdoor webcams represent an additional valuable source of visual content that can be exploited to monitor snow cover. The use of selected webcams that point to mountain landscapes poses different advantages and disadvantages with respect to user-generated photographs. On the one hand, the images captured by a webcam do not need to go through the relevance classification pipeline described in Section 3.1.2. In addition, most webcams capture images every 1 to 15 minutes, thus ensuring a **very high temporal density**. On the other hand, the **spatial density is lower** than the one of user generated photographs, because the deployment and maintenance of a webcam is more time consuming w.r.t. publishing a photograph.

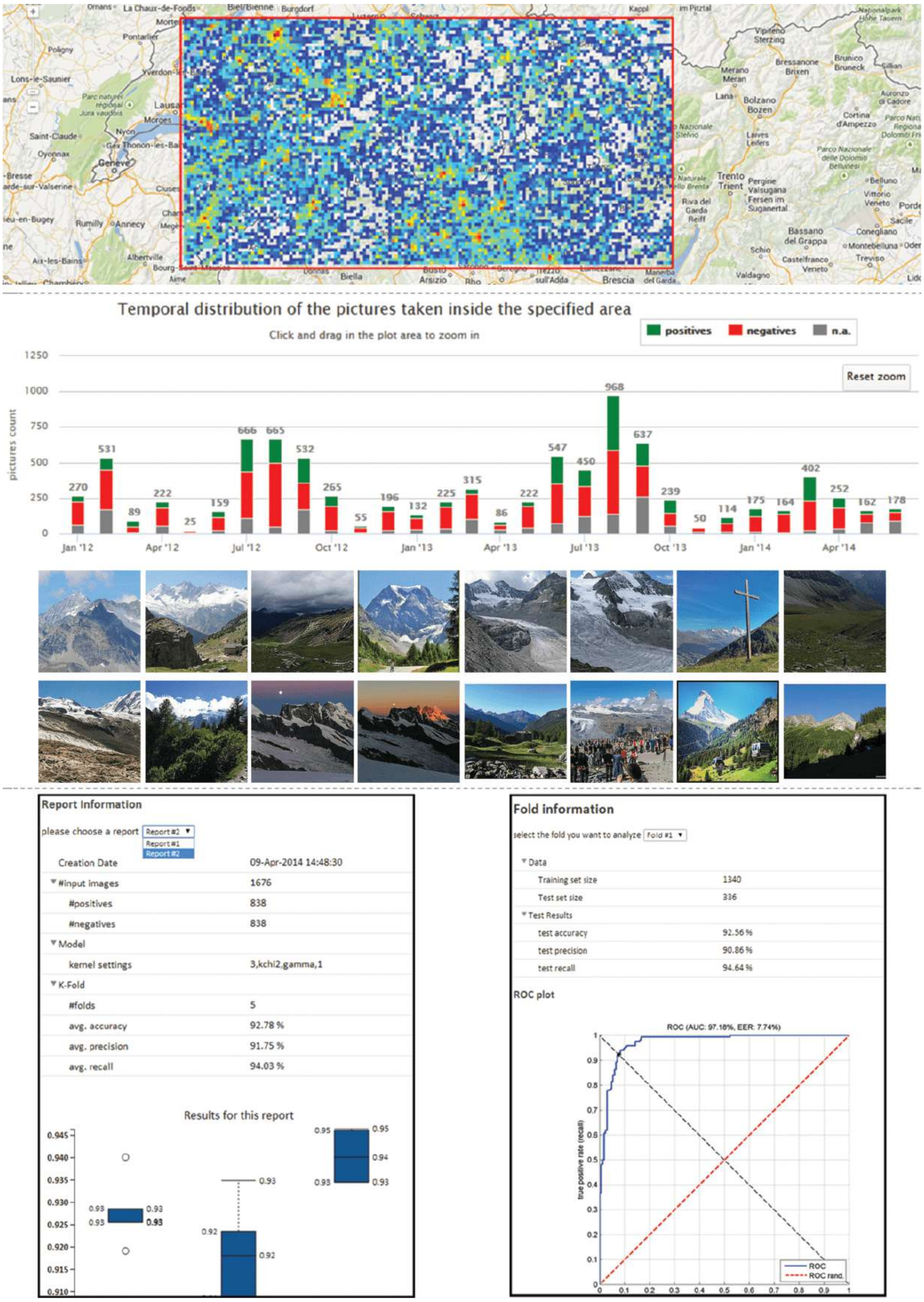
3.2.1 Crawling Mountain Webcams

In order to facilitate the integration of webcams into web pages, a public webcam usually exposes a URL which returns the most recent available image. Furthermore, webcam web servers tend not to respect standard HTML headers such as *Last-Modified*. From implementation perspective, the webcam crawler loads the list of all the webcams in the dataset at the boot and starts asynchronous infinite loops, one for each webcam. Each loop iteration checks the corresponding webcam image and adds the image to the dataset if it is changed w.r.t. the previous iteration, then idles for $1'$ and starts over again. Since downloading the entire image to check a webcam every minute requires unfeasible bandwidth for a single server - the new/old image check is performed only on a portion of the image. Namely, only the first 5KB of the image are downloaded, hashed and compared to the previous webcam hash: if the hash is different, it is saved as the new hash and the rest of the image is downloaded. Furthermore, after the crawler boots, the first image acquired from every webcam is discarded, as there are no guarantees on its timestamp (some webcams, due to failures, propose the same images for days or months).

Populating Webcam List

The first version of the webcam list included $\sim$ 100 webcams manually found through search engines using relevant queries and keywords. Then we used *webcams.travel*[3] - the largest webcam directory containing more than 60 k webcams worldwide - and queried for all the webcams in the Alpine area thorough the public API, resulting in $\sim$ 3.3 k webcams. We set up a crowdsourcing experiment where annotators were proposed several images from a single webcam (to mitigate the fact that some of the images could have been affected by low visibility) and asked to classify if the webcam

[3] http://webcams.travel



Temporal distribution of the pictures taken inside the specified area
Click and drag in the plot area to zoom in
🟩 positives 🟥 negatives ▨ n.a.
Reset zoom
pictures count
1250
1000
750
500
250
0
270 531 89 222 25 159 666 665 532 265 55 196 132 225 315 86 222 547 450 968 637 239 50 114 175 164 402 252 162 178
Jan '12 Apr '12 Jul '12 Oct '12 Jan '13 Apr '13 Jul '13 Oct '13 Jan '14 Apr '14

Report Information

please choose a report Report #2 ▾
Report #1
Report #2

Creation Date	09-Apr-2014 14:48:30
▾ #input images	1676
#positives	838
#negatives	838
▾ Model	
kernel settings	3,kchi2,gamma,1
▾ K-Fold	
#folds	5
avg. accuracy	92.78 %
avg. precision	91.75 %
avg. recall	94.03 %

Fold information

select the fold you want to analyze Fold #1 ▾

▾ Data	
Training set size	1340
Test set size	336
▾ Test Results	
test accuracy	92.56 %
test precision	90.86 %
test recall	94.64 %

ROC plot

Figure 3.3: *Screenshots of the internal crawler web portal, including the spatial heatmap of photograph distribution (top); temporal distributions of positive, negative and discarded photographs with corresponding examples (middle); reports of the conducted classification experiments (bottom).*

was framing mountains or not. Out of these $\sim$ 1.8 k webcams were classified as positive (contain mountains) and added to the webcam crawler. Figure 3.4 shows the map of all the webcams and the corresponding positions in our dataset.

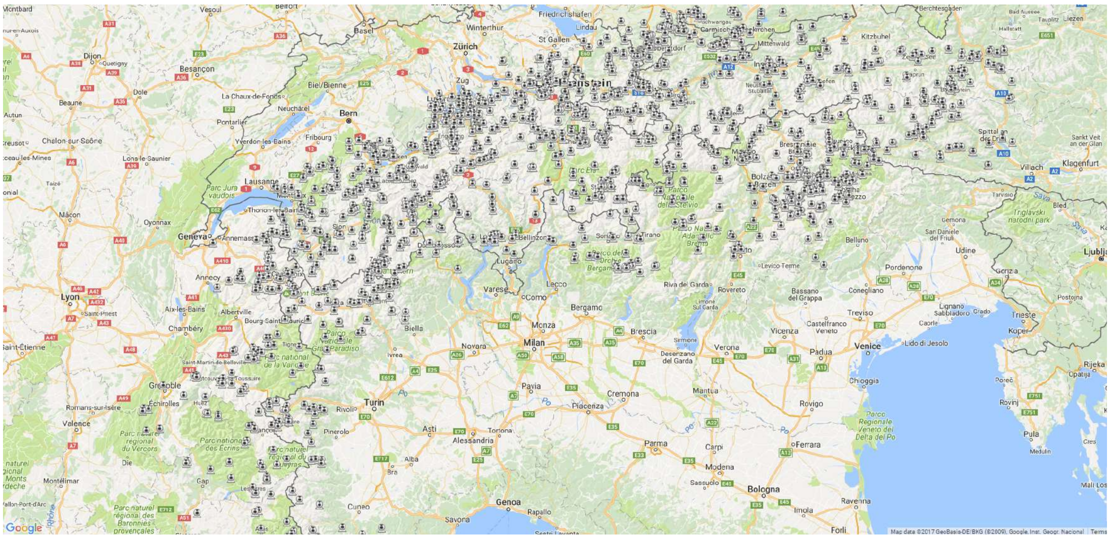


Figure 3.4: *Map of all the webcams in our dataset.*

3.2.2 Bad Weather Filtering

Due to bad weather conditions that significantly affect short- and long-range visibility (e.g., clouds, heavy rains and snowfalls), only a fraction of the images can be exploited as a reliable source of information for estimating snow cover. In this respect, we manually screened $1\ 000$ images crawled from 4 webcams (Valmalenco - Italy, Bormio - Italy, Metschalp - Switzerland, Hohsaas - Switzerland) in daytime hours (9:00 - 18:00) and we observed that $67\ \%$ of them were not suitable for further analysis due to insufficient visibility.

Therefore, we devised a simple algorithm that automatically filters out those images acquired during bad weather conditions. The key assumption is that, when visibility is sufficiently good, the skyline of the mountain profile is not occluded. For each webcam, we create a binary mask L with the same size of the acquired image. Such binary mask indicates those pixels $p = (x, y)$ that are in the neighborhood of the skyline. Hence

$$L(p) = \begin{cases} 1 & \text{if } \exists r \in \mathcal{L} : \|p - r\| \leq \tau \\ 0 & \text{otherwise} \end{cases},$$

where $\mathcal{L}$ denotes the set of pixels that belong to the skyline, $\|\cdot\|$ computes the Euclidean norm. We empirically set $\tau = 0.04\, h$, where h denotes the height of the image in pixels. Then, for each image acquired by a webcam, we compute its edge map E and we binarize the result. We define a function $f(\cdot)$ that, given an image, returns the number of columns that contain at least one non-zero entry, and the skyline visibility score as

$$v = \frac{f(E \cdot L)}{f(L)},$$

where $\cdot$ denotes the pixel-wise product between two images of the same size. The value of v is in the interval $[0, 1]$ and can be intuitively interpreted as the fraction of the whole skyline that is visible in a given image. We retain for further processing only those images for which $v \geq \bar{v}$, where $\bar{v}$ is a threshold, which was set to 0.75, based on the experiments illustrated below. The proposed method retains images in which clouds do not occlude the skyline, although they might still be present and interfere with estimating the snow cover. However, transient clouds are handled and removed by the method described Section 3.2.3.

Experiments

In order to build a reliable test dataset, we manually labeled 1 000 images collected from 4 webcams. Each image was manually tagged as "good weather", if the entire mountain area was visible and not occluded by clouds, or as "bad weather" otherwise. The classifier was evaluated using a ROC curve, which shows the True Positive Rate (TPR) vs. the False Positive Rate (FPR), illustrated in Figure 3.5. The temporal frequency of the webcam image acquisition is high, so a large number of images is available. Hence, the choice of the threshold parameter $\bar{v}$ was driven by the goal of having low FPR. Namely, $\bar{v}$ was set to 0.75 (corresponding to the point marked in Figure 3.5), obtaining a TPR equal to 87.4 % at FPR 3.5 %.

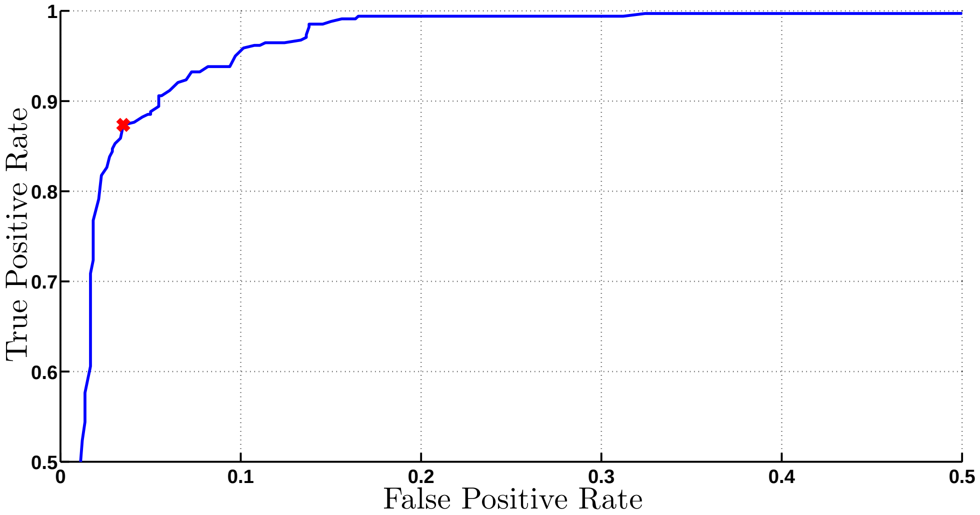


Figure 3.5: *The ROC curve of the webcam image weather classifier when varying the threshold $\bar{v}$ (positive stands for good weather).*

3.2.3 Aggregating Daily Images

Good weather images might suffer from challenging illumination conditions (such as solar glares and shadows) and moving obstacles (such as clouds and persons in front of the webcam). At the same time, snow cover changes slowly over time, so that one measurement per day is sufficient. Therefore, we aggregated the images collected by a webcam in a day, to obtain a single representative image to be used for further analysis. We adopted a simple median aggregation algorithm, which can deal with images taken in different conditions, removing transient occlusions and glares. Given

N good weather daily images $I_1, \ldots, I_N$, we define the Daily Median Image (DMI) as

$$DMI(x, y) = med\{I_1(x, y), I_2(x, y), \ldots, I_N(x, y)\},$$

where $med\{\cdot\}$ denotes the median operator, which is applied along the temporal dimension. Figure 3.6 shows an example of a DMI generated by aggregating 11 images. The aggregation attenuates the different illumination conditions and removes the persons standing in front of the webcam partially covering the mountain.

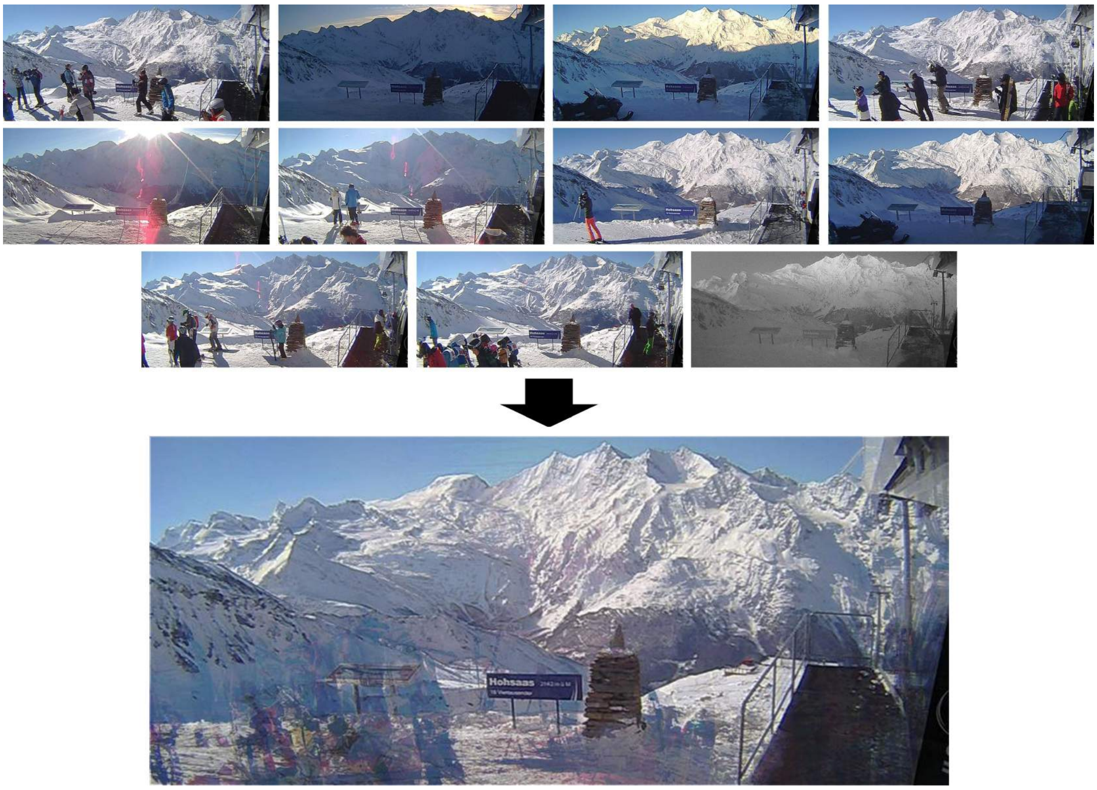


Figure 3.6: *An example of a Daily Median Image (bottom) performed on 11 daily images (top).*

A challenging factor in the aggregation of the daily images lies in the fact that it is common for the webcam orientation to slightly vary during the day. This phenomenon might occur due to strong winds. The DMI of a webcam suffering from temporal jittering results in a blurry image, unsuitable for further analysis. To handle this issue, we performed image registration with respect to the reference frame of the first image. A global offset is computed by means of the cross-correlation between the two skyline edge maps. Each image is compensated by this offset before computing the DMI. Figure 3.7 shows an example DMI obtained without (top) and with image registration (bottom).

3.3 Dataset Spatio-Temporal Analysis

We analyzed the user-generated photographs and webcam images in order to get insights on the spatio-temporal distribution of our datasets. Given the 300 km × 160 km region the photographs were crawled from, we split it in a 5 km × 5 km step grid.

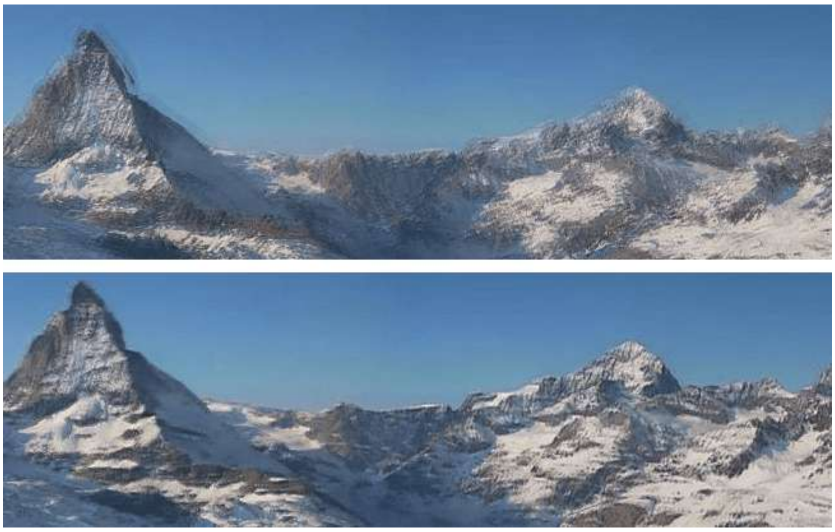


Figure 3.7: *An example of a Daily Median Image (DMI) performed without (top) and with image registration (bottom).*

We analyzed all the photographs and the images from all the webcams in our dataset that are placed in the same region acquired in a 6 months period (from December 1st, 2014 to May 31, 2015). We define *spatial coverage* as the fraction of the grid cells that do contain ad least 1 image in the whole period, and *temporal frequency* as the average number of images contained in a non-empty grid cell in the observation period. Table 3.4 reports the results obtained for both photographs and webcam images. The results show that photographs have better spatial coverage, whereas webcams have lower spatial coverage and much higher temporal frequency.

Table 3.4: *Spatio-temporal photographs and webcam images distribution.*

	Spatial Coverage	Temporal Frequency
Photographs	38 %	$\sim$ 10
Webcam Images	19 %	$\sim 10^4$

The temporal frequency of the user-generated photographs was very low: an average 25 km^2 region that has at least one photograph produces 1 - 2 photographs per month. Clearly, this is insufficient for the large scale environmental analysis: in our use case we aim at monitoring the snow cover at daily level.

Given the unsatisfactory amount of user-generated photographs, the environmental experiments performed in this thesis (Chapter 6) use various webcams, but not photographs. However, all the processing algorithms that are presented in the next chapters (Chapter 4, Chapter 5) are perfectly suitable both for photographs and webcam images. In fact, such algorithms are evaluated on webcam images as well as on Flickr photographs. We argue that, since the volume of user-generated photographs on the web is growing rapidly [102], the proposed techniques could be applied also to user-generated photographs with successful environmental impact in the near future.

CHAPTER 4

Mountain Image Geo-Registration

The distance between the photograph shooting location and the framed mountains can be very high, reaching easily tens of kilometers. Thus, the photograph geotag only is not sufficient for the analysis of the depicted mountains. We need to understand which portions of the image represent which mountains, ideally, identify the geographical correspondence of each pixel of the image: estimate whether it is a terrain surface or sky, what is the corresponding geographical area, what are its GPS coordinates, altitude and distance from the observer. In computer vision, *image registration* is the process of transforming different sets of data into one coordinate system. We call our process *geo-registration*, i.e. transforming the photograph pixel coordinates into the real-world geographical coordinates. State-of-the-art review specific to the mountain image processing and geo-registration is proposed in Section 2.3.4.

Given a terrain model the photograph should be geo-registered with, from geometrical perspective, three properties of a photograph must be estimated in order to understand its position w.r.t. the real world: the **shooting location** (latitude, longitude and altitude), the **direction** of the photograph (i.e. the orientation of the camera during the shot) and the **size** of the photograph expressed in real-world units. The alignment (i.e. photo-to-terrain position estimation) is the key element for a successful geo-registration. Figure 4.1 shows an example of such alignment.

Section 4.1 describes a heuristic algorithm for the photograph-to-terrain alignment that is suitable for offline processing. Then, Section 4.2 proposes the supervised learning variant of the former approach that can be performed in real-time on low power mobile devices.

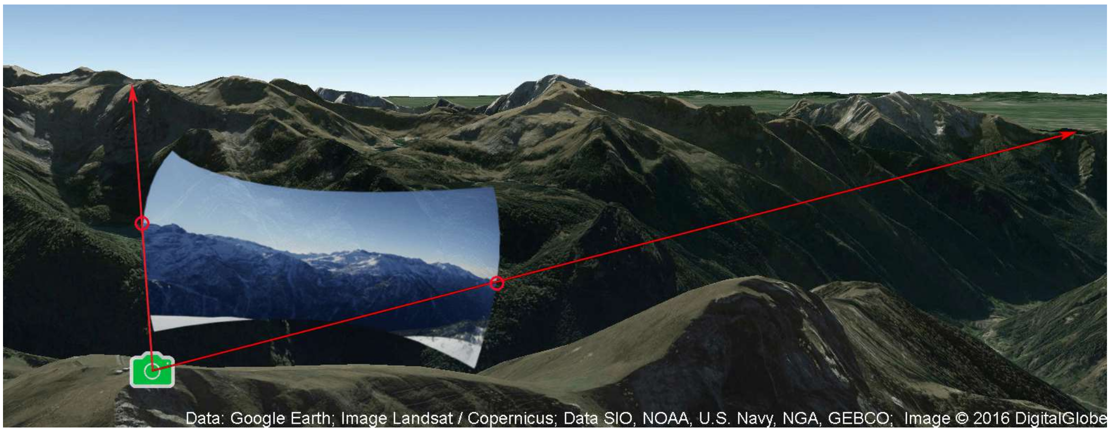

Data: Google Earth; Image Landsat / Copernicus; Data SIO, NOAA, U.S. Navy, NGA, GEBCO; Image © 2016 DigitalGlobe

Figure 4.1: *An example of a mountain photograph-to-terrain alignment. The green icon indicates the shooting location, red lines indicate the projection of the photograph boundaries.*

4.1 Heuristic Photo-to-Terrain Alignment Approach

The latitude and longitude of the photograph acquisition location are known since we retain only images accompanied by a geotag. The altitude is estimated to be the terrain elevation at the geotag location. This implies the assumption that the photographer/-camera was not placed on a particularly high artificial construction or was not aboard of an aircraft - generally safe assumption for non-urban outdoor photography.

The size of the image can be described through its Field Of View (FOV), i.e. the extent of the observable world captured by the photograph, expressed in degrees. The FOV can be calculated given some of the properties of the camera that are stored in the EXIF container of the image.

The direction of the photograph, instead, is less trivial to estimate: although the EXIF specifications contain a value representing the camera azimuth angle during the shot (*GPSImgDirection*), it is usually not populated even by devices that have the capacity to do so (i.e. cameras and mobile devices with magnetometer sensor). The direction of the photograph must be inferred based on the image content. Such problems are usually tackled thorough the identification of some objects that act as points of reference. We propose a method for the direction estimation that identifies the only objects we can be sure to be in the image: the mountains themselves. Luckily, there is no need to deal with object tracking and movement problems, thanks to the fact that mountains are among the most motionless and immutable objects on the planet.

Not all mountain features are well suited for the identification: the color features can drastically change withing few hours due to a different illumination and the pattern features can easily mutate as the vegetation is growing or snow is falling/melting. Hence, we identify mountains and mountain slopes using only the *edge features*, as the mountain profile is the only visual property that does not change in time. The idea is, so, to discover the photograph direction by finding an orientation that matches the mountain profile seen in the photograph with the mountain profile that should be seen in that direction.

A Digital Elevation Model (DEM) is a regular grid of the terrain surface that specifies the terrain elevation for every point of the grid. It can also be seen as a 3D model of

the Earth (or its portion). Given an observation point, a 360° panoramic view of the synthesized terrain (and the mountains) can be generated[1].
Our algorithm searches for the best overlap between photograph mountain profiles and the mountains on the rendered panorama. The algorithm can be used both to identify geographical properties of the photograph pixels (e.g. generate a photograph depth map that captures the distance from the observer for every pixel) or to identify the position on the photograph of some geographical objects. We evaluate the algorithm in the latter use case: given a geotagged photograph and the geographical position of the mountain peaks, we estimate the position of the mountain peaks on the photograph.
The algorithm proceeds in four steps, which are illustrated in Figure 4.2:

- **Preprocessing** (Section 4.1.1) ensures that the photograph and the virtual panorama are scaled correctly and can be successfully overlapped.
- **Edge Extraction and Filtering** (Section 4.1.2) extracts the relevant edges from the photograph and the panorama.
- **Global Matching** (Section 4.1.3) finds the best matching between the two edge maps.
- **Local Matching** (Section 4.1.4) locally adjusts the global matching through a non-rigid warping.

4.1.1 Preprocessing and Scaling

In order to be able to find the correct overlap between the photograph and the panorama the two should have the same scale, i.e. the same mountains should have the same pixel size. Since the photograph and the panorama are taken/generated from the same location, the angular size of the mountains on the photograph and the one of the mountains on the panorama are equal by definition. Thus, the scaling problem consists in ensuring that both the photograph and the panorama has the same ratio of angular to pixel size.
Let w_p/w_r be the photograph/panorama pixel width respectively, f be the photograph horizontal Field Of View (FOV). Considering that the panorama horizontal FOV is 2π by construction, we define k the factor by which the photograph should be scaled:

$$k = f \frac{w_r}{2\pi w_p} \tag{4.1}$$

Figure 4.3 shows a simplified schema of a digital camera. The observable world is projected though the lens to the sensor that has a certain size (*sensor width* and *height*) and is placed at a certain distance from the lens (*focal length*). The angle formed between the sensor edges and the center of the lens is, indeed, the FOV.
The focal length can be fixed or variable (on cameras mounting the zoom lens), either way it is commonly stored in the EXIF container of the photograph in the corresponding field (*FocalLength*). The sensor dimension, instead, is a physical property of the sensor and the camera, but it is not stored in the EXIF. In our implementation we consult a

[1] During the early stages of this work we used an external web API for the panorama generation, kindly provided by `www.udeuschle.de`. Later, we developed a proprietary efficient panorama generator that works both on desktop and mobile platforms, using SRTM DEM (`www2.jpl.nasa.gov/srtm`) and OpenStreetMap (`www.openstreetmap.org`) data.

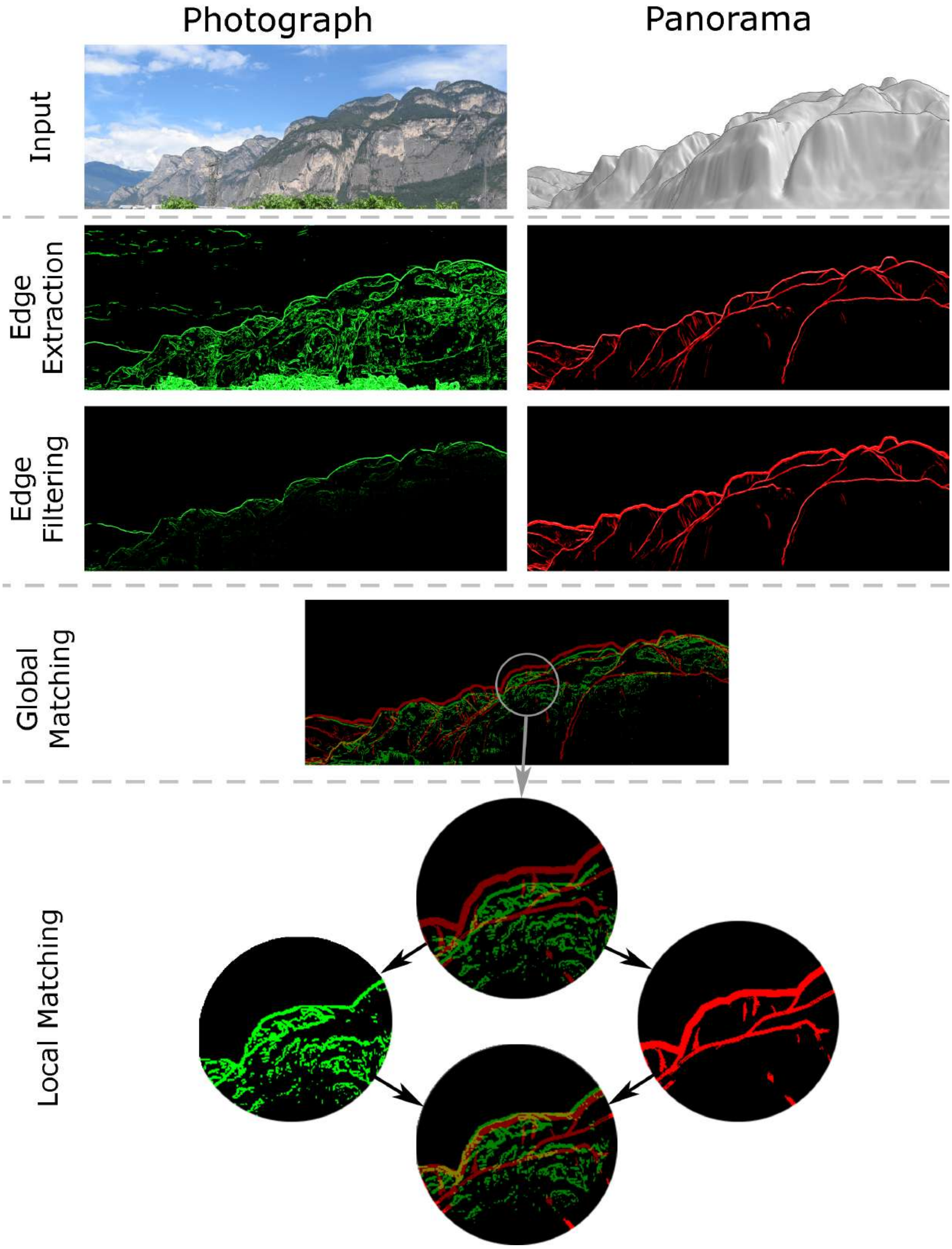


Figure 4.2: *Schematic example of the photograph direction estimation, including the input photograph and the corresponding panorama, edge extraction and filtering, global and local matching. The figure depicts only the relevant portion of the panorama due to the space constraints, however, the panorama must be intended as a full 360° cylindrical image.*

database of digital cameras[2] and retrieve the sensor size by the camera model (both

[2] Kindly provided by `www.digicamdb.com`.

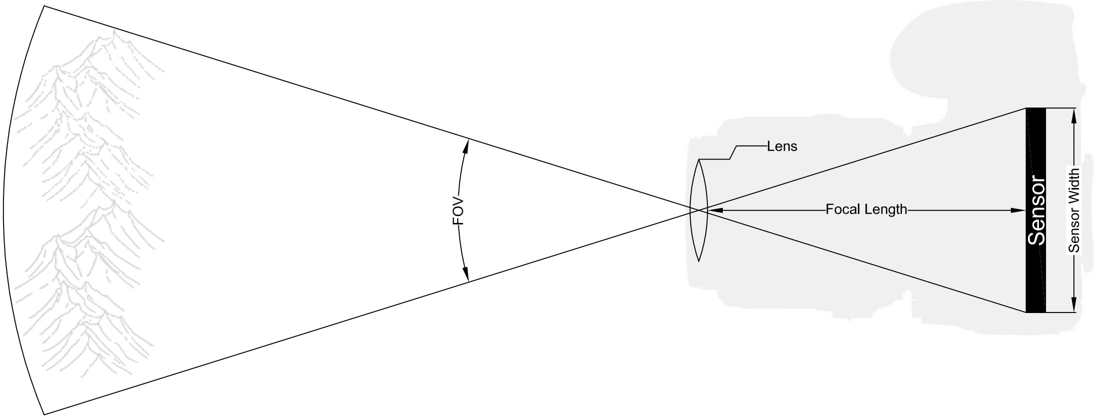


Figure 4.3: *Simplified schema of a digital camera (without considering camera lens distortion).*

stored in the EXIF: *Make* and *Model*). Let l be the focal length and s be the sensor width expressed in the same units as l, the computation of the scaling factor becomes:

$$k = arctan(\frac{s}{2l})\frac{w_r}{\pi w_p}$$

The scaling, however, does not provide a perfect consistency of the photograph w.r.t. the panorama, since the photograph is affected by the camera lens distortion. Furthermore, the same distortion can not be artificially applied to the panorama because it is dependent on the camera orientation, which is unknown. In order to mitigate this issue we apply a reverse camera lens distortion to the photograph and only then proceed to align it w.r.t. the panorama.

4.1.2 Edge Extraction and Filtering

The matching between the photograph and the panorama relies on the terrain profiles, hence, we apply an edge extraction algorithm to both the photograph and the panorama. Furthermore, we use edge extraction algorithm that assigns both edge intensity and orientation to every pixel: this allows the matching algorithm to distinguish between similarly- and differently-oriented edges.

In our implementation we use Compass [150] edge detector, that uses the distributions of pixel colors rather than the mean. It searches for the orientation of a diameter that maximizes the difference between two halves of a circular window. Compass edge detector is superior to a multidimensional gradient method in situations that often result in false negatives and it localizes edges better as scale increases.

However, the resulting photograph edge map contains edges that represent non-terrain boundaries. We call them *noise edges*, since they do not correspond to any edge of the panorama and thus contribute negatively to the correct match. The noise edges can be found either below the skyline (foreground objects such as trees, houses, persons) or above the skyline (e.g. clouds). Furthermore, only boundaries that represent mountain-on-sky and mountain-on-mountain contours are relevant for the matching, while all other contours from non-terrain objects create noise edges. In fact, we observed that the amount of the noise edges is, usually, at least an order of magnitude higher then the relevant edges. Such amount of noise edges jeopardizes the whole matching and can lead to a random direction estimation.

To mitigate this problem, first, a skyline detection algorithms is applied, and all edges contained above the skyline are removed (considered to be clouds and other obstacles). The skyline detection algorithm is an adaptation of [114]: using the already existing edge map, we apply a dynamic programming algorithm that finds the path through the image with the lowest cost. Second, a filtering procedure is applied to the edges of the photograph, by decreasing the strength of the edge points as the vertical position decreases.

The panorama does not contain noise edges by definition, so its edge map is kept intact, furthermore, a morphological dilation is applied to emphasize the edges corresponding to the skyline. The skyline of the panorama can be easily identified as the upper envelope of the edge map, by keeping, for each column of pixels, the topmost edge point. Figure 4.2 shows an example of the edge extraction from both the photograph and the panorama together with the photograph edge filtering and the panorama edge dilation.

4.1.3 Global Photograph-to-Panorama Matching

The matching between the photograph and the corresponding panorama is performed using a Vector Cross-Correlation (VCC) technique proposed in [4], which takes into account both the strength and the direction of the edges. We observed that all the photographs in our dataset has a close-to-zero tilt angle, so we assume the tilt to be $0°$. This allows us to approach the matching problem as a 2D cylindrical alignment, instead of a 3D spherical one. The cylindrical matching, in turn, can be implemented as a 2D image matching in Cartesian space as Figure 4.4 shows.

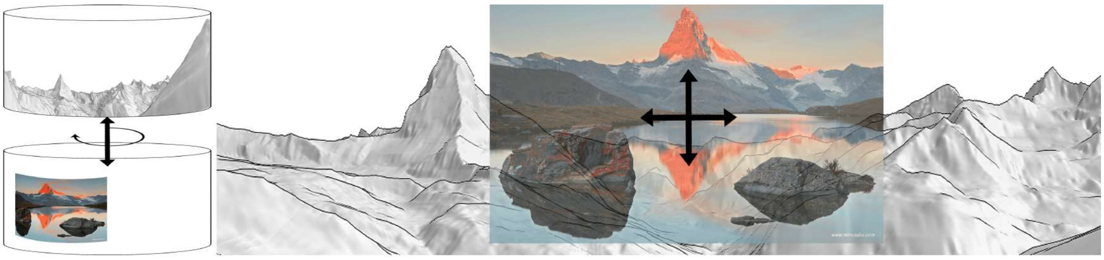


Figure 4.4: *The cylindrical matching (left) can be performed with a 2D cross-correlation in the Cartesian space (right).*

The output of the VCC is a correlation map that, for each possible horizontal and vertical displacement between the photograph and the panorama, indicates the strength of the matching. The strength of a single matching is expressed as the sum of the angular similarity operator between every pair of photograph and panorama overlapping edges. Let (ρ_e, θ_e) be the polar representation of a single edge e, the angular similarity operator between two edges is expressed as:

$$M(e_1, e_2) = {\rho_{e1}}^2 {\rho_{e2}}^2 cos2(\theta_{e1} - \theta_{e2})$$

The cosine factor is introduced in order to handle the noise by penalizing differently oriented edges: the score contribution is maximum when the orientation is equal, null when the edges form a $\frac{\pi}{4}$ angle and reaches the minimum negative when the edges are perpendicular. This penalization avoids that noise edges that randomly overlap with terrain edges contribute in a positive way to a wrong match position.

This correlation matrix can be computed efficiently with the fast Fourier transform. Let f and p be the 2D complex matrices of the photograph and the panorama edge maps respectively, $\hat{f}$ and $\hat{p}$ their respective 2D Fourier transforms, the computation of the VCC matrix becomes:

$$VCC(f,p) = Re\{\hat{f}^2\, \overline{\hat{p}^2}\}$$

Global alignment can match mountain edges also below the skyline and is robust with respect to skyline detection errors. However, the global maximum of the correlation is not necessarily the correct match. This might occur, for example, when some edges of the photograph happen to match the shape of different portions of the panorama. As such, the top-K matches are further analyzed by the refinement step below.

Refining Global Matching

For each of the top-K candidate matches, we measure the Hausdorff distance [2] between the skyline edge points of the photograph and of the panorama, when the two are overlapped at the candidate matching position. A scoring function is computed, which combines the Hausdorff distance and the rank position computed by the initial global alignment. The candidate with the highest score is then chosen as the best match between the photograph and the panorama.

4.1.4 Local Photograph-to-Panorama Matching

Once the correct match between the photograph and the panorama is found, the geographical information of any photograph pixel can be estimated by considering the corresponding pixel on the panorama. Vice versa, the mountain peaks can be projected from the panorama to the photograph.

However, even the best match does not necessarily imply that the panorama and the photograph can be overlapped perfectly. This is shown in Figure 4.5: first, the rightmost part of the photograph does not match the actual skyline, due to the occlusion of a mountain slope close to the virtual observer; second, it is not possible to simultaneously match all the three peaks in the leftmost part of the photograph by means of a simple rigid displacement. The differences between the panorama and the photograph edges arise from several factors: the DEM used to generate the panorama has a finite spatial resolution (namely 30 m in Figure 4.5) and carries some altitude error; the photograph is affected by optical distortion while the panorama is not; the geotag of the photograph contains some errors, thus, the panorama is not generated from the exact same location.

We mitigate this problem by proposing a local matching approach: we choose a set of *local points* in which we want to further improve the matching. Every point is given a certain radius of freedom of movement to find a better matching within its neighborhood. For each local point, a separate VCC procedure is applied, similar to the one used in the global alignment step. Specifically, for each point we consider a local neighborhood centered in the location identified by the global alignment. In this way each

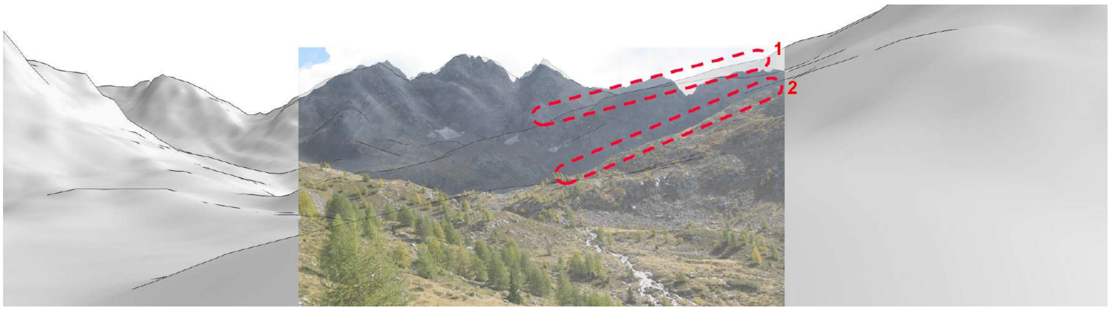


Figure 4.5: *Example of a photograph-to-panorama matching. The three peaks can not be overlapped perfectly by any global matching. Furthermore, the rightmost part of the photograph is different w.r.t. the panorama: red areas 1 and 2 identify the same mountain edge on the panorama and photograph respectively. This error is probably due to a wrong photograph geotag.*

point position is refined by identifying the best match in his local neighborhood. Figure 4.2-bottom shows an example of a local matching of a single local point. Overall, this approach can be seen as an application of a non-rigid warping of the photograph with respect to the panorama for a better edge alignment.

Following the use case of mountain peak identification, we consider all the peaks identified by the global matching step to be local points and apply the local matching.

4.1.5 Mountain Identification Evaluation

We evaluate the performance of the alignment algorithm by estimating mountain peak positions on a set of photographs selected from the ones crawled from Flickr, as described in Section 3.1. We manually inspected a random subset of 200 photographs and the corresponding panoramas generated based on the accompanying EXIF metadata to make sure that a plausible matching existed. Indeed, in some cases, we found that the geotag was available but incorrect, such that the generated panorama could not be matched to the photograph by any means. Finally, we retained 162 photographs in our test set. Then, the ground truth data was generated by an alignment tool developed ad-hoc, which allows the annotator to find the correct position of the photograph w.r.t. the panorama and then to locally warp the image by overlapping each mountain peak present in the photograph to the corresponding one in the panorama.

Measures

For each peak $i = 1, \ldots n$, let (x_i^p, y_i^p) and (x_i^r, y_i^r), denote the pixel coordinates in the coordinate system of the photograph and of the panorama, respectively. When the photograph is aligned with a displacement $(\Delta x, \Delta y)$, we define the angular error in the position of the i-th peak as

$$\epsilon_i(\Delta x, \Delta y) = \sqrt{d_x(x_i^r, \Delta x + x_i^p)^2 + d_y(y_i^r, \Delta y + y_i^p)^2}\,,$$

where

$$d_x(x_1, x_2) = (360/w_r)\min(w_r - |x_1 - x_2|, |x_1 - x_2|)$$

is the angular distance (in degrees) between two points along the azimuth, given the circular symmetry of the panorama, and w_r is the number of pixels corresponding to

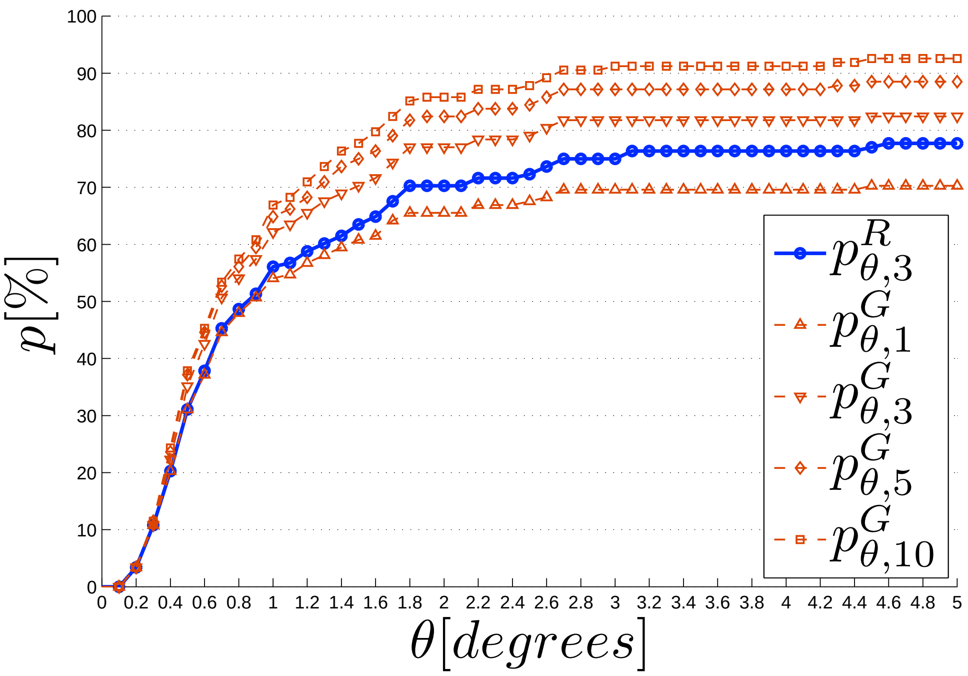


Figure 4.6: *Performance of the photograph-to-terrain global matching and the refinement step.*

360°. Similarly

$$d_y(y_1, y_2) = (360/w_r)|y_1 - y_2|$$

where the same angular resolution in degrees/pixel is assumed due to small elevation angles. When creating the ground truth, the images are warped so as to minimize the average angular error

$$\epsilon(\Delta x, \Delta y) = (1/n)\sum_{i=1}^{n} \epsilon_i(\Delta x, \Delta y)$$

and to find the best displacement

$$(\Delta x^*, \Delta y^*) = \arg\min_{\Delta x, \Delta y} \epsilon(\Delta x, \Delta y)$$

Note that $\epsilon^* = \epsilon(\Delta x^*, \Delta y^*)$ cannot always be reduced to 0, due to the coarse nature of the panorama.

Let $(\Delta x^G_k, \Delta y^G_k)$, $k = 1, \ldots, K$, denote the displacements of the top-K candidate matches of global alignment. We define $p^G_{\theta,K}$ as the fraction of the photos in the test set that have at least one candidate match displacement $(\Delta x^G_k, \Delta y^G_k)$ lying within angular distance θ from the ground truth $(\Delta x^*, \Delta y^*)$. The refinement step selects $(\Delta x^R_K, \Delta y^R_K)$ to be one of the displacements $(\Delta x^G_k, \Delta y^G_k)$ (not necessarily the best). Then, $p^R_{\theta,K}$ is the fraction of photographs for which the difference between $(\Delta x^R_K, \Delta y^R_K)$ and $(\Delta x^*, \Delta y^*)$ is below θ. Note that $p^R_{\theta,K} \leq p^G_{\theta,K}$ by construction, and the equality holds if the refinement step is always able to identify the correct match within the top-K candidates.

The local alignment step computes a different displacement $(\Delta x_i^L, \Delta y_i^L)$ for each of the n peaks. Then, the average error is defined as

$$\epsilon^L = (1/n) \sum_{i=1}^{n} \epsilon_i(\Delta x_i^L, \Delta y_i^L)$$

Results

Figure 4.6 shows the performance of the global matching on the whole dataset. It can be observed that $p_{\theta,K}^G$ saturates when θ exceeds 3°. Specifically, 69.6 % of the photographs are aligned with an average error below 3°, when considering the top-1 match. The fraction of correctly aligned photos grows to 81.8 %, 87.2 % and 91.2 % when K is 3, 5 and 10, respectively. Diminishing returns in the average error are observed when increasing K; thus, we selected $K = 3$ in the refinement step by trial and error method, which results in 78 % of correctly aligned photos. The refinement performance curve lies approximately halfway between the top-1 and top-3 curves of global matching. This shows the benefit of introducing the refinement step and its ability to pick the correct candidate from the top-3 candidates.

Taking a deeper look into the dataset, Table 4.1 describes the performance of the proposed method depending on the different properties of the visual content, manually annotated in two ways; first, we marked whether the photograph contains clouds (80 out of 162); second, we marked the presence of mountains close to the observer that might occlude the skyline in the background (49 out of 162). The presence of clouds is one of the main obstacles to be addressed. This is due to the fact that, when clouds partially occlude the skyline, the outcome of the skyline detection algorithm might fail. In addition, edge points due to clouds above the skyline might compromise the filtering procedure, since the latter is based on the assumption that there are no edges above the skyline. In the case of global matching, the fraction of correctly matched photographs grows to 72.4 % and 82.9 % in the absence of clouds, when considering the top-1 and top-3 candidates, respectively. Conversely, the presence of clouds leads to a reduction of correct matches, which represent, however, at least 66.7 % of the cases. The performance of the refinement step is also affected by the presence of clouds, being equal to 77.6 % / 72.2 % when clouds are absent / present. The impact of clouds is higher in the refinement step than in the top-3 candidates global matching, since the former relies heavily on the correctness of the estimated skyline.

Another issue lies in the presence of mountain slopes nearby the observer. Indeed, in this case small errors in the geotag might lead to a panorama which does not correctly represent the viewpoint of the photograph. This situation is clearly visible in the rightmost part of Figure 4.5. In the case of global matching, the fraction of correctly matched photographs grows to 74.8 % and 89.3 % in the absence of nearby mountains, when considering the top-1 and top-3 candidates, respectively. A similar behaviour is observed for the refinement step (81.6 %).

Local matching further improves the matching between the photograph and the panorama. This is measured by comparing the average angular error between the peak positions after the refinement step, $\epsilon(\Delta x_K^R, \Delta y_K^R)$, with the one obtained after local matching, ϵ_L. In our experiments, we found that the error decreased from $\epsilon(\Delta x_K^R, \Delta y_K^R) = 0.99°$ to $\epsilon_L = 0.78°$, i.e., a 21 % reduction with the radius of the local neighborhood set to 7.5°.

	$p^G_{3,1}$	$p^G_{3,3}$	$p^R_{3,3}$
Whole dataset	69.6 %	81.8 %	75.0 %
Absence of clouds	72.4 %	82.9 %	77.6 %
Presence of clouds	66.7 %	80.6 %	72.2 %
Absence of nearby mountains	74.8 %	89.3 %	81.6 %
Presence of nearby mountains	57.8 %	64.4 %	60.0 %

Table 4.1: *Performance results decomposed by dataset categories and photograph content properties*

Unfortunately, it was not possible to compare our results with those obtained by other algorithms based on similar techniques, due to the lack of a publicly available dataset and unspecified quantitative evaluation metrics [4]. Instead, [3] and [119] address different problems (respectively, geo-tag estimation and relevant image retrieval) and cannot be compared directly with our work.

4.2 Mobile and Real-Time Photo-to-Terrain Alignment

One of the goals of this thesis is to study how the social engagement of the users can benefit automatic web content analysis. To this end, we developed a user-facing dedicated web platform and a public Augmented Reality (AR) mobile application that performs real-time photo-to-terrain alignment. The idea is to entertain and engage the users providing real-time information about the mountains being framed with the mobile device in order to receive more mountain photographs and contributions. Among other features, the AR application identifies the mountain peaks and displays their name on the screen in real-time.

The heuristic/offline approach (described in Section 4.1), however, has several flaws if used in a real-time scenario. While the architecture of the application and the engagement aspects are described in Section 7.3, in this section we discuss the algorithmic part of the mobile image alignment (and specifically mountain peak identification) task, which requires significant adaptations of the heuristic/offline approach.

The main challenges induced by the mobile and real-time AR requirements include:

- Lower computational power w.r.t. offline architectures.
- Higher accuracy required: while it is tolerable for a data mining pipeline to misidentify the photograph direction and the mountain peak positions (the image can be discarded), an erroneous peak identification on a mobile application used live produces a disappointing user experience and the enriched image, once saved, can not be easily fixed on a small screen device.
- Faster response time: peak positions must be overlaid in real-time and no overhead for image processing initialization is acceptable, because mobile users do not tolerate delays in the order of seconds.

On the other hand, a significant advantage of the mobile version is the availability in real-time of the position and orientation sensor values, which, although subject to error, provide an estimate of the panorama in view.

The preprocessing of the image becomes a trivial task on a mobile device: the location is provided by the GPS sensor, while the FOV is provided directly by the camera component, eliminating the need for the FOV computation. The input image captured by the camera is then scaled using Equation 4.1.

4.2.1 Edge Extraction

The former methods work well for offline peak detection, because they are applied to pre-filtered images (fixed webcams have a view that does not change and can be manually checked once and for all for suitability; user-generated photographs go through an offline binary classification step to retain only samples with obstacle-free skyline view). But they are not well suited to a mobile AR scenario, where it is more likely that the camera is used in adverse weather conditions and in presence of transient occlusions of the skyline. In these cases, a cloud, a high voltage cable, or a roof would be recognized as part of the mountain skyline; this would impact the heuristic edge filtering, i.e., a cloud edge would be treated as skyline and the mountain slope below it would be considered as noise. Such erroneous classification would hamper the alignment with the panorama and the positioning of peaks, yielding an unacceptable user's experience.

Furthermore, although the proposed edge detector (Compass) is superior w.r.t. the naive edge detection approaches, it is highly time-consuming (extracting edges from a single mountain photograph takes minutes on top of the line mobile phones). The refining step of the global matching algorithm is also computationally expensive, since it calculates the Hausdorff distance for the top-K matching candidates.

In order to decrease the computational effort and increase the robustness even to small, transient occlusions we devised the following variations:

- The edge map is extracted from the photograph with a simple and fast edge detection algorithm (Canny [20]).
- Instead of dealing with mountain-on-mountain edges, we consider only the skyline edges (i.e. mountain-on-sky) to be relevant. Consequentially, the panorama edge map is defined as the upper edge pixel of each column.
- Every edge of the photograph is fed to a supervised learning classifier that estimates whether the pixel corresponds to a landscape skyline (positive) or not (negative). All negative edges are removed from the edge map. Furthermore, the orientation of the edges is disregarded and the intensity of the remaining edges is assigned proportionally to the probability of the edge to be positive (as estimated by the skyline classifier). The skyline classifier is described in detail in Section 4.2.2.

4.2.2 Skyline Detection

We developed an approach, which finds the *landscape skyline* of a photograph, i.e., the set of all the points that represent the boundary between the terrain slopes and the sky. Every edge pixel of the input image is fed to such binary classifier, and only positive edges are retained. The landscape skyline classification problem can be formulated as follows: given a $N \times M \times 3$ patch of the input image centered in a pixel that corresponds to an edge, estimate whether the central pixel represents a landscape skyline or not.

We developed the classifier based on the application of a Convolutional Neural Network (CNN) supervised learning algorithm. The choice of the CNN over other machine learning algorithms (e.g. Logistic Regression, SVM, Random Forest) was motivated by the ability of the CNN to learn the best features to employ, which avoids their manual, and subjective, definition.

Recently, Convolutional Neural Networks have been demonstrated as an effective class of models for understanding image content, giving state-of-the-art results on image recognition, segmentation, detection and retrieval [97]. CCN architectures make the explicit assumption that the inputs are images, which allows to encode certain properties into the architecture. These then make the forward function more efficient to implement and vastly reduce the amount of parameters in the network [96].

A typical downside of using the CNN is the need of a very large amount of training data. This downside was not an obstacle in our case, because the items to be classified are small patches extracted from the input image edges; in our experiments, an average 640 × 480 pixel outdoor image contained tens of thousands of such edge pixels. Furthermore, as described below, the pixel-wise dataset annotation was not required.

A CNN consists of a number of convolutional and subsampling layers optionally followed by fully connected layers. In our network the input images are scaled to 640×480 pixels and for each candidate edge pixel a $28 \times 28 \times 3$ patch centered in the pixel position is extracted. The adopted network topology is described in Table 4.2.

Layer	Type	Input	Filter	Stride	Output
Layer 1	Conv	$28 \times 28 \times 3$	$5 \times 5 \times 3 \times 20$	1	$24 \times 24 \times 20$
Layer 2	Pool Max	$24 \times 24 \times 20$	2×2	2	$12 \times 12 \times 20$
Layer 3	Conv	$12 \times 12 \times 20$	$5 \times 5 \times 20 \times 50$	1	$8 \times 8 \times 50$
Layer 4	Pool Max	$8 \times 8 \times 50$	2×2	2	$4 \times 4 \times 50$
Layer 5	Conv	$4 \times 4 \times 50$	$4 \times 4 \times 50 \times 500$	1	$1 \times 1 \times 500$
Layer 6	Relu	$1 \times 1 \times 500$	max(0,x)	1	$1 \times 1 \times 500$
Layer 7	Conv	$1 \times 1 \times 500$	$1 \times 1 \times 500 \times 2$	1	$1 \times 1 \times 2$
Layer 8	Softmaxloss	$1 \times 1 \times 2$			1×2

Table 4.2: *Landscape skyline classifier CNN topology.*

CNN Application

Normally, a machine learning algorithm for binary pixel-level classification would require us to extract a patch for every edge and perform a skyline/non-skyline classification on that patch as shown in Figure 4.7.

Given a CNN, instead, we are able to remove the last softmax layer from the network and apply the CNN to the whole image in one global convolution, resulting in an impressive speed-up. In our tests a global CNN convolution took $\sim$ 300 ms on a Nexus 6 phone, while running independently every edge patch through the CNN took 5 s - 10 s, based on the number of edges in the photograph.

The result of the CNN application is a matrix that assigns the skyline classification score to every pixel of the image (except the few pixels on the border). Figure 4.8-bottom-left shows an example of such matrix that may seem to be inaccurate, since the sky pixels tend to have a positive score: however, we must consider that the CNN is

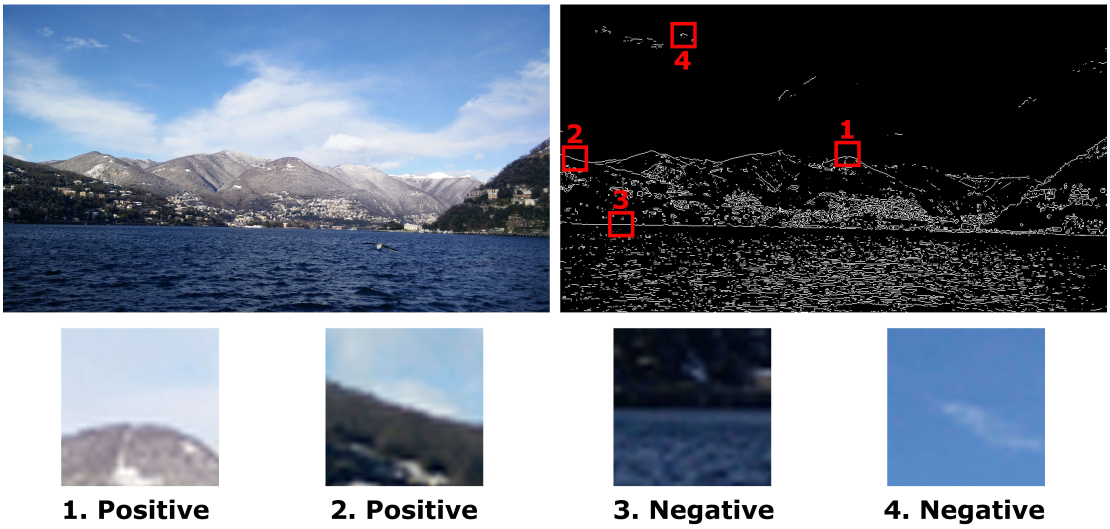


Figure 4.7: *Examples of patches centered in edge pixels extracted from a mountain image.*

trained to classify only the edge pixels. While the skyline likelihood score is provided for every pixel, the CNN is not trained to classify non-edge pixels as negative, thus, the classification of non-edge pixels is random and irrelevant. In fact, this does not create a problem, since we perform a pixel-wise multiplication between the skyline map and the edge map, retaining only edges that are classified as skyline. This is clearly shown in Figure 4.8 (bottom right).

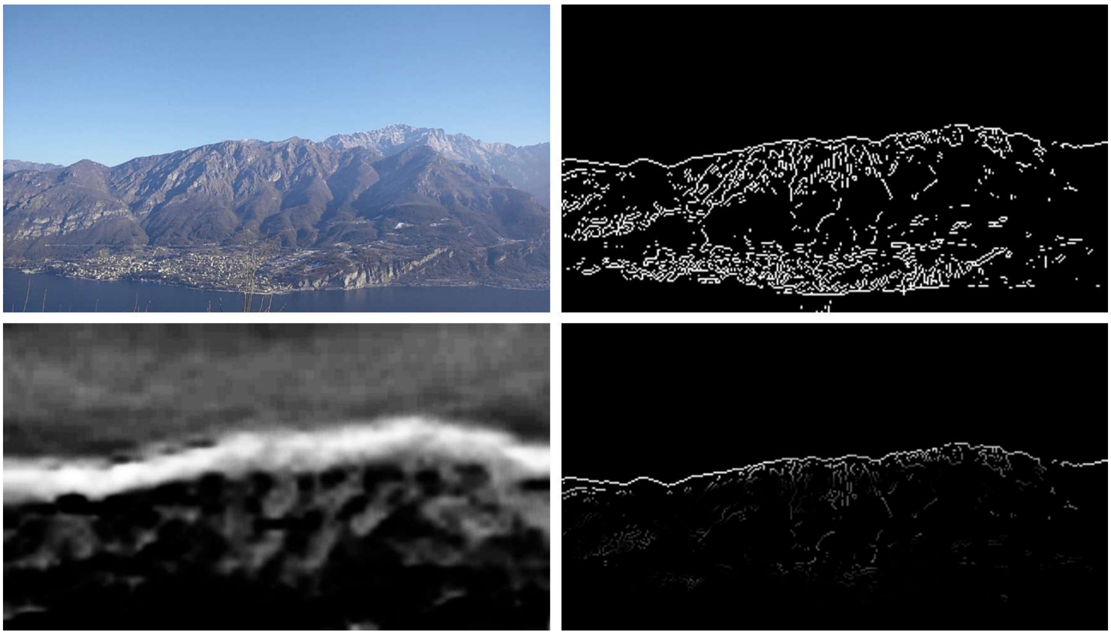


Figure 4.8: *Example of the CNN application including the original image (top left), the corresponding edge map (top right), the CNN skyline score matrix (bottom left, white - positive, black - negative) and the final skyline edges (bottom right, the scalar product of the edge map and the skyline score matrix).*

Dataset Collection and Experiments

The required classification task is narrow and specific: we want to identify only *landscape* skyline pixels, consequentially, all the edges between a non-terrain object (artificial obstacle, living being, foreground vegetation) and the sky should be classified as negative. Furthermore, we want the dataset to be representative of mountain views that are normally seen by users using the application (containing obstacles, taken in adverse weather conditions, etc.). Since we were not able to identify an existing dataset with such characteristics, we created one specifically for this task.

Photographs dataset: we started with selecting $\sim$ 8.2 k random images from our user-generated mountain photograph dataset crawled from Flickr (Section 3.1). In order to filter out photographs that were not relevant examples (a tiny portion of the skyline is visible, extremely bad weather conditions with no mountain visibility, etc.), for every photograph, the annotator was asked if the photograph represents a credible scenario in which the mobile application could be used. If the annotator answered negatively - the photograph was discarded, otherwise the annotator was invited to draw the landscape skyline over the photograph. Out of 8.2 k photographs 1.7 k were discarded (20.6 $\%$) and 6.5 k annotated with the landscape skyline.

Since every skyline was annotated by a single user, we performed a second crowdsourcing experiment to ensure the annotation quality. The original photograph with the annotated skyline in overlay was displayed to another annotator and asked if the skyline seemed correct, discarding the skylines that were marked as incorrect. The filtering phase removed 0.55 $\%$ of the annotated skylines.

For each of the remaining photographs all the patches corresponding to the edge points were extracted and classified automatically (positive if the point is located no more than d pixels from any annotated skyline point, negative otherwise). With this semi-automatic procedure, it was possible to generate the massive amount of pixel-level training data necessary to train the CNN without actually performing a pixel-level annotation.

In order to avoid unbalancing the dataset and prioritizing the edge-rich images, we fixed a threshold N and we kept only a random subset of N positive and N negative patches from each image (or kept all the patches if less than N). We set $d = 5$ and $N = 400$, which resulted in $\sim$ 5.7 M total patches (approximately half positive and half negative).

Selecting a random subset of the patches to form the test set would result in a strong bias in the evaluation: statistically, for every patch in the test set there would be very similar patches (extracted from the same photograph) in the train set. The capacity of the classifier to estimate the skyline in novel illumination and terrain scenarios would be compromised and the performance overestimated. Thus, we split the dataset at *image level*: all the patches of a random subset of 65 $\%$ of the images formed the train set, all the patches from a subset of 25 $\%$ of the images formed the validation set, while all the patches from the remaining 10 $\%$ of the images formed the test set.

The performance of the classifier trained and tested on the aforementioned dataset resulted in 95.25 $\%$ accuracy, and the visual inspection of test set image skylines was satisfactory. However, during the field testing of the application we noticed that the performance were very poor when the weather conditions were not perfect. After the analysis of the dataset we realized that the problem consisted in the fact that Flickr photographs tend to depict very good weather conditions (since users choose their best shots to be uploaded) so the CNN was not trained to deal with cloudy weather, which is a common scenario in daily app usage.

Mixed dataset: in order to mitigate the good weather bias we enriched the dataset with webcam images. For each webcam in our dataset we randomly extracted 2 - 3 images taken during daylight hours no closer than 2 month to each other, ending up with a total of 4.7 k webcam images. Webcams capture images on a regular basis independently from the weather, thus, this dataset extension allowed us to capture the realistic weather scenarios in all the four seasons of the year. All the images were fed to the same annotation pipeline (1.2 k were discarded due to extremely bad weather and visibility) and their skyline extracted, resulting in 2.5 k annotated webcam images.

As a first experiment we added a small portion of the webcam images to the Flickr photographs dataset and replicated the classifier evaluation: the accuracy of the classifier decreased, confirming that there was indeed a bias towards the good weather (experiments on a dataset of 6.5 k photographs plus 0.9 k webcam images resulted in 92.4 % accuracy). Trained on the full dataset (6.5 k + 2.5 k), however, the CNN was able to adapt to the various weather conditions, obtaining a **final accuracy of 94.6 %**. Figure 4.9 depicts the ROC curve obtained by the final CNN configuration.

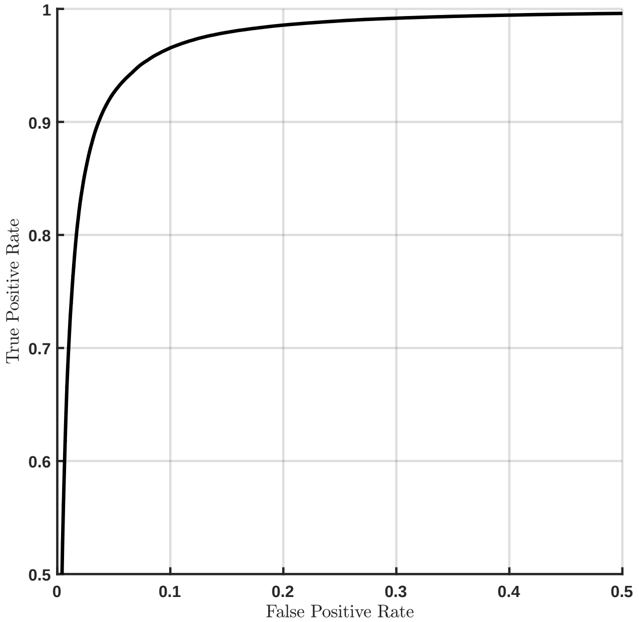


Figure 4.9: *The ROC curve obtained by the pixel-level binary CNN classifier.*

4.2.3 Global Matching

Since the orientation of the edges is discarded, a normal cross-correlation is used during the global matching. The refinement of the global matching using the Hausdorff distance is also removed.

Instead, we use the sensed orientation of the device to improve the performance of the matching step. Since the match between the virtual panorama and image skyline is approximate, each candidate peak position receives a score, which is an estimate of the confidence of the matching algorithm (the cross-correlation score). Such score can be manipulated to take into account the agreement between orientation as sensed from the compass and estimated by the orientation sensor of the device. For example, a kernel function based on the difference between the sensed and estimated orientation can be used as a scale factor.

Furthermore, the computation of the matching can be avoided in the areas of the image in which the kernel factor is equal to zero, because those regions would provide an unreliable peak position estimation. Such optimization decreases the computation time: we assume a maximum 15° orientation sensor error and perform the photo-to-panorama alignment not in the whole 360° panorama, but only in a $30^\circ + FOV$ portion of it.

Thanks to the supervised learning skyline detection, the algorithm is able to deal with very adverse conditions: Figure 4.10 shows an example of such matching. The input image (top left) is taken from behind a window, the corresponding fragment of the panorama (middle left) contains two mountain peaks (red arrows). The edges extracted from the input image (top center) contain an enormous amount of noisy edges (mountain vegetation, houses, window frame) that would make the alignment with the panorama impossible; the CNN filtering procedure (top right, green points) successfully retains only skyline edge pixels. The panorama skyline to match is extracted simply by picking top points (middle center, red points); the alignment between the two skylines (middle right) allows us to project the two peak positions on the input image with high precision (bottom, augmented image).

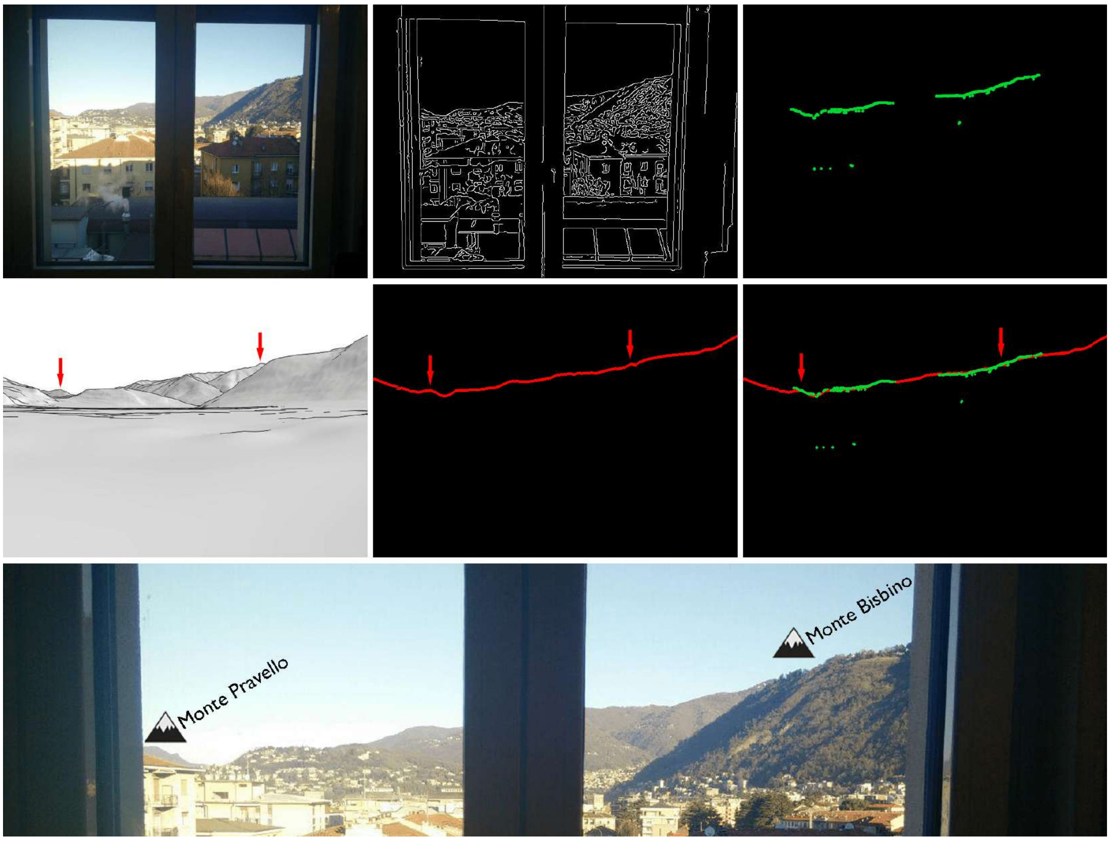


Figure 4.10: *Example of the peak identification in presence of many noisy edges.*

Non-Zero Tilt

The offline version of the matching algorithm assumes the camera tilt as negligible (equal to $0°$) and reduces the problem to the matching between two cylinders, avoiding the (much more costly) spherical match. This assumption proved viable experimentally; mountain ranges are far from the position of the user and the error induced by a moderate tilt is compensated by the skyline matching algorithm. On a mobile device the assumption of zero or constant tilt must be relaxed, to cope with the movements of the mobile device made by the user during a viewing or shooting session. To avoid switching from 2D cylindrical to 3D spherical alignment, which would jeopardize the response time, we designed an approximate approach: the input image is rotated by the tilt provided by the orientation sensor, standard 2D alignment is performed, and the final peak coordinates are rotated in the inverse direction at the end. This method deals with tilting effectively and preserves the fast response time of the 2D alignment, to obtain corrections to the 3D object positions.

4.2.4 Occlusion Management

The virtual panorama view contains only the peaks that could be visible by an observer based on the elevation model; in the real image, virtually visible peaks can be occluded by irrelevant objects, such as houses, people or even clouds or fog. The CNN network used for edge filtering in the mobile AR scenario helps dealing with occlusions: the network is trained to recognize the *landscape skyline*, i.e., the portion of the topmost edges that actually represent the boundary of a mountain slope. This capability supports effective occlusion detection. Given a correct alignment between the landscape skyline of the image and the virtual skyline of the panorama, the peaks that are actually visible in the image will have fragments of the landscape skyline in their vicinity, while occluded peaks will not. Thus, once the alignment is found, for each peak a visibility score v is defined as the number of landscape skyline points located no farther than d pixels from the peak position. A peak is considered visible if $v \geq \bar{v}$ (where $\bar{v}$ is a fixed threshold). Figure 4.11 shows an example of peak identification with 3 virtually visible peaks. In this case, peak n.2 is occluded by the bell tower; indeed, besides a few false positive pixels, the bell tower contour is absent in the overall identified landscape skyline (top right). After the alignment, the neighborhood of each peak is analyzed (bottom right): peaks n.1 and n.3 present a large number of landscape skyline points (green dots) in their vicinity, while peak n.2 does not, so it is marked as non-visible and not included in the augmented image (bottom left).

4.2.5 Experiments

The evaluation of the matching accuracy was performed on the *VENTURI Mountain Dataset* [144]. The data set is a collection of 12 outdoor sequences accompanied with GPS positions and orientation sensor logs, resulting in 3 117 frames. For each frame the position of the mountain peaks is manually annotated. We measured the performance in terms of average peak position angular error. The observed average peak position error was $1.32°$, which is lower than the minimum error obtained by the authors of [144] and defined suitable for mobile computation, namely $1.87°$. The average time currently required by peak identification algorithm to process a frame is less than 400 ms. Such

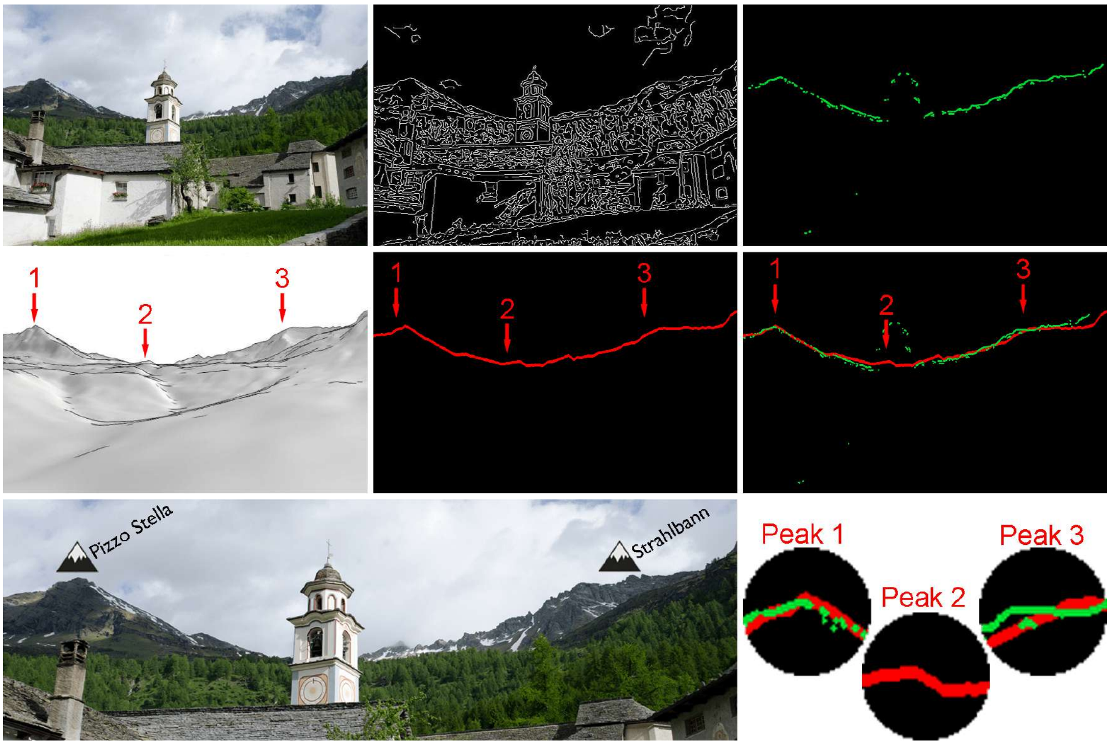


Figure 4.11: *Example of the peak identification in presence of occlusions.*

timing is totally dependent on the architecture and characteristics of the device being used, which in this case correspond to a Motorola Nexus 6 with Chipset Qualcomm Snapdragon 805, CPU Quad-core 2.7 GHz Krait 450, GPU Adreno 420, RAM 3GB, OS Android 5.1.1.

The average frame process time can be higher on lower-end mobile devices, however, due to the architecture of the system that is described in Section 7.3, the *on-screen peak positioning is always real-time* thanks to the sensor data, while the time complexity of peak identifiers influences only the update frequency of corrected peak positions.

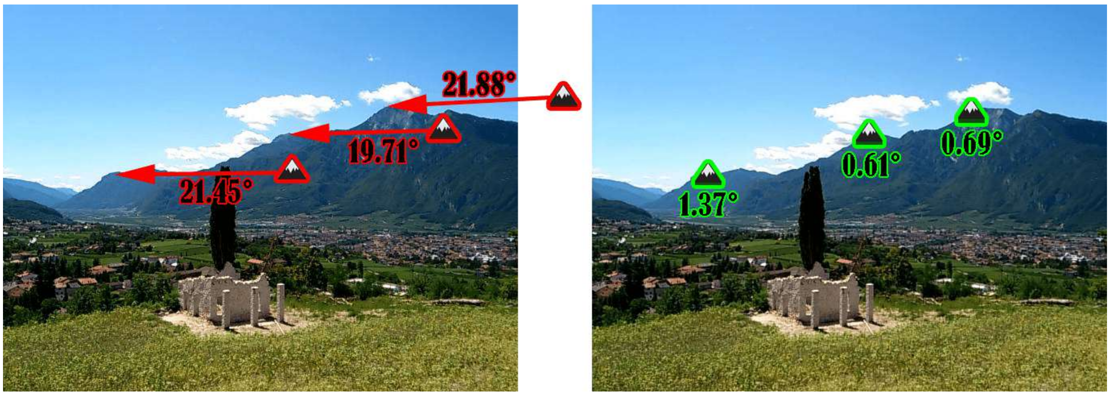


Figure 4.12: *Example of the peak identifications based on the device sensor only (left) and alignment approach (right). Images from the Venturi dataset [144].*

Figure 4.12 shows an example of the mountain identification process in the mobile

AR scenario. Initially, the on-screen peak positions are determined only through the orientation sensor data (left, red icons represent the predicted positions, arrows the real positions and the angular error is reported). After the photo-to-panorama alignment is performed, the peak positions are estimated more precisely (right).

CHAPTER 5

Estimating Snow Cover

Once a valid image that contains a mountain slope is retrieved and geo-registered, the portion of the image that represents the mountain area can be analyzed w.r.t. the concerned phenomena. In this chapter we describe a specific use case, which is snow cover monitoring. However, similar techniques can be applied to different problems, such as vegetation cover monitoring [186], water bodies [168], etc.

Given an image, our goal is to compute a *Virtual Snow Index (VSI)*, which, thanks to the orthorectification [120] of the photograph, can be correlated with the fraction of the visible area covered by snow. The area of the image that corresponds to the mountain surface must be divided into snow and non-snow areas. Although such segmentation of a mountain picture is often a trivial task for a human eye, it represents a challenging computer vision problem. As an example, Figure 5.1 shows several 7×7 pixel patches extracted from a webcam image. If someone was asked to classify these patches as snow or terrain areas without looking at the original image in the lower part of the figure, one would probably state that the first 3 patterns represent terrain, and the last 4 correspond to snow areas. Looking at the whole image, though, it would be possible to notice that, counter-intuitively, the patches n.2 and n.3 correspond to snow covered areas, whereas patches n.5 and n.6 are extracted from terrain areas. This example shows how the pixel-level snow classification heavily depends on the context of the image, and not only on single pixel intensities.

5.1 Snow Mask Identification

In this section we illustrate and evaluate approaches for pixel-level snow classification described in the literature. All listed methods adapt in an implicit or explicit way to different illumination conditions: threshold derived from statistical analysis of the

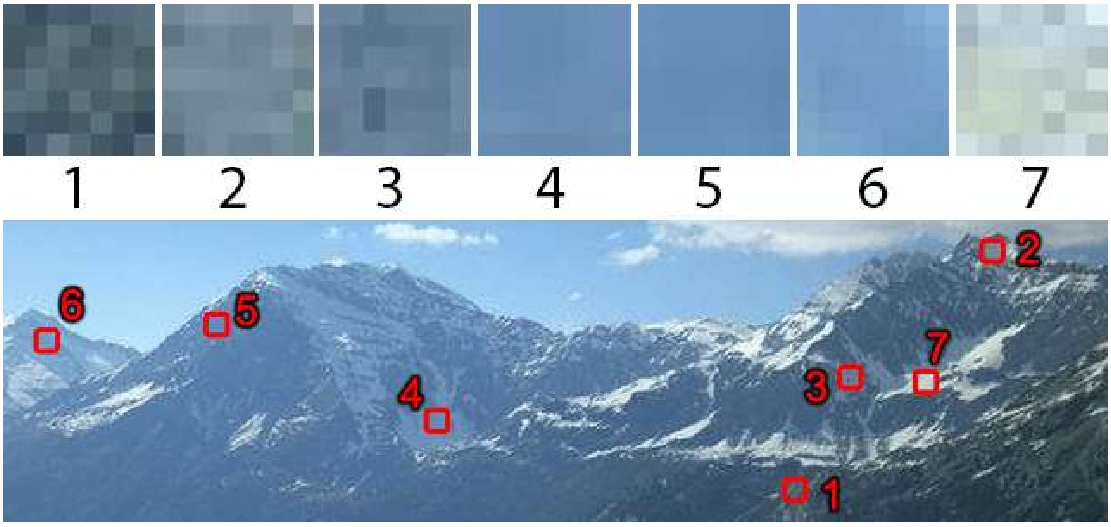


Figure 5.1: *Several patterns of a webcam image. Patterns 1,5,6 represents terrain/vegetation area, while patterns 2,3,4,7 belong to the snow covered area.*

pixel intensities [153], empirically defined color bands [84], and probabilistic model fitting [149]. Even so, all of them classify pixels as "snow" or "non snow" considering exclusively their intensity values. Conversely, given the challenging nature of the task, **we propose** and study the benefits obtained by a **novel supervised learning algorithm** that considers also the context of each pixel. State-of-the-art review specific to the image snow identification problem is proposed in Section 2.3.4.
Let I denote the input image and M the binary mask having the same size as I, where $M(x, y) = 1$ indicates that the corresponding pixel of the image belongs to the mountain area, and $M(x, y) = 0$ otherwise. The snow cover estimation is performed by a pixel-level binary classifier that, given I and M as input, produces a snow mask S that assigns to each pixel a binary label denoting the presence of snow.

$$S(i,j) = \begin{cases} 1 & if\ I(i,j)\ \textit{corresponds to snow covered area} \\ 0 & otherwise \end{cases} \quad \forall\{(i,j)|M(i,j) = 1\}$$

As a **baseline method**, we consider a naïve method, henceforth called **<u>Fixed Threshold</u>**, which applies a simple threshold to a grayscale image, assuming that snow pixels are brighter. Given an input grayscale image G and a threshold value $\bar{t}$, the resulting snow mask is defined as:

$$S(x,y) = \begin{cases} 1 & if\ G(x,y) \geq \bar{t} \\ 0 & otherwise \end{cases}.$$

The methods for pixel-grained snow classification evaluated in this work include:

<u>Snow-noSnow</u>: Salvatori et al. [153] propose a pixel level snow classifier called *Snow-noSnow*. It is based on the analysis of the blue component of an RGB image, because the snow surface presents higher reflectance values in the blue wavelength range. The authors claim that in 90% of the cases the histogram of any RGB component of a mountain image is shaped as a bimodal distribution. Let B denote the blue component of the image normalized in the range $[0, 255]$ and BH the histogram of the intensity values of $\{B(x,y)|M(x,y) = 1\}$ where M denotes the mountain area mask. The classifier of [153] applies a threshold to each pixel of the blue component:

$$S(x,y) = \begin{cases} 1 & if\ B(x,y) \geq t \\ 0 & otherwise \end{cases},$$

where t is equal to the first local minimum of BH greater then $\bar{t}$, or $t = \bar{t}$ if such local minimum does not exist. The parameter $\bar{t}$ represents the lowest empiric intensity value of a snow pixel.

RGB Normalized Difference Snow Index (RGBNDSI): Hinkler et al. [84] describe a classifier that applies a threshold not on a single color band, but on an empirically derived band, called RGBNDSI. The idea is to find a fictitious band, which is related to the Mid-Infrared (MIR) band used for Normalized Difference Snow Index calculation. Such index is used for the snow cover analysis of satellite imagery [152]. Let R, G and B denote, respectively, the three components of a true color image and let:

$$RGB = \frac{(R+G+B)}{3},$$

$$RGB_{high} = \frac{B^3}{R^3}G^3,$$

$$\tau = 200(a(avg(RGB_{high})) + b),$$

$$MIR_{replacement} = \frac{\tau^4 max(RGB(x,y))}{RGB^4},$$

where $MIR_{replacement}$ is an empirical approximation of the MIR band, τ is an index of the brightness of the image and RGB_{high} is an empirically derived matrix. The authors state that τ can be expressed as the mean of RGB_{high}, but a further linear transformation is applied to improve the performance in case of a large fraction of dark pixels. The values of a and b are derived by the authors for the specific camera used in the experiments, thus can not be applied to our context. For this reason, as suggested in [84], we set $\tau = avg(RGB_{high})$. Finally, the derived color band to be thresholded is defined as

$$RGBNDSI = \frac{RGB - MIR_{replacement}}{RGB + MIR_{replacement}},$$

and the estimated snow mask is:

$$S(x,y) = \begin{cases} 1 & if\ RGBNDSI(x,y) \geq \bar{t} \\ 0 & otherwise \end{cases}.$$

Once again, the threshold value $\bar{t}$ is derived empirically. To this end, a statistical threshold selection method proposed by Salvatori et al. [153] can be applied. The RGBNDSI method is an extension of the Snow-noSnow method, which replaces the blue band with an empirically derived one.

Gaussian Mixture Model (GMM): Rüfenacht et al. [149] propose a snow classifier based on a GMM, where all the pixels to be classified are considered points in a 3 dimensional color space. A Gaussian mixture distribution with $k \geq 2$ components is fitted to the set of points $\{I(i,j)|M(x,y) = 1\}$. The Gaussian component with the highest mean intensity value is considered as the snow component, whereas all the others are deemed non-snow components. Each pixel is then labeled as snow, if its probability to belong to the snow component $p(x, y)$ is higher than a given threshold $\bar{t}$:

$$S(x,y) = \begin{cases} 1 & if\ p(x,y) \geq \bar{t} \\ 0 & otherwise \end{cases}.$$

Being a parametric model, the GMM has several parameters that can be optimized. Following the tests of combinations of number of mixture components, as well as type of covariance matrix (spherical and full) performed in [149], we adopt the best reported configuration (namely, XYZ color space, 3 components, spherical covariance matrix).

Supervised Learning Snow Classifiers: in addition to the methods previously proposed in the literature, we propose supervised learning methods that, differently from the traditional approaches, consider also the context of every pixel.
For each pixel, a feature vector of 33 elements is built and fed as input to a binary classifier. Given an image I, represented with a 3 dimensional color space, the feature vector of each pixel $\{(x, y)|M(x, y) = 1\}$ is obtained as the concatenation of 3 feature vectors, one for each color band I^k, $k = 1, 2, 3$. The 11 elements feature vector of each color band includes: 9 values for the pixel intensities contained in the 3×3 neighborhood of the analyzed pixel, 1 value representing the global intensity GI, and 1 for the local intensity $LI(x, y)$. The global intensity is defined as the average intensity of all the pixels representing the mountain area:

$$GI = avg(\{I^k(x_i, y_i)|M(x_i, y_i) = 1\})$$

The local intensity is the average intensity of the pixels within the mountain area, defined as:

$$LI(x, y) = avg(\{I^k(x_i, y_i)|M(x_i, y_i) = 1 \wedge \|(x, y) - (x_i, y_i)\| \leq \bar{d}\})$$

The extent of the neighborhood is conveyed by the radius $\bar{d}$, which was set to 15 pixels. We evaluated this approach feeding the feature vectors to three supervised learning classifiers: **Support Vector Machine (SVM)** [187], **Random Forest (RF)** [85] and **Logistic Regression (LR)** [111].

5.2 Snow Mask Post-Processing

As mentioned in Chapter 3, it is common for a webcam to face bad weather conditions. If all the daily images are affected by low visibility it is not possible to produce the Daily Median Image (DMI) and to estimate the snow cover. Also, if the DMI is generated with few images, it can still suffer from solar glares and occlusions. In order to robustly estimate snow cover, it is possible to exploit the fact that such phenomenon varies slowly in time and that the neighborhood pixels are likely to belong to the same class ("snow" or "non-snow"). To this end, a post-processing method is proposed, which allows us to estimate the snow cover also for the days where no input data is available (due to missing data from the webcams or when all images are taken in bad weather conditions). Let $S_i, i = 1, \ldots, D$, denote the snow mask of the i-th day, given a number N of estimated daily snow masks observed in an interval of $D \geq N$ days.

We obtain all missing snow masks by linear interpolation. For each day i, such that S_i is missing, we consider the closest available masks, i.e. S_{i-k_1} and S_{i+k_2}:

$$S_i(x, y) = \frac{k_1}{k_1 + k_2} S_{i-k_1}(x, y) + \frac{k_2}{k_1 + k_2} S_{i+k_2}(x, y).$$

Once snow masks are computed for each day, we apply a **median filter in the spatio-temporal domain** to each pixel of each mask, defined as

$$S_i^{s,t}(x, y) = med\{S_{i-t}(x - s, y - s), \ldots, S_{i+t}(x + s, y + s)\},$$

where $med\{\cdot\}$ denotes the median operator, s and t are respectively the extent of the spatial and temporal window.

5.3 Snow Classification Performance Evaluation

In this section we describe the experiments performed in order to evaluate the accuracy of the proposed snow classifiers. We explore different datasets with different characteristics and describe which methods perform well/poorly in which conditions.

5.3.1 Datasets

To evaluate the performance of the snow cover estimation methods we considered 3 different datasets. The *Webcams* dataset comprises the images collected from two publicly available webcams placed in proximity of ground meteorological stations. This allowed us to have a reliable source of data for the study of the consistency of the snow estimations with respect to other measurements, such as air temperature. The *PermaSense* dataset was collected by the PermaSense project[1] at the Matterhorn field site and used in [149]. The *Photos* dataset is a subset of randomly extracted user-generated mountain photographs crawled from Flickr, as described in Section 3.1. Figure 5.2 shows a sample image from each dataset, highlighting with the opacity the region of interest (i.e. the binary mask M), while Table 5.1 reports the detailed information about the datasets. For each dataset, a subset of the images uniformly distributed over time was selected, and for each image the groundtruth snow mask was created by manually tagging all the mountain area pixels as “snow” or “non-snow”. Each image of the *Photos* dataset was included in the labeled image set. A total of 7 M pixels contained in 59 images were manually labeled. Each dataset has its own specific characteristics and is studied separately.

In order to normalize the testing conditions, all the input images were downsampled so that at least one of the dimensions reached a fixed maximum value ($\bar{w}$ and $\bar{h}$ respectively). A scale factor $k = min(1, max(\frac{\bar{w}}{w}, \frac{\bar{h}}{h}))$ was applied to each image, where w and h are respectively the width and the height of the image. In our implementation we set $\bar{w} = 640$ and $\bar{h} = 480$.

For each dataset, we defined P_i as the collection of all the pixels that were assigned a label snow/non-snow. P_i was split in a subset of 80 k samples forming the test set, and the remaining samples were assigned to the training set. In order to evaluate the ability of the supervised learning classifiers to adapt to different imaging conditions, they were trained using the data equally distributed from all the datasets.

[1] http://www.permasense.ch

Table 5.1: *Description of the datasets used for snow cover estimation experimental study.*

Dataset	Description	Location	# images	# labeled images
Webcams	Webcam #1: Single mountain, well defined snow line	Bormio, Italy	343	10
	Webcam #2: Plural mountain peaks, snow at different altitudes	Valmalenco, Italy	338	10
PermaSense	Webcam framing a small portion of Matterhorn mountain	Switzerland	2491	19
Photos	Random sample of crawled mountain photographs	Italy-Switzerland border	20	20

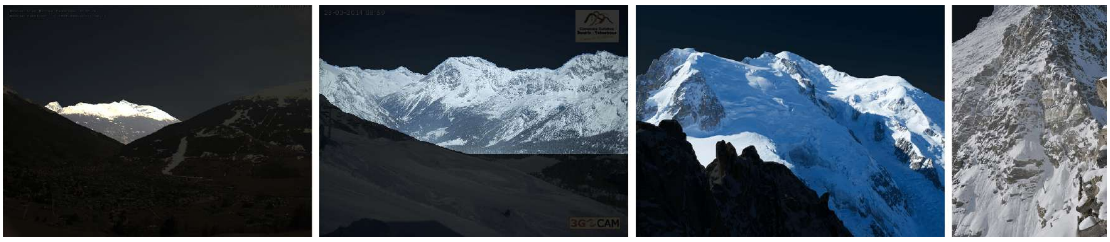


Figure 5.2: *Sample images of the four different datasets (from left to right:* Webcam 1, Webcam 2, Photos, PermaSense*).*

Results

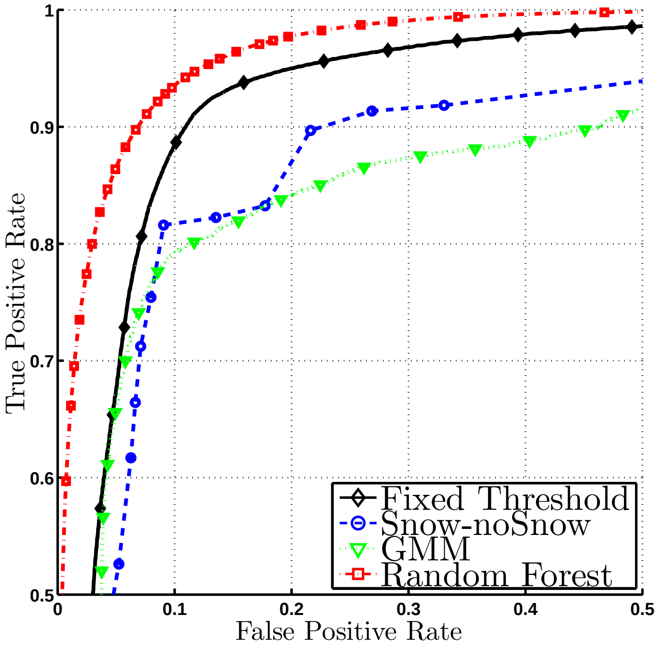


Figure 5.3: *ROC curves obtained in the Webcams dataset by the different snow classifiers.*

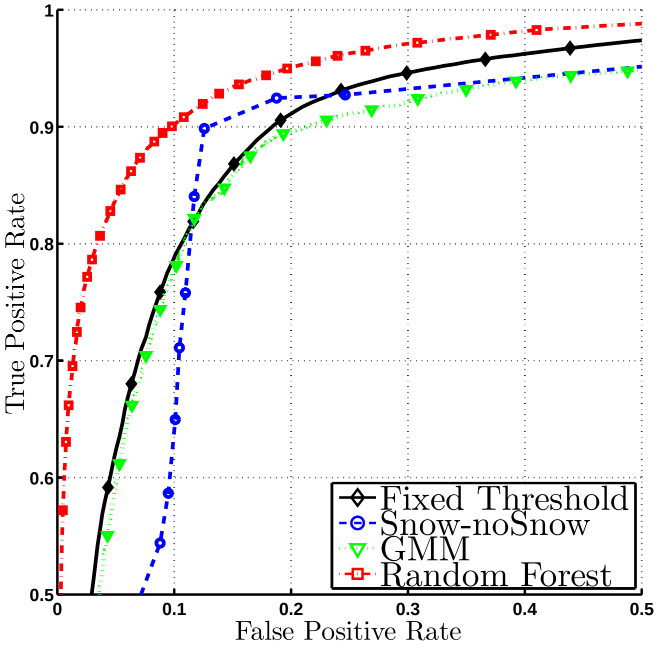


Figure 5.4: *ROC curves obtained in the PermaSense dataset by the different snow classifiers.*

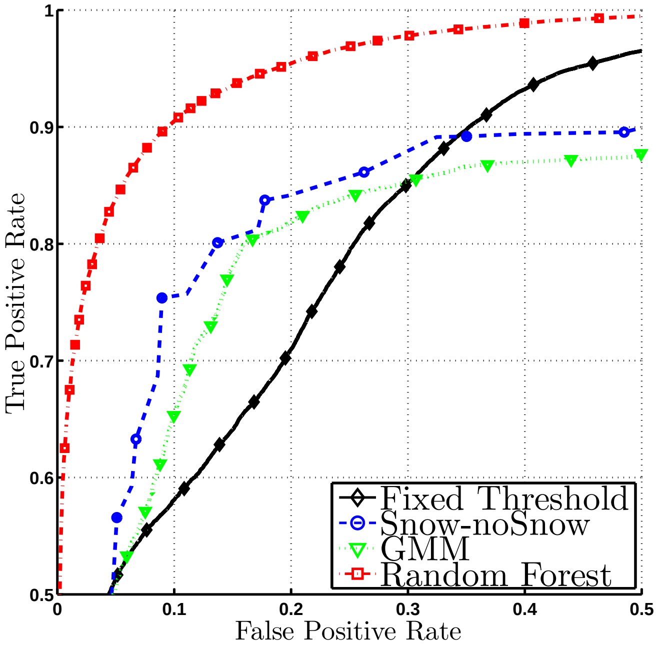


Figure 5.5: *ROC curves obtained in the Photos dataset by the different snow classifiers.*

Figure 5.3 shows the ROC curve obtained on the *Webcams* dataset. We report results for the *Fixed Threshold* baseline, the *Snow-noSnow* classifier, and the *GMM* classifier with 3 Gaussian components. We also include the results obtained by the *Random Forest (RF)* classifier - as the best performing supervised learning method - which was trained on equally balanced data from all datasets. The number of trees of the RF is 50, all the variables are selected for each decision split. Dealing with images taken from the same webcam, the *Fixed Threshold* method adapts efficiently to the common illumination factors and performs better than the other non-supervised methods (including *RGBNDSI*, not shown to avoid cluttering the figure). The *RF* method dominates the others, showing the improvement obtained by exploiting the pixel context.

Table 5.2 shows the True Positive Rate (TPR) obtained by all classifiers when keeping the False Positive Rate (FPR) fixed at 0.1. The GMM method was evaluated with 2 and 3 Gaussian components. The first column contains the TPR obtained without using Daily Median Images (DMIs), as described in Section 3.2.3, and without the spatio-temporal median filter. In this case, the image with the highest skyline visibility score (described in Section 3.2) is used as the representative of each day. The second column shows the TPR with the DMI. The third column specifies the TPR obtained with both the spatio-temporal median filter and the DMI. The use of the DMI improves the performance of all methods, while the spatio-temporal median filtering has a positive impact only on those methods obtaining low TPR (*GMM* methods) and a negative impact on more accurate methods (*RF*, *SVM*, *LR*, *RGBNDSI*, *Snow-noSnow*, *Fixed Threshold*), due to over-smoothing. Namely $S_i^{s=1,t=1}$ was computed, but the same trend has been observed for different values of s and t.

Figure 5.4 shows the ROC curves obtained within the *PermaSense* dataset. Since the images in the *PermaSense* dataset represent a fragment of the *Matterhorn*, it was not possible to apply the skyline visibility score described in the Section 3.2. Hence, we used the statistical methods for the bad weather image filtering based on color analysis, proposed by the authors of the *GMM* method [149]. Table 5.3 shows the TPR obtained for the *PermaSense* dataset keeping the FPR fixed at 0.1. All the classifiers, with the exception of the *RF*, *SVM*, *LR* benefit from the spatio-temporal median filtering. Analogously to the *Webcams* dataset, all the methods benefit from the use of Daily Median Images.

Figure 5.5 depicts the ROC curves obtained on the *Photos* dataset. The major difference with respect to the other datasets is the fact that the *Fixed Threshold* method is dominated by all the others. These results are as expected, as the photographs were taken in different locations, with different cameras and different conditions. The *Fixed Threshold* method is not able to find a threshold value that is suitable for all images, while the other methods are more capable of adapting to varying conditions. In this case, we do not report the results as in Table 5.2 because median filtering and DMI can not be applied to this dataset that contains spatially and temporally independent images.

5.3.2 Computing Virtual Snow Indexes

The final goal of the described pipeline is to produce snow indexes (derived form the extracted snow masks) to be fed into environmental models.

Table 5.2: *TPR obtained in the* Webcams *dataset by the different snow classifiers.*

	without Median without DMI	without Median with DMI	with Median with DMI
Random Forest	79.7	**93.5** (+13.8)	**91.6** (+11.9)
SVM	**81.8**	92.6 (+10.8)	**91.6** (+9.8)
Linear Regression	78.3	89.7 (+11.4)	88.8 (+10.5)
GMM3	73.7	79.2 (+5.5)	82.7 (+9.0)
GMM2	79.9	80.2 (+0.3)	83.3 (+3.4)
RGBNDSI	72.9	87.1 (+14.2)	87.0 (+14.1)
Snow-noSnow	68.6	81.7 (+13.1)	80.5 (+11.9)
Fixed Threshold	76.0	88.4 (+12.4)	87.6 (+11.6)

Table 5.3: *TPR obtained in the* PermaSense *dataset by the different snow classifiers.*

	without Median without DMI	without Median with DMI	with Median with DMI
Random Forest	83.8	**90.2** (+6.4)	**89.2** (+5.4)
SVM	84.7	89.2 (+4.5)	88.7 (+4.0)
Linear Regression	**87.0**	87.9 (+0.9)	87.2 (+0.2)
GMM3	70.6	77.8 (+7.2)	83.2 (+12.6)
GMM2	60.2	78.6 (+18.4)	84.7 (+24.5)
RGBNDSI	73.6	78.3 (+4.7)	84.7 (+11.1)
Snow-noSnow	59.7	64.1 (+4.4)	87.1 (+27.4)
Fixed Threshold	66.5	78.8 (+12.3)	82.3 (+15.8)

Physical Snow Indexes

Thanks to the image geo-registration and orthorectification (using the associated topography data) it is possible to estimate the geographical properties of every pixel, such as its corresponding terrain area and altitude. Consequentially, it is possible to compute snow cover indexes that actually correspond to real-world measures. For example one may calculate the exact snow covered area expressed in square kilometers or the snow line altitude (the point above which snow and ice cover the ground) expressed in meters. As a proof of concept we devised an index that depicts the latter example, estimating the snow line altitude given a snow mask of a geo-registered mountain photograph.

Given a snow mask S, the area of the image covered by the mountain was split into N horizontal bands. Let A_i denote the altitude of the lowest pixel of the i-th horizontal band (where A_1 and A_N corresponds respectively to the lowest and highest altitude bands). We define *Snow Line Altitude (SLA)* index a vector of N elements, where SLA_i is a number in the range $[0, 1]$ that defines the fraction of the i-th horizontal band containing snow pixels. In other words, SLA can be seen as the snow cover percentage at different altitude levels. Given an SLA, we estimate the snow line altitude L as $L = A_k + SLA_k(A_{k+1} - A_k)$, for a value of k such that $SLA_{k-1} < \bar{s}$ and $SLA_k, \dots, SLA_N \geq \bar{s}$ where $\bar{s}$ is the threshold that defines the maximum negligible snow cover percentage.

Non-Physical Snow Indexes

The snow cover indexes, however, can also not correspond to a specific real-world measure. The absence of a intuitive representation of the snow index does not imply that it is not useful. As long as the index is correlated with the snow cover dynamics, a data-driven environmental model can benefit from being receiving such index in input. To prove this thesis we propose several indexes and environmentally evaluate them in Chapter 6.

From the snow mask S, the snow indexes are computed as follows. Let H denote a real value map having the same size as I, where $H(x, y)$ denotes the altitude of the terrain in the point that corresponds to $I(x, y)$ (e.g., altitude in meters, that can be estimated from the projection of the pixel on the DEM), for each (x, y) such that $M(x, y) = 1$ (i.e., for pixels representing points within the mountain region of the image). M and H can be obtained during the mountain peak identification phase, because each pixel of the photo gets aligned with a pixel of the virtual panorama; the latter carries information not only about the edges of the mountains, but also about the type (terrain/sky) and altitude of each DEM pixel. Let $\hat{H}$ denote the linearly normalized version of H, where the minimum/maximum altitude corresponds to 0/1. Then, a *virtual snow index* for an image I is defined as $\sum_{(x,y)|S(x,y)=1} VSI(x, y)$, where VSI is a virtual snow index function that transforms a pixel position into a snow relevance coefficient. We tested three different snow feature functions:

$$\begin{aligned} VSI^1(x, y) &= \hat{H}(x, y)^2 \\ VSI^2(x, y) &= \frac{\hat{H}(x, y)}{N_{x,y}} \\ VSI^3(x, y) &= 1 \end{aligned} \tag{5.1}$$

where N is the number of horizontal bands in which we split the mountain area of the image and $N_{x,y}$ denotes the number of mountain pixels belonging to the same band of a pixel $I(x, y)$. The first snow index weights each snow-covered pixel quadratically w.r.t. its altitude, the second one weights each snow-covered pixel linearly w.r.t. its altitude and normalizes the score w.r.t. the number of pixels contained in the same band, and the third one weights each snow-covered pixel uniformly regardless of its altitude. The values for the three indexes are computed for each day and their operational value is assessed in Section 6.3.

CHAPTER 6

Environmental Applications and Evaluations

In this chapter we describe the environmental evaluation of the virtual snow measures, which are obtained from the geo-registered web images as described in Section 5.3.2. Differently from the evaluations performed in the previous chapters, in which we study the accuracy of the proposed methods (e.g. how often a pixel classified as snow does actually represent snow), here we evaluate the practical environmental usefulness of the obtained data.

The work presented in this chapter is the result of the collaboration with environmental researches, co-authors of [23, 71].

A common approach for the environmental evaluation of non-authoritative data is to compare it against an authoritative environmental ground truth. Examples include: (i) comparing an estimated continental snow cover [178], vegetation cover [177] or cloud presence [132] map against satellite observations; (ii) comparing detected meteorological events against official government event lists [32]; (iii) comparing inferred snowfall and rainfall measurements against official meteorological data [100]. Such evaluations, however, are common to be subject of critics stating that the works propose a new way to replicate already existing data, while there is no proof of its usefulness. While these works usually provide justifications on why the new approach could be useful even if the authoritative data already exist (e.g. complement areas where satellite data is missing [177], faster event identification [77] or even weakness of satellites to space debris military attacks [177]) - the objective environmental utility of such data tends actually to remain unproven.

This phenomena puts the researchers in a vicious circle: the environmental monitoring community wants a proof that the proposed data is useful w.r.t. the existing sources, so it must be **novel** data not available from authoritative sources; however, proving that the proposed data is correct requires a comparison with an already existing ground truth.

To mitigate this limitation, instead of evaluating the *correctness* of the obtained data, we evaluate its practical *usefulness*. While the correctness is an objective quality, the usefulness, by definition, is subjective and is related to a specific context and a use case. Thus, we define a use case, in which we prove the authoritative environmental measurements to be relevant (a data-driven environmental model, which operation is influenced by snow dynamics). Furthermore, such model has a well defined performance metric. By adding Virtual Snow Indexes (VSI) to the model we are able to evaluate the operational value of the VSI, i.e. the performance impact triggered by the usage of VSI as input data.

Specifically, the operational value of the obtained VSI is assessed for a real world water management problem in the snow-dominated catchment of Lake Como, a regulated lake in Northern Italy, where snow melt is the most important contribution to the seasonal storage. The VSI operational value is quantified by comparing, via simulation, the performance of the lake operating policies designed using VSI and traditional snow information, with the performance of the baseline policy obtained by regulating the lake without snow information [70]. We define the model, apply an input selection procedure (in order to prove that the official snow information is actually relevant to the model) and evaluate the impact of the VSI. Numerical results show that such information is effective in extending the anticipation capacity of the lake operations, ultimately improving the system performance.

This form of assessment provides an indirect validation of the utility of web and crowdsourced information as the VSI extracted from web mountain images and the traditional observational data collected with dedicated tools are not always comparable directly due to the difference in their physical interpretation and spatio-temporal resolution (e.g., geo-located photos allow estimating the presence of snow, but not the physical measures usually employed in snow process models, such as the snow water equivalent).

The rest of this chapter is structures as follows: Section 6.1 introduces the snow process monitoring problem and how it is managed in the state of the art; Section 6.2 reports two preliminary qualitative experiments that study the correlation between the extracted snow information and the authoritative data; Section 6.3 describes how we assess the operational value (i.e. the environmental usefulness) of VSI; Section 6.4 provides the details of the case study and Section 6.5 reports the results.

6.1 Snow Processes Monitoring

Snow is a key component of the hydrologic cycle in many regions of the world. Despite recent advances in environmental monitoring are making a wide range of data available, continuous snow monitoring systems able to collect data at high spatial and temporal resolution are not well established yet, especially in inaccessible high latitude or mountainous regions.

Snow accumulation and melting are fundamental components of the hydrological cycle in many watersheds across the world (e.g., [88, 129]). Approximately 40-50% of the Northern Hemisphere is covered by snow [141] and snow plays a key role in mountain areas, which, in Europe, account for 40% of the total surface [157].

In such contexts, an accurate characterization of snow availability and its evolution in time can be extremely valuable for a variety of operational purposes, from avalanche prediction (e.g., [142, 158]), water systems operations through medium to long-term streamflow forecast (e.g., [1, 181]), or drought risk management (e.g., [169]). The projected temperature increase induced by climate change, with consequent reductions of large volumes of snowpack and acceleration of the water cycle in many mountainous areas, will further amplify the importance of better understanding snow dynamics [6, 105].

Snow processes are generally monitored through both ground monitoring networks (e.g., [16,121]) and remote sensing (for a review, see [45,101] and references therein). Yet, both sources have serious limitations in alpine contexts mainly related to the high spatial (e.g., [133]) and temporal variability of snow related processes [10, 47, 72]. Ground stations are generally very coarsely distributed. Satellite products provide data on a denser grid but are diversely constrained depending on the sensors installed [131]. High spatial and temporal resolution imagery (i.e., daily maps with spatial resolution of about 500 m) can be derived from Moderate Resolution Imaging Spectroradiometer (MODIS) products, which are, however, strongly affected by the weather, because optical sensors cannot see the earth surface when clouds are present [139]. Space-board passive microwave radiometers (e.g., AMSR-E) penetrate clouds but have coarse spatial resolution (25 km). Finally, the use of active microwave systems (e.g. RADARSAT) is so far limited to the detection of liquid water content.

The last few years have seen a rising interest in complementing traditional observations by using cameras and short-range visual content analysis techniques [14], which allow improving the temporal and spatial resolutions for specific applications. Many case studies showed that the use of one or several time-lapse cameras allows mapping both the spatial and temporal patterns of a variety of snow characteristics, including glacier velocity, snow cover changes, or detailed monitoring of snowfall interception (see [140] and references therein). However, most of these systems generally rely on cameras designed and positioned ad hoc (e.g., [84]), possibly including in the camera view some specific objects, such as flags or sticks, which simplifies the calibration of geometry and colors (e.g., [59,64,106]). In addition, the use of these cameras is generally very expensive and often requires intensive manual efforts in the image processing phase. This latter includes a variety of crucial, time-consuming operations, such as the selection of photographs with good meteorological and visibility conditions, the photo-to-terrain alignment and orientation, and the labeling of snow covered pixels for estimating the total snow cover (e.g., [42,49]).

6.2 Qualitative Experiments

We performed two qualitative analyses of the virtual snow cover estimations to show the consistency with other environmental variables. Section 6.2.1 reports tests performed on non-physical virtual measures (VSI, defined in Section 5.3.2), while Section 6.2.2 reports tests performed on physical virtual measures (SLA, defined in Section 5.3.2).

6.2.1 Virtual Snow Index

A qualitative analysis of the Virtual Snow Index (VSI) can be performed by comparatively analyzing the trajectory of the VSI with respect to the snow height observations in the closest ground station or with respect to some physical variables closely related to the snow dynamics. We calculated the VSI (using VSI^3 as defined in Equation 5.1) of a single webcam placed in Northern Italy. Figure 6.1 contrasts the historical trajectory of the VSI in 2013 with the trajectories of the snow height observations (left) at Oga San Colombano station that is located around 15 km far from the webcam, and with the freezing level registered in the area (right). Despite some differences due to the different locations of the webcam and the ground station, the first comparison shows similar temporal patterns: most of the snowmelt occurs between April and first half of May, followed by a late snowfall at the end of May; no snow is present since late June, with the first snowfall of the next winter observed in early October. The comparison between the VSI and the freezing level shows a negative correlation between low values of freezing level from January to March as well as in November and December, which are associated to high values of the VSI. On the contrary, the freezing level increases in summer time in correspondence to low and zero values of the VSI. Moreover, it is worth noting the consistency in the oscillations of the two trajectories especially in winter time, when the snow accumulation is captured by increasing values of the VSI associated to decreasing freezing levels and, viceversa, the snow melting corresponds to decreasing values of the VSI and increasing freezing levels.

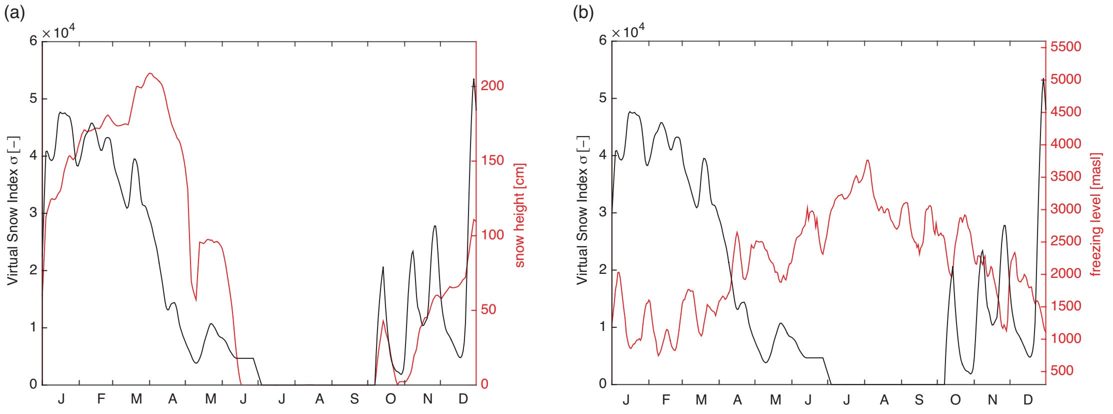


Figure 6.1: *Comparison of the trajectories during 2013 of the VSI with the snow height measured at Oga San Colombano (left) and with the freezing level (right).*

6.2.2 Snow Line Altitude

As a second test, we studied the snow line altitude, i.e. the minimum elevation at which snow is present. We studied the snow line altitude dynamics for one of the webcams used in the *Webcams* dataset (described in Table 5.1). We acquired 40 k images during a two month period going from May 15th, 2015 to July 14th, 2015. For 49 days (out of the 61 days of the monitored period) at least one good weather image was retrieved and the corresponding Daily Median Image (DMI) was generated as described in Section 3.2.3. Then, a snow mask was extracted for each DMI and the missing day snow masks estimated by the interpolation described in Section 5.2. We performed the photo-to-terrain orthorectification that allowed us to estimate the altitude of every pixel on the webcam photographs. Finally, the Snow Line Altitude (SLA) was

estimated for each observed day as described in Section 5.3.2.
Figure 6.2 shows the trend of the SLA (smoothed by a median filter with the window size equal to 4 days), along with the air temperature registered by a nearby ground station. It can be observed that the snow melting process was characterized by four occurrences when the snow level altitude increased abruptly. This behavior is correlated with the four temperature peaks observed by the meteorological station. Furthermore, the obtained snow line trend can not be computed through the traditional snow monitoring techniques, due to the low temporal frequency of the satellite imagery and the low number of physical measurement stations placed at such high altitudes. This example shows a possible application of the snow cover estimation based on public visual content, and confirms the consistency of the proposed methods and metrics.

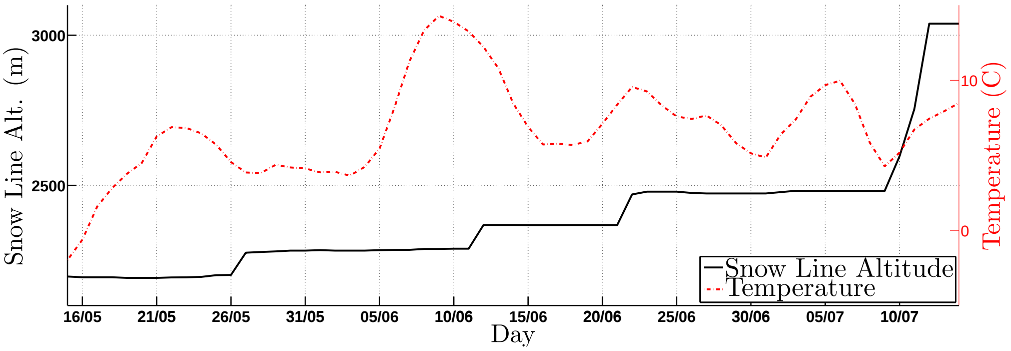


Figure 6.2: *The snow level altitude and the temperature trends during the observation period.*

6.3 Virtual Snow Indexes Operational Value Assessment

The assessment of the operational value of virtual snow measures is done by comparing the performance of alternative operating control policies for the regulation of Lake Como. A control *policy* is a function returning the quantity of water to be released u_t at each time instant $t = [0 \dots T-1]$, as dependent upon an information vector $\mathbf{z}_t$, i.e., $u_t = p(\mathbf{z}_t)$. In our study we employ the following policies:

- Perfect Control Policy (PCP): an ideal policy used as an upper bound of the system performance, which makes always the optimal decision based on perfect knowledge of current and future system conditions.

- Baseline Control Policy (BCP): considers only limited information (day of the year and current lake level).

Furthermore, we employ Informed Control Policies (ICPs), which are defined by extending the input $\mathbf{z}_t$ of the BCP with the selected information, i.e., $\mathbf{z}_t = (t, l_t, \mathbf{I}_t)$. Specifically, we study the following ICP:

- ICP_{SH} : considers BCP inputs and the official snow height observations received from ground measurement stations in the region of interest.

- ICP_{SWE} : considers BCP inputs and the Snow Water Equivalent (SWE), an authoritative snow quantification produces by the Italian Environmental Protection Agency (ARPA) thanks to ground measurement stations and satellite data.
- ICP_{VSI} : considers BCP inputs and the Virtual Snow Indexes (VSI) produced by the pipelines described in this work.
- $ICP_{SWE+VSI}$: considers BCP inputs, SWE input and VSI input.

The assessment quantifies how closer the ICPs get to the perfect one, in comparison to the baseline. This evaluation methodology, called Information Selection and Assessment (ISA) framework [70], consists of 3 steps that are described in the next subsections:

- Quantification of the expected value of perfect information, i.e., the potential for improving operations under the assumption of perfect knowledge of future conditions (Section 6.3.1).
- Automatic selection of the most valuable information to improve current operations (Section 6.3.2).
- Evaluation of the selected information on the resulting control policy performance (Section 6.3.3).

6.3.1 Expected Value of Perfect Information

The Expected Value of Perfect Information (EVPI) is the performance gain that can be achieved under the assumption of perfect foresight on the future [184]. If the value of EVPI is small, a limited information policy already performs close to the best strategy and thus the benefit of additional input data approximating future system conditions is limited.

The availability of perfect knowledge of the future external drivers (e.g., lake inflows) is equivalent to assume that the operator is an omniscient oracle implementing a Perfect Control Policy (PCP), consisting of an optimal sequence of release decisions $u^{PCP}_{[0,T-1]}$, conditioned on the current system status (i.e., the time instant t and the current lake level l_t), and on the perfect knowledge of the future inflows. In the experiments illustrated below, the PCP is built by solving the control policy design problem over a 2-year horizon in which the sequence of inflows is known. This is a standard nonlinear optimization problem and can be solved by either a local optimization method (e.g., gradient-based) or a global optimization method (e.g., direct search). Alternatively, since the objective functions in our application are time-separable, we use deterministic dynamic programming (DDP), which is more efficient and provides an almost exact solution.

PCP performance ($\mathbf{J}^{PCP}$) has a relative value, because it depends on the physical characteristics of the system, e.g., the ratio between the lake capacity and its inflow. The EVPI has, hence, to be estimated as the distance between $\mathbf{J}^{PCP}$ and the performance of a Baseline Control Policy (BCP), defined as a simple closed-loop control policy, where $\mathbf{z}_t$ includes the time index t and the current lake level l_t.

In a single-objective scenario, the EVPI is simply the difference between the (scalar) performance of the PCP and BCP . In a multi-objective case, the evaluation is more complex; the performance objectives $\mathbf{J}^{PCP}$ and $\mathbf{J}^{BCP}$ are vector functions and the solution is not unique, but rather a set of Pareto optimal solutions (Pareto Front). Among the commonly used metrics (see [123]), the ISA framework adopts the hypervolume indicator (HV), which captures both the proximity of the Pareto Front $\mathbf{J}^{BCP}$ to the ideal one $\mathbf{J}^{PCP}$ as well as the distribution of the BCP solutions in the objective space. The hypervolume measures the volume of objective space dominated ($\preceq$) by the considered set of solutions (S). Then, the HV indicator is defined as the ratio of the hypervolumes of the solutions produced by BCP and PCP :

$$HV(BCP, PCP) = \frac{\int \alpha(\mathbf{s}_{BCP}) ds_{BCP}}{\int \alpha(\mathbf{s}_{PCP}) ds_{PCP}} \tag{6.1a}$$

with

$$\alpha(\mathbf{s}) = \begin{cases} 1 & \text{if } \exists s' \in S \text{ such that } s' \preceq s \\ 0 & \text{otherwise} \end{cases} \tag{6.1b}$$

If policy A has a value of HV greater than a policy B, the solutions produced by A dominate a larger fraction of the objective space, which means that A is better than B in pursuing the multiple objectives of the system. The EVPI can then be computed as the difference between the HV of PCP (i.e., 1 by definition) and the HV of BCP .

6.3.2 Most Valuable Information Selection

A large value of EVPI indicates that a control policy endowed with more information can approach the performance of the ideal, omniscient one. The ISA methodology helps identify the input information that enables the informed policy to approximate as much as possible the optimal sequence of decisions $u^{PCP}_{[0,T-1]}$.

The set Ξ_t of candidate inputs may comprise *exogenous variables*, i.e., variables that are observed in the time interval $[0, T-1]$ but are not part of the problem formulation; examples are rainfall, temperature, snow presence, etc. Since Ξ_t can comprise redundant and collinear variables, its smallest subset $\mathbf{I}_t \in \Xi_t$ that carries the most valuable information must be identified, as the one that best explains the optimal sequence of decisions. Several techniques can be used to solve this feature selection problem [62], such as cross-correlation analysis, mutual information analysis, or input variable selection methods. We use the hybrid model-based/model-free Iterative Input Selection (IIS) algorithm (Algorithm 1), which can approximate strongly non-linear functions and scale to large datasets made of long time series and many candidate variables [62]. Given a generic output variable v^o and the set of candidate inputs $\mathbf{v}^i$, IIS first ranks the inputs w.r.t. a statistical measure of significance and adds the best performing input v^* to the current set of selected variables $\mathcal{V}$. This step avoids the inclusion of redundant variables: after an input is selected, all the other inputs highly correlated with it will rank low in the next iterations. Then, the algorithm estimates a model of v^o with input $\mathcal{V}$, such that $v^0 = \hat{m}(\mathcal{V})$, and estimates the model performance with a distance metric D (e.g., the coefficient of determination) as well as the model residuals ($v^o - \hat{m}(\mathcal{V})$), which become the new output at the next iteration. The algorithm stops when the next best input variable selected is already in the set $\mathcal{V}$, or when over-fitting conditions are reached. Among the many alternative model classes, IIS relies on extremely randomized trees (Extra-Trees), a tree-based method proposed by [65] that was empirically

Algorithm 1 Iterative Input Selection

```
Inputs: a dataset $\mathcal{F}$ of candidate inputs $\mathbf{v}^i$ and
the output variable to explain $v^o$.
Initialization:
Set $\mathcal{V} \leftarrow 0, \hat{v}^o \leftarrow v^o, D_{old} \leftarrow 0$
Iterations: repeat until stopping conditions are met
- select the most relevant input $v^* \in \mathbf{v}^i$ to explain $\hat{v}^o$
- if $v^* \in \mathcal{V}$, return $\mathcal{V}$ endif
- $\mathcal{V} \leftarrow \mathcal{V} \cup v^*$
- $\hat{m}(\cdot) \leftarrow$ Extra-Trees$(\mathcal{F}, v^o, \mathcal{V})$
- $\hat{v}^o \leftarrow v^o - \hat{m}(\cdot)$
- $\Delta D \leftarrow D(v^o, \hat{m}(\cdot)) - D_{old}$
- $D_{old} \leftarrow D(v^o, \hat{m}(\cdot))$
- until $\Delta D < \varepsilon_D$
return $\mathcal{V}$
```

demonstrated to outperform other models in terms of modeling flexibility, efficiency, and scalability with respect to the input dimensionality. Moreover, Extra-Trees structures can be exploited to infer the relative importance of variables, as required for their ranking [22].

6.3.3 Expected Value of Sample Information

After selecting the most valuable information $\mathbf{I}_t \subset \Xi_t$, the next step is to design the Informed Control Policies (ICPs) that exploits such information to make decisions. As mentioned in Section 6.3 we define four ICPs (ICP_{SH} , ICP_{SWE} , ICP_{VSI} and $ICP_{SWE+VSI}$) and search the optimal control policy with approximate dynamic programming methods.

We use the evolutionary multi-objective direct policy search (EMODPS), a simulation-based technique that combines direct policy search, nonlinear approximating networks, and multi-objective evolutionary algorithms [67]. EMODPS exploits the parameterization of the control policies p_θ and explores the parameter space Θ to find a policy (p_θ^*) that optimizes the expected system performance (J_θ, conventionally assumed to be a cost), i.e., $p_\theta^* = \arg\min_{p_\theta} J_\theta$ where the policy p_θ is parameterized by parameters $\theta \in \Theta$ and the problem is constrained by the dynamics of the system. Finding p_θ^* is equivalent to finding the corresponding optimal policy parameters θ^*. A tabular version of the EMODPS method is illustrated in Algorithm 2.

In general, we expect the ICP to fill the performance gap between the upper and lower bound solutions (i.e., the PCP and BCP), and to produce a performance $\mathbf{J}^{ICP}$ as close as possible to $\mathbf{J}^{PCP}$. The benefit associated to the use of the selected information is called *Expected Value of Sample Information* (EVSI) and can be quantified by means of the same metrics used for the evaluating the EVPI (see Section 6.3.1). However, since the relative contribution of each component of $\mathbf{I}_t$ to the ICP performance might not be equivalent to the relative contribution in explaining the optimal sequence $u_{[0,T-1]}^{PCP}$, which is the metric on which information selection is performed, the ISA procedure is applied iteratively. At first, we consider only the first candidate variable selected by the IIS algorithm, assuming that it also has the highest potential to improve the lake op-

Algorithm 2 Evolutionary Multi-Objective Direct Policy Search.

```
Initialization:
Generate a random parameter values population $\{\theta^1, \dots, \theta^P\}$
Iterations: repeat until stopping conditions are met
- generate a trajectory $\tau^i$ via model simulation according to the
  stochastic transition function $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, \varepsilon_{t+1})$
  and following the policy $p_{\theta^i}$ (with $i = 1, \dots, P$)
- compute performance $J^1_{\theta^i}, \dots, J^q_{\theta^i}$, with $i = 1, \dots, P$
- generate new population by selection, crossover and mutation
  w.r.t. the best individuals (i.e., non Pareto-dominated solutions)
```

erations. We design informed policies conditioned on this variable only, and estimate the corresponding EVSI by comparison with the PCP and BCP performance. We then iterate the procedure by incrementally adding variables to the exogenous information vector $\mathbf{I}_t$, designing the associated policy, and evaluating the corresponding EVSI. When either the attained performance is satisfactory or the marginal improvement in the EVSI between two consecutive iterations is negligible, the procedure ends.

6.4 Case Study: Lake Regulation with Virtual Snow Indexes

Lake Como is a regulated lake in the Adda River basin, Italy (Figure 6.3). The lake has an active storage capacity of 254 Mm^3 and is fed by a 3 500 km^2 alpine catchment that reaches altitudes over 4 000 m asl. Downstream from the lake, the Adda River serves a dense network of irrigation canals belonging to four agricultural districts for a total irrigated area of 1 400 km^2 (green area in Figure 6.3). Major cultivated crops are maize and temporary grasslands, while minor crops include rice, soybean, wheat, tomato, and barley. The hydro-meteorological regime in the catchment is the typical sub-alpine one, with scarce discharge in winter and summer, and peaks in late spring and autumn due to snowmelt and rainfall, respectively. In particular, snowmelt from May to July is the most important contribution to the formation of the seasonal storage (Figure 6.4).

The alpine orography constrains the accurate monitoring of snow dynamics. The existing ground stations (46 over the 10 500 km^2 alpine area in the Lombardy region) provide a very coarse coverage of the region and are not sufficient to reliably monitor the snow coverage and the associated water content. This is instead estimated by the Regional Agency for Environmental Protection (Agenzia Regionale per la Protezione dell'Ambiente - ARPA[1]), which produces estimates of Snow Water Equivalent (SWE) through a hybrid procedure combining snow height and temperature data from ground stations, measures of snow density in few specific locations, satellite-retrieved data of snow cover (ARPA currently adopts MODIS [139]), and model outputs for spatially interpolating these data. As a result of this complex procedure, ARPA elaborates a weekly estimate of SWE. Such reports are delivered only weekly due to the well known limitations of snow products derived from optical sensors associated to the frequent satellite occlusion by cloud coverage. This limitation is particularly restrictive in the alpine region, where previous studies observed an average cloud occlusion of 63 % over a

[1] www.arpalombardia.it

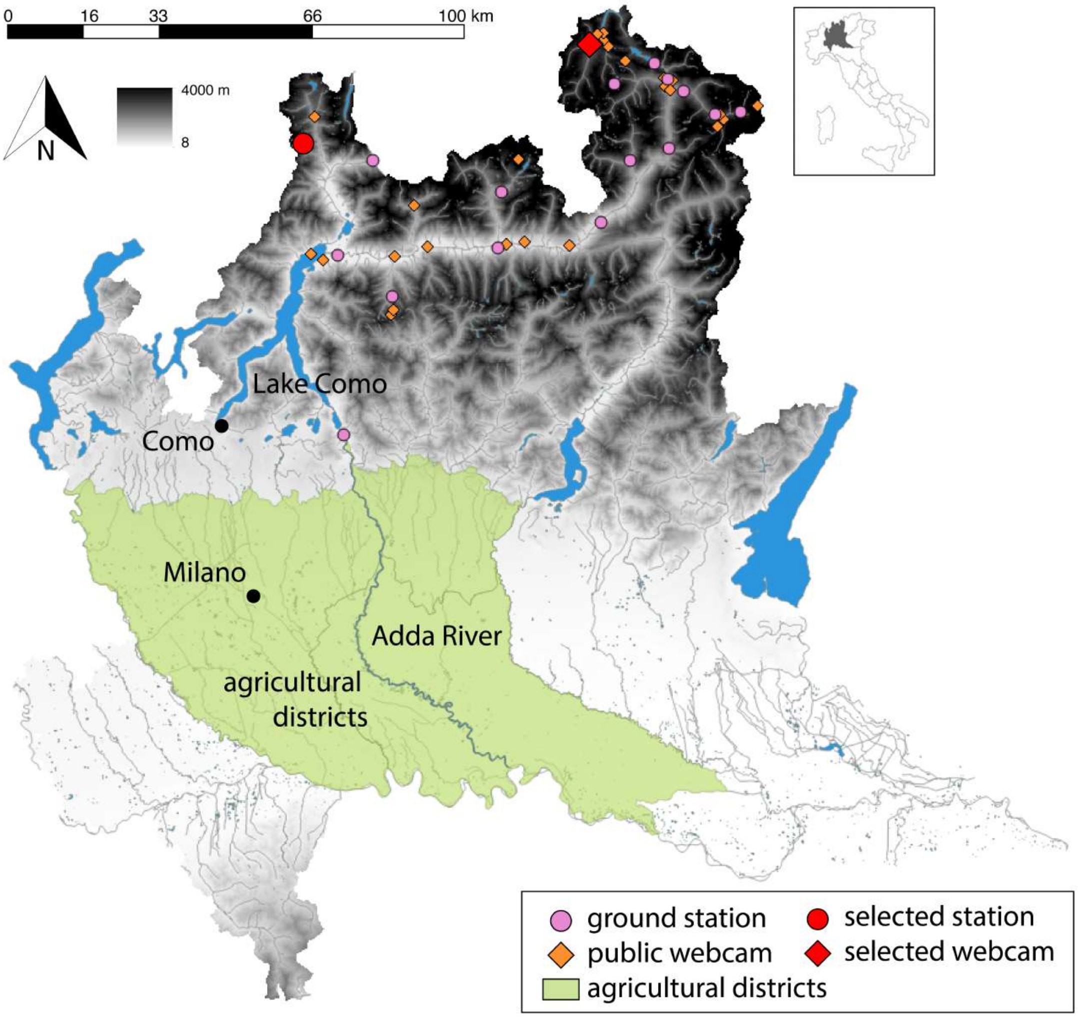


Figure 6.3: *Adda River Basin: Lake Como, Adda River, downstream agricultural districts, ground stations, and public webcams.*

five year monitoring period [138], with critical episodes of cloud coverage lasting for more than 25 days per month in winter time. On the contrary, webcams are less affected by cloud coverage and can provide observations during cloudy days as shown illustratively in Figure 6.5. In this study, we contrast the operational value in informing the lake operation of four different snow-related data sources: (i) daily observations of snow height from coarsely distributed ground stations; (ii) weekly SWE estimate provided by ARPA; (iii) daily values of the VSI extracted from public web images; (iv) both SWE and VSI data.

The existing regulation of the lake is driven by two primary, competing objectives: water supply, mainly for irrigation, and flood control in the city of Como, which is the lowest point of the lake shoreline. In particular, the agricultural districts downstream would like to store the snowmelt volume for the summer water demand peak, when the natural inflow is not sufficient to satisfy the irrigation requirements (see the magenta area in Figure 6.4). Yet, storing such water increases the lake level and, consequently,

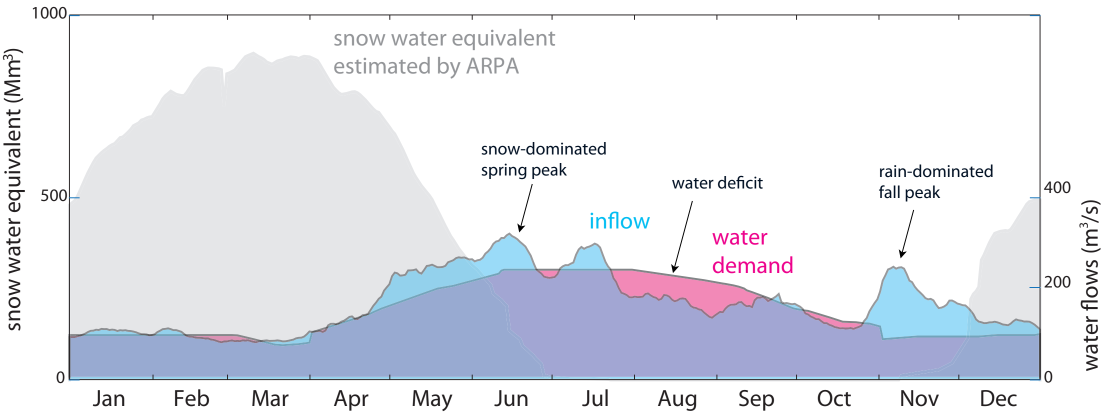


Figure 6.4: *Hydro-meteorological regime of Lake Como.*

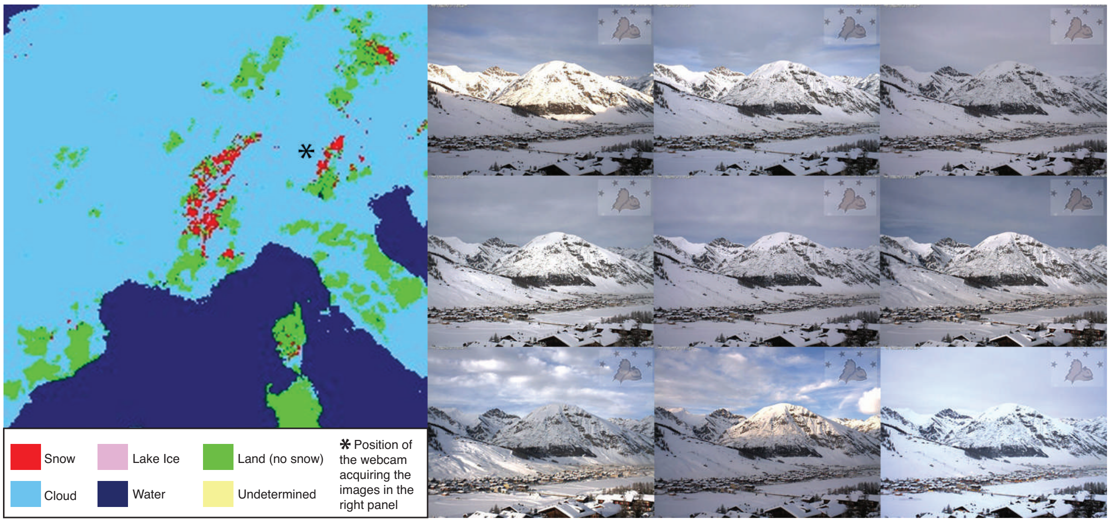

Snow
Lake Ice
Land (no snow)
Cloud
Water
Undetermined
✱ Position of the webcam acquiring the images in the right panel

Figure 6.5: *Comparison of MODIS daily snow cover map (left panel) with the images acquired by a webcam (right panel) on Jan. 9, 2014 at the location denoted by the asterisk in the map.*

the flood risk, which would be instead minimized by keeping the lake level as low as possible. On the basis of the existing works e.g. [21, 66], these two objectives are formulated as follows:

- *Flood control*: the average annual number of flooding days in the evaluation horizon H, defined as days when the lake level h_t is higher than the flooding threshold ($\bar{h}$=1.24 m):

$$J^{flood} = \frac{1}{H/365}\sum_{t=1}^{H}\Lambda(h_t) \quad \text{where} \qquad (6.2)$$
$$\Lambda(h_t) = \begin{cases} 1 & \text{if } h_t > \bar{h} \\ 0 & \text{otherwise} \end{cases}$$

- *Irrigation supply*: the daily average quadratic water deficit between the lake release r_{t+1} and the daily water demand w_t of the downstream system, subject to the minimum environmental flow constraint q^{MEF} to ensure adequate environmental conditions in the Adda River:

$$J^{irr} = \frac{1}{H}\sum_{t=1}^{H} \max\left(w_t - \max(r_{t+1} - q^{MEF}, 0), 0\right)^2 \tag{6.3}$$

  This quadratic formulation aims to penalize severe deficits in a single time step, while allowing for more frequent, small shortages [83].

6.4.1 Content Selection for the Case Study

Our dataset contains more than 100 M mountains images located across the Lake Como catchment; such images are produced by 2 k webcams and include also more than 600 k photos produced by users. However, not all relevant images are directly exploitable for the assessment; environmental models require a very long observation period: a statistically significant evaluation *requires observations spanning multiple years*, to cope with seasonal effects, thus, our sources lack a long enough time series to be usable as input. A manual search found historical images of a few webcams and aggregated them to the dataset. Specifically, we found one webcam with enough historical data, which was chosen for the experiments. The webcam is placed in Livigno, while the snow height data are measured at the Truzzo ground station (see Figure 6.3). The mountain framed by the webcam is positioned inside the Como Lake catchment and its snow level is known to affect the lake water dynamics. Even with a single webcam, the experimental results described in this chapter demonstrate a significant utility of the (virtual) snow-related data.

6.4.2 Policy Design

The selected policy parameterization strongly influences the choice of the optimization approach. In scenarios where the complexity of the policy parameterization, and consequently the number of parameters to optimize, is high, evolutionary algorithms (EAs) have been successfully applied, also in presence of high-dimensional decision spaces and noisy and multimodal objective functions [18]. In our framework, the Informed Control Policies (ICPs) are designed via EMODPS by parameterizing the policies as Gaussian radial basis functions, which have been demonstrated to be effective in solving this type of multi-objective policy design problems [68, 69], particularly when exogenous information is used for conditioning the operations [70]. To perform the optimization, we use the self-adaptive Borg MOEA [80], which has been shown to be highly robust in solving multi-objective optimal control problems, where it met or exceeded the performance of other state-of-the-art MOEAs [185]. Each optimization was run for 2 million function evaluations. To improve solution diversity and avoid dependence on randomness, the solution set from each formulation is the result of 30 random optimization trials. The final set of Pareto optimal policies for each experiment is defined as the set of non-dominated solutions from the results of all the optimization trials.

The ideal set of operating policies (PCP), which assume perfect foresight of future inflows, were designed via Deterministic Dynamic Programming. The weighting method is used to aggregate the two operating objectives (i.e., flood control and irrigation) into a single objective, via convex combination.
The traditional baseline regulation of the lake (BCP) is represented in terms of a set of operating policies conditioned on the day of the year d_t and on the lake level h_t. Also these policies were designed via EMODPS.

6.5 Results and Discussion

The performance of the set of Informed Control Policies (ICPs) is contrasted with the baseline solution (BCP), namely the traditional lake regulation conditioned on the day of the year and the lake level, and the upper bound solution, namely an ideal set of policies (PCP) designed under the assumption of perfect foresight of future inflows. We assess the operational value of authoritative snow measures (ICP_{SH} and ICP_{SWE}), then we evaluate the operational value of virtual snow measures (ICP_{VSI}). Finally, we asses the potential of the VSI to *complement* the authoritative data, by validating the performance of $ICP_{SWE+VSI}$, which uses both SWE and VSI inputs.

Quantifying the EVPI

The first step of the ISA framework (Section 6.3) estimates the Expected Value of Perfect Information by contrasting the PCP and the BCP . Figure 6.6 shows the performance of the PCP (represented by black squares) evaluated over the horizon 2013-2014. The black circles represent the performance of the BCP (traditional control policies conditioned on the day of the year and the lake level). The performance of the Informed Control Policies is discussed later on in this section.
Both axis are to be intended as *lower is better*, with the best solution located in the bottom-left corner of the figure. Visual comparison of the Pareto Fronts shows that the potential for improvement over the BCP is large: the gap between basic information and the perfect knowledge is substantial in terms of both operating objectives, as represented by the area between the line passing through the black squares and the black circles.
Quantitative EVPI assessment is provided by the HV indicator in Table 6.2, where the difference between BCP and PCP is equal to 0.29, confirming the gap between Perfect and Baseline Control Policies and consequentially the large potential improvement for IPCs.

Exogenous Variables Selection

The question whether snow information can help making more informed decisions is addressed by using the ISA framework to identify the most informative exogenous variables $\mathbf{I}_t \subset \Xi_t$. We first evaluate the day of the year and the lake level together will all the exogenous variables except our virtual snow indexes, to check if the official snow information is relevant. The rationale for retaining the day of the year and the lake level is to extend the BCP and avoid selecting exogenous variables correlated with t or l_t. We perform 20 runs of the IIS algorithm to filter the randomness associated to the construction of the Extra-Trees models. Despite the limited length of the time

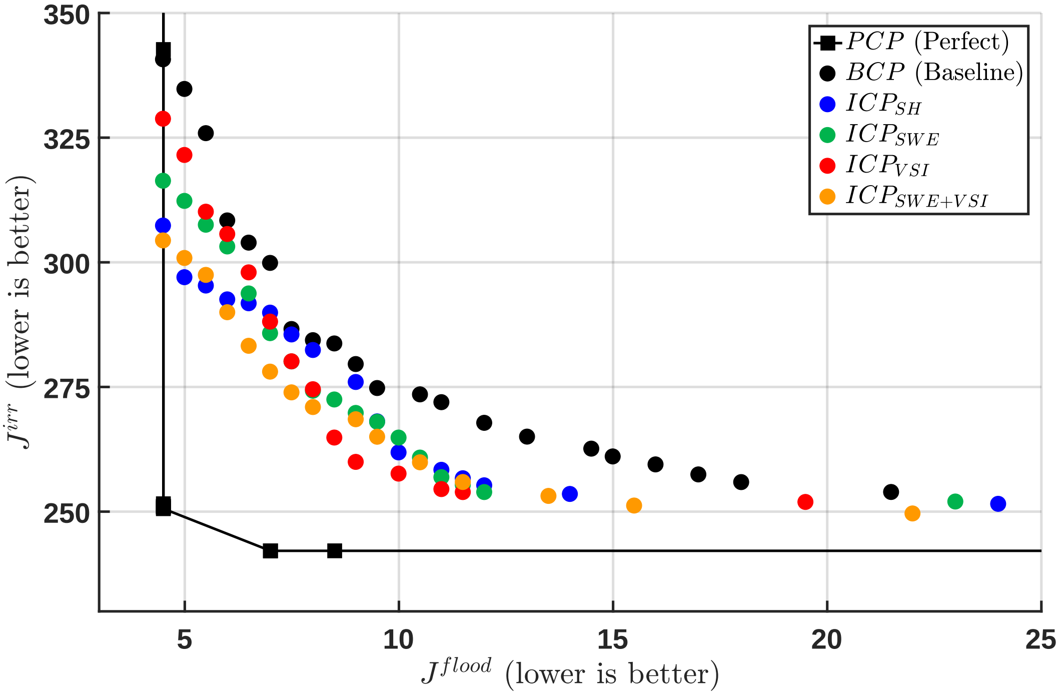


Figure 6.6: *Pareto front of the performance in terms of irrigation supply and flood control of the different operational policies.*

series (only 2 years), which introduces some variability across the runs, the best result consistently selects as most valuable information the day of the year (t), the lake level (l_t), and the official SWE estimated provided by ARPA (SWE_t). This confirms the key role of snow dynamics in the Lake Como system. Using the 3 variables selected, the Extra-Trees model approximates the optimal sequence of release decisions $u^{PCP}_{[0,T-1]}$ attaining a model performance, evaluated using the coefficient of determination (D), equal to 0.639. Table 6.1 shows the contributions of variables and statistics over the 20 repetitions.

Table 6.1: *Variables selected by the IIS algorithm.*

Variable	Best-run	Selection frequency, 20 runs	Average contribution (R2), 20 runs
t	0.231	100%	0.334
l_t	0.348	85%	0.194
SWE_t	0.060	45%	0.108

Benefits of Official Snow Information

The persistence of the day of the year (t) and the lake level (l_t) as the first two most relevant variables is not surprising, given the strong influence of the seasonality on both the hydro-meteorological regime and the water demand. This information is the only one used in the BCP (black points in Figure 6.6). To quantify the value of the *official* snow measures, we assess the performance of the ICP_{SH} and ICP_{SWE} conditioned on the information vectors $\mathbf{z}_t = (t, l_t, SH_t)$ and $\mathbf{z}_t = (t, l_t, SWE_t)$ respectively. The resulting ICP_{SH} and ICP_{SWE} policies are represented in Figure 6.6 by blue and green points respectively. The comparison of the performance of the BCP and these policies shows a relevant contribution of the official snow information: this is due to an improved medium-long term foresight, as snow information provides useful insight about the expected water availability during the next summer, when the irrigation demand is high and the conflict between flooding and irrigation is critical. A quantitative evaluation of the EVSI associated to this variable is given by HV reported in Table 6.2: HV increases from 0.7079 to 0.7881 (i.e. +11.3 %) when moving from BCP to ICP_{SH} and slightly less for ICP_{SWE} : 0.7848 (i.e., +10.9 %).

Benefits of Virtual Snow Indexes

The question of whether the Virtual Snow Indexes (VSI^k, defined in Equation 5.1) can be useful w.r.t. the official ARPA data is addressed through the following experiments: first, we replace the ARPA SWE variable with each virtual snow index VSI^k and analyze the performance of the corresponding policy ICP_{VSI^k} . Second, we explore the possibility of complementing - instead of replacing - the ARPA SWE variable with the VSI, by evaluating the performance of $ICP_{SWE+VSI^k}$, conditioned on the information vector $\mathbf{z}_t = (t, l_t, SWE_t, VSI^k_t)$. We report only the results obtained using the third snow index VSI^3 (henceforth, simply VSI), because it consistently outperforms the other two indexes.

Figure 6.6 shows that the performance of the VSI is comparable to, and sometimes higher than, the one of the ARPA SWE and snow height measurements. In fact, the red Pareto Front intersects the green one and the blue one, with some ICP_{VSI} solutions outperforming the ICP_{SH} and ICP_{SWE} points. This observation is confirmed by the corresponding values of HV reported in Table 6.2, with ICP_{VSI} obtaining a higher value than the two policies that use official snow information, which corresponds to a 11.6 % improvement with respect to the BCP .

Finally, the performance of $ICP_{SWE+VSI}$, conditioned on both SWE_t and VSI (orange points in Figure 6.6) outperforms both BCP and all the other ICPs. Such superiority is certified by the values of HV: $ICP_{SWE+VSI}$ obtains an HV value equal to 0.8158, which corresponds to a 15.2 % improvement with respect to the BCP and a 4 % improvement with respect to ICP_{SWE} .

Table 6.2: *Quantification of Expected Value of Perfect and Sample Information in terms of hypervolume indicator.*

Policy	HV	ΔHV
BCP (Baseline)	0.7079	–
ICP_{SH} (Snow Height)	0.7881	+11.3%
ICP_{SWE}	0.7848	+10.9%
ICP_{VSI}	0.7898	+11.6%
$ICP_{SWE+VSI}$	0.8158	+15.2%
PCP (Perfect)	1.0	–

Results Conclusions

In this section we described the results obtained by different control policies in terms of a visual Pareto front observation and in terms of a quantified hypervolume measure. We proved that following claims are true for the defined case study:

- the potential improvement w.r.t. the baseline control policy is large;
- official snow information is ranked as highly relevant by the variable selection algorithm;
- official snow information improves the performance of the control policy by more than 10 %;
- virtual snow information is able to successfully replace the official snow information with a slight performance improvement;
- virtual snow information improves the performance even when the control policy is already using the official snow information, thus, the two are not duplicates.

These results suggest a significant potential for complementing the official snow estimations with virtual snow indexes derived from public web media. No orthorectification was performed since the simplest virtual snow index (all snow pixels contribute with the same score) resulted in the best performance.

CHAPTER 7

Crowdsourcing and Citizen Science

Multimedia processing is a highly competitive research field, where software systems need to emulate the human capacity of recognizing the meaning of objects, a skill matured in millions of years of evolution. Several state of the art works [12,61] advocate a quantum leap in the openness of multimedia processing platforms and the involvement of human beings in all multimedia processing phases as the key factors for innovation in next-generation multimedia processing applications.

In this chapter we describe how *putting humans in the loop* [60] can improve the automatic environmental monitoring approach presented in this thesis. We propose a human computation approach, whereby the contribution of human performers is integrated within processing pipelines, to support tasks where human judgment is superior or complementary to the pure algorithmic approach. We empower the fully automatic visual data processing with a human computation that combines the conventionally isolated areas of crowdsourcing, social network and gaming with a purpose.

According to [61], the intervention of humans in the computation can occur at three levels:

- *implicit*, in which the computerized system directly harnesses the sensing capacity of humans, e.g., by using biometry to register the unconscious reactions caused by the exposition to content;
- *decisional*, in which the computerized system exploits the explicit rationality of individuals, e.g., in knowledge extraction or result comparison/evaluation;
- *social*, in which the computation exploits the capability of humans to work cooperatively towards the achievement of complex tasks in a more efficient and faster manner, e.g., by distributing across large communities micro-tasks in knowledge extraction or result evaluation.

Specifically, we explore the potentiality of using the decisional and social computations thanks to the diffusion of mobile devices, social networks and online games that has spawned a novel generation of hybrid applications, associated under the generic label of *citizen science*, which harness the online, voluntary contribution and cooperation of common people for the resolution of complex tasks in environment monitoring and a variety of other domains, including computer vision, transport, bio-medical research, and more [126]. The common traits of these applications include:

- The use of people as soft sensors, to acquire data about the physical environment to be monitored or analyzed (e.g., mobility routes for traffic management, photos of hazardous events for disaster monitoring). Such soft sensing could happen on demand, by proactively engaging people to collect data upon necessity, or passively, by collecting traces of people's activities (e.g, phone calls, tweets, photos) produced originally for other purposes.
- The fusion of heterogeneous data, coming not only from people, but also from conventional sensors (e.g., fusion of satellite imagery and user generated photos for environment monitoring, extraction of users' profiles based on water consumption data and activity logs on a utility consumers' portal [63]).
- The need of validating data, for improving input accuracy and training/tuning data processing algorithms. Validation could be either automatic (e.g, via machine learning approaches), delegated to people, e.g., via crowdsourced quality control campaigns or both automatic and crowdsourced.
- The provision of mechanisms for recruiting, engaging, and retaining people, who contribute voluntarily and should be acknowledged for their participation.

Simplifying, citizen science applications could be regarded as hybrids between scientific workflows and online digital games:

- Like scientific workflows, they must amass vast collections of heterogeneous data, assess their validity, possibly improve their quality, and subject them to a processing pipeline leading to the creation of useful output knowledge.
- Like online digital games, they must trigger the motivation of users, engage them to the accomplishment of achievements, and possibly foster cooperation-competition behavioral patterns.

Such functional aspects intertwine with non-functional requirements:

- Cloud and SaaS deployment: besides the usual reasons for adopting SaaS and cloud, a virtualized back-end facilitates the collection and integration of data from heterogeneous sources under a same data model; furthermore, a multi-tenant data repository enables statistical analysis across different data sets (e.g., consumption data collected from different utility companies).
- Plug-in interfaces: although citizen science applications follow a common recipe (back-end data processing pipeline plus gamified crowd interfaces), the ingredients vary. Data processors are specific of the application at hand, as well as the interfaces for performing tasks published to the crowd.

- Multi-modality: the user interfaces benefit from a dual deployment, for both fixed (notably, PC) and mobile devices. The mobile interface supports tasks that must be executed in near real-time and exploit the user's location; the fixed (e.g., PC) interface better serves more elaborated tasks, such as textual annotation or accurate verification of content.

We propose a generic three-tier architecture for rapid development of citizen science applications, described in Section 7.1. The architecture is generic and can be applied to any citizen science problem (both to environmental and other types of monitoring, such as urban monitoring, emergency management, etc.).

Section 7.2 describes how the proposed generic architecture has been applied to a complex citizen-participated scientific workflow, called *SnowWatch*. SnowWatch is a web platform aimed at snow cover monitoring through the fusion of the algorithms for automatic web content processing described in this thesis and the crowdsourcing and citizen science approaches. The client tier consists of a public web portal and an augmented reality mobile application.

The mobile application is meant to engage the crowd in taking and uploading mountain pictures, thus, increasing the *SnowWatch* dataset of available photographs. However, since it is required to work in real-time, a part of the backend algorithms has to be implemented client-side, in the mobile application. To deal with this non-trivial task we devise a novel framework for outdoor augmented reality applications, which is described in detail in Section 7.3.

7.1 Rapid Prototyping of Citizen Science Applications

This section discusses a generic software architecture we devised for the rapid development of citizen science applications, based on three main tiers:

- A **back-end** that supports the composition of data processing workflows, by the collation of independent, loosely-coupled data acquisition and analysis modules. Differently from traditional scientific workflow systems, the pipeline engine can delegate data acquisition and processing tasks not only to automatic services, but also to a crowd of contributors.
- A **client tier**, which can hosts multiple applications, web and mobile, that implement common interfaces for publishing tasks to workers and collecting their contributions.
- A **middle tier** independent of both the data processing back-end and of the client crowdsourcing applications, which factors out the engagement policies and achievement rewarding rules enacted to secure people participation and durable commitment.

7.1.1 General Architecture and Data Model

Figure 7.1 overviews the general architecture, which consists of three tiers, back-end, middle tier and client tier, tied together by a common data model.

Figure 7.2 shows the essential elements of the conceptual data model shared by the components of the system architecture. Its purpose is the representation of user's activities in gamified applications for data processing. It draws on previous user and

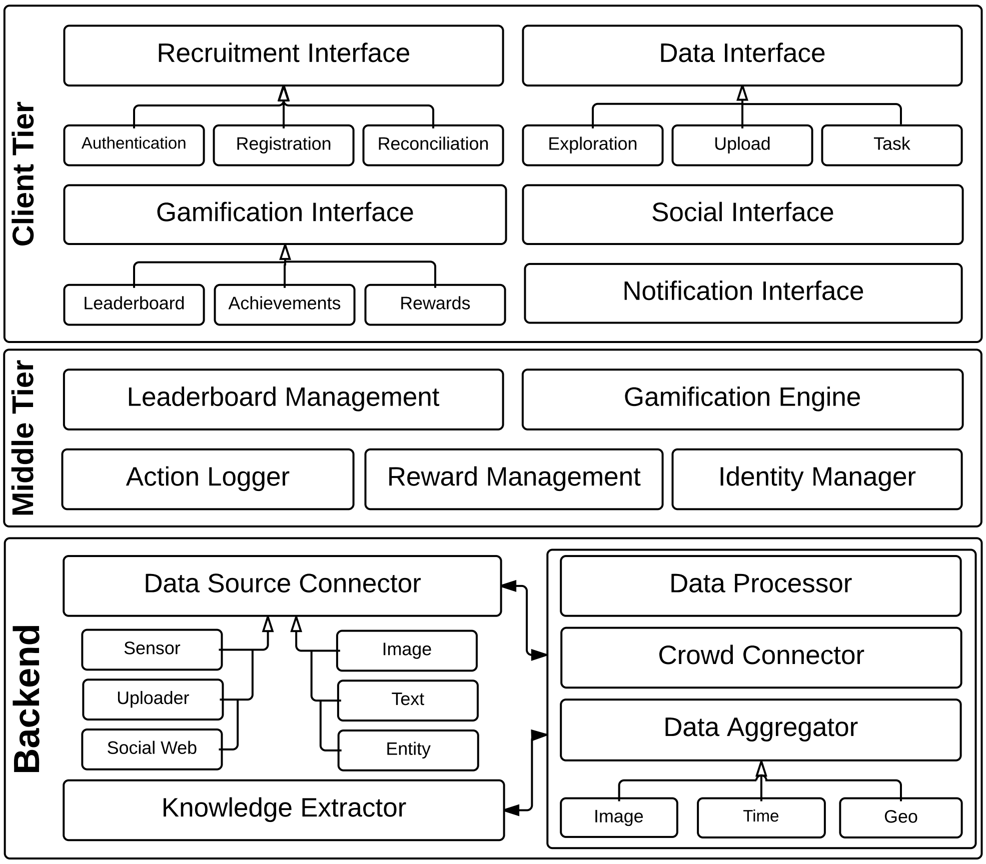


Figure 7.1: *Overview of the proposed generic architecture, components, and service interfaces.*

task models (most notably, WS Human Task [90], SWUM [24], and [13]); it describes users and their activities, the latter formalized as *tasks* associated with operational data (called *observations* in the context of citizen science applications). To these classic concepts of data- and crowd-intensive processes, it adds the notions of user's *actions*, *achievements* and *rewards*, to correlate the task model to the gamification aspects.

The data model is organized in four sub-models. The **User and Social Model** (in yellow): introduces humans into the conceptual model, by expressing the roles they play and their social interactions (e.g., friendship links, physical neighborhood, etc). The **Observation Model** (in blue): describes the task-related data (observations); they are the main input of the data processing pipelines and carry the latent knowledge that must be extracted; observations are characterized as sensor-generated or citizen-provided. The **Process Model** (in green, simplified for brevity in Figure 7.2): focuses on the tasks and their execution constraints. The **Gamification Model** (in red): expresses the engagement and rewarding mechanisms typical of gaming and of gamified applications (including scores, achievements, rewards, and leader boards).

The backbone is the user taxonomy in the **User and Social Model**. The main concept is the *Citizen*, which specializes into *Active Citizen*, and *Competitor Citizen*. The generic Citizen entity represents a user who interacts by *consuming* observations (e.g., accessing the knowledge produced by the system as a mere observer). A citizen who performs tasks, and thus can also *produce* observations, is called Active Citizen; if she is also engaged into gaming or gamified applications, she is called Competitor Citizen.

The **Gamification Model** defines the properties and associations characterizing the Competitor Citizen, with the entities *Action*, *Reward*, and *Achievement*. The Competitor is rewarded for her performed Actions, which result from the execution of Citizen Tasks; a task can be associated with zero, one or multiple actions (e.g., a task that requires the user to log-in into the system and execute a piece of work may be associated with two rewarded actions: a log-in action, rewarded with a fixed amount of points, and a work-specific action, rewarded with a variable amount of points depending on the quality of the contribution, difficulty of the work, etc). An Action is defined by a *name*, the *Area* of interest (e.g., participation, socialization, education, content creation, content assessment, etc), and the *score*. Different policies can be associated to an action, i.e., whether the action can only be rewarded once or multiple times, repeated only after a given time interval, executed only on one specific object or repeated on multiple objects, rewarded up to maximum number of points or before a due date, etc. By earning points with Actions in a given Area, a Competitor earns points, attains Achievements and spends earned points by redeeming Rewards. Achievements are a virtual recognition, such as digital badges, whereas Rewards are real goods that the users can redeem, consuming her points. Both can be specialized to describe a variety of ways to challenge users and acknowledge their contribution.

The **Observation Model** comprises the *Observation* entity, which represents the task operational data of a citizen science application: typically, a piece of text, image, video, or video stream describing a phenomenon, contextualized in a given *Location* (optional) on a give *Date* (optional). It can be automatically generated from a *Sensor* or provided by a Citizen. A Sensor is defined by a *Type* (physical, e.g., a water flow sensor, or virtual, e.g., a webcam producing mountain photos from which snow measures can be extracted); a *Frequency* which specifies the temporal rate at which the Sensor produces Observations and a *Status* (enabled, stand-by, disabled or faulted). The source of an Observation can also be a citizen, with two possible scenarios: the Observation is acquired passively by crawling social networks and content sharing sites (*Crawled Observation*), or is uploaded proactively by an Active citizen (*Uploaded Observation*). *Annotation* represents (possibly noisy) meta-data extracted from an Observation by humans or services (e.g., the position of a mountain peak).

The **Process Model** describes the hybrid service and human task pipelines by which the Observations are processed to generate knowledge. It supports a simplified version of BPEL4People and WS Human Task workflow concepts, including a notion of *Process*, consisting of *Tasks* that involve the interaction with, the processing, or the creation of Observations; a Task can be *Automatic*, i.e., executed by software components (e.g., a classification algorithm), or human (*Citizen Task*), i.e., performed by an Active Citizen. Executed tasks are recorded with their run-time meta-data (*StartTime*, *EndTime*, etc.). The Process Model connects to the Gamification Model as follows: if a Citizen Task is associated to a Gamification action, the outcome of the task becomes visible to the Gamification Engine, which determines a reward for the Competitor Citizen executing the task and adds the corresponding amount of points to her profile.

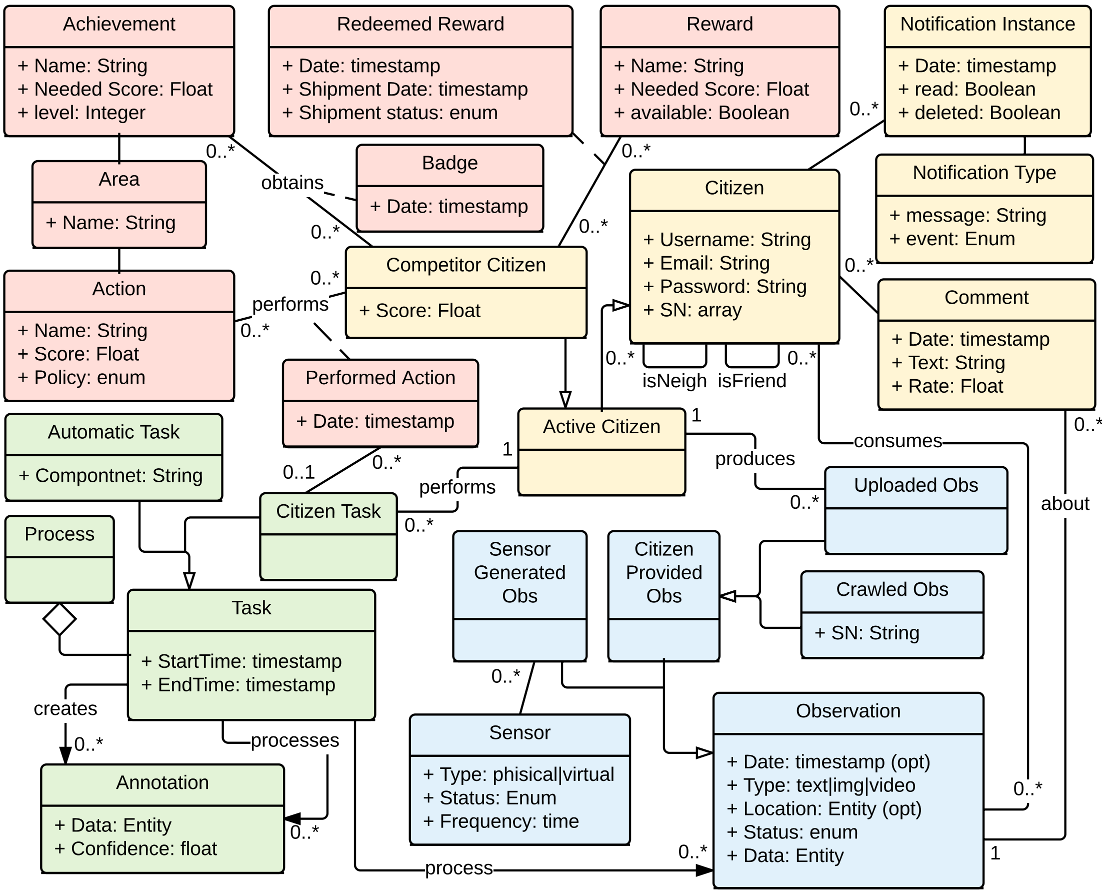


Figure 7.2: *Excerpt of the common data model for data integration, including components of the User and Social Model (yellow), Observation Model (blue), Process Model (green) and Gamification Model (red).*

7.1.2 Back-end: Data Acquisition and Processing

The back-end tier of the architecture of Figure 7.1 is responsible of the connection to the data sources and the processing of observations with a mix of automatic and human steps. It comprises the run-time support for the execution of processes following a simplified version of the BPEL4People and WS-Human Task specifications. A number of automatic task types are predefined, and are assigned to abstract processors. An abstract processor specifies an interface and an interaction protocol, which is (manually, at design-time) instantiated by concrete services that offer the computational resources for executing the task.

The *Data Source Connector* specifies the interface and protocol for integrating data acquisition services, and is further refined into specializations adapted to various classes of data sources. As an example, Figure 7.3 (left) shows the interface and the interaction protocol of the Webcam Data Source Connector, which crawls frames from webcams. Data source connectors can be specified also in the case in which the data source is a human being: in this case the interaction protocol refers to the Task GUI and Crowd Connector that implement the data acquisition step. An example of this situation is provided next, for data processors.

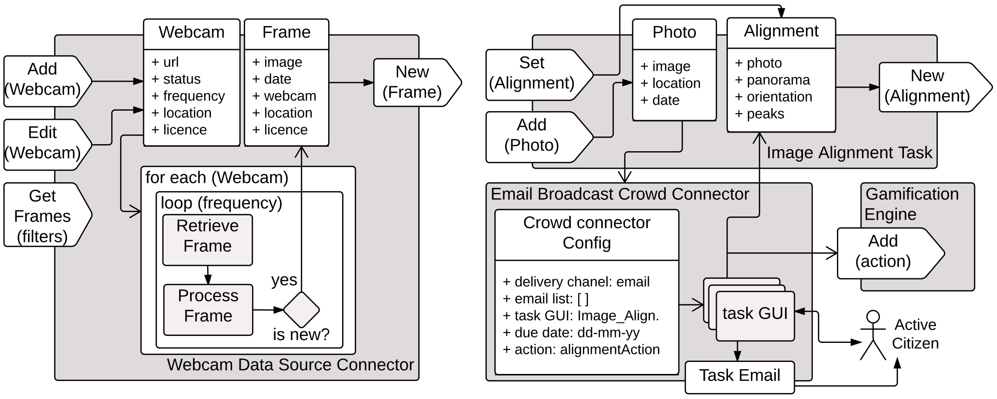


Figure 7.3: *Specifications of the Webcam Data Source Connector (left) and of the Alignment Citizen Task (right)*

The *Data Processor* interface specifies the common traits of services that execute observation processing steps, by transforming input observations into intermediate output or final knowledge. Examples of predefined processors, with default concrete implementations, include: text pre-processors (language detectors, stop-word removers, stemmers), text classifiers (topic detectors, tweet relevance classifiers) image low level and high level features extractors, image classifiers (whole image, pixel-level), and image object detectors (e.g., mountain peak detectors). Data processors can be specified also in the case in which the processor is a human being: Figure 7.3 (right) shows the case of the *Human_Image_Alignment Connector*: the interaction protocol refers to the Citizen Task GUI and to the Crowd Connector used to crowdsource the task and the Gamification Engine.

The *Crowd Connector* interface factors out the non-functional properties related to the crowd interaction from the Citizen Task. In this way, one can define a crowd interaction mode once, as a specific crowd connector, and reuse it in multiple tasks with different purposes, data, and GUIs. The non-functional properties of a crowd connector include: the delivery channel(s), the criteria for people assignment (push of task to people, task publication for people to pick), the reference to the task rendition GUI (embedded or linked), the triggering event, the due date, the output aggregation policy. Predefined crowd processors are provided, which range from the simplest case, (push email messages to a workers' list, with a link to an external task execution GUI), to full-fledged crowd campaigns executed with dedicated frameworks. Figure 7.3 (right) shows the reuse of a Crowd Connector for simple push email campaigns in the delivery of an image alignment task.

The *Data Aggregator* interface supports the instantiation of services that extract aggregate properties from data sets. Examples include the extraction of trending topics, users, and hashtags from sets of Tweets, the computation of statistics from time series or spatial data, the computation of median daily images from webcam photo series.

Finally, the *Knowledge Extractor* interface caters for the description of services that compose high-level, user-oriented information from low-level observations and intermediate data or pull together multiple outputs to deliver a high-level representation amenable for publication in a human interface. For example, provided knowledge extractors compute snow altitude indexes from processed mountain images and overlay images with peak names and positions and snow-related meta-data.

7.1.3 Middle Tier: the Gamification Engine

The middle tier hosts support services for enabling the registration of users and managing their identity (*Identity Manager*) and logging actions (*Action Logger*). Its most important component is the *Gamification Engine*, which is a rule based engine transforming actions into points. Such points determine the automatic unlocking of Achievements (e.g., the assignment of badges), establish the status of users in local or global competitions (handled by the *Leaderboard Manager*), and enable the redemption of a Reward by the user (handled by the *Reward Manager*). The gamification rules are described declaratively in the data model and executed by the Gamification Engine. A gamification rule has the form:

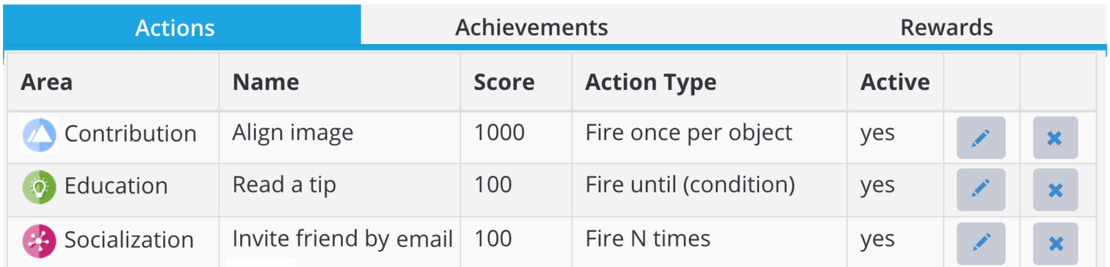
```
WHEN action A of user U occurs on object O at time T
IF Action_Mapping(A, M)
THEN Assign_Points(U, A, O, T, M.parameters)
```

The rule is triggered by the occurrence of an action (A), performed by a user (U), at time T, possibly on some known object (O). This event can be signaled to the Gamification Engine via service calls produced either by the successful termination of a Citizen Task (e.g, the processing of an observation) or by system events (e.g., the user entering in the top-ten of a leaderboard). The rule fires only if there is an active mapping defined for it, which means that there is a working association between the action and the set of parameters that determine the calculation of the points and achievements associated with it. Mappings can be defined but not enabled, e.g. for debugging purposes. If an active mapping is found, its parameters are collected (type of action, base points) and the rule is executed. Execution takes in input the relevant user, the action, the affected object, the time-stamp of execution, and the mapping parameters, assigns the proper amount of points to the user, and checks if some achievement is reached. The predefined types of actions include:

- **Fire once/N times**: the action is rewarded only at the first occurrence (or for the first N occurrences).
- **Fire once per object**: the action can be rewarded an unbound number of times, but only on distinct objects (for example, as many times as learning objects are read by the user, but only once per distinct learning object).
- **Fire before**: the action is rewarded only before a due date.
- **Fire after**: the action is rewarded only if a given time span has elapsed after its last occurrence (e.g., reward at maximum one log-in action per day).
- **Fire until**: the action is rewarded only before a given condition becomes true (for example, only the first N users that perform it are rewarded).

The calculation of the points by default is independent of the affected object and amounts of base points to grant is specified in the action properties. However, more elaborated policies can be plugged in by specifying further action types (e.g., smoothing the

amount of points over time, basing it on some attributes of the object that qualify the difficulty of the task, etc.)
An example of the admin gamification GUI, including areas, actions, achievement and rewards is shown in Figure 7.4.



Area	Name	Score	Action Type	Active		
Contribution	Align image	1000	Fire once per object	yes	✎	✖
Education	Read a tip	100	Fire until (condition)	yes	✎	✖
Socialization	Invite friend by email	100	Fire N times	yes	✎	✖

Figure 7.4: *Admin GUI for editing the Gamification Data model entities and relationship.*

7.1.4 Client Tier

The client tier comprises a set of predefined GUI utility components for building gamified crowdsourcing interfaces. The *Recruitment* interface supports registering and authenticating users into the system; it also manages identity reconciliation, to support users registered to multiple connected applications. The *Notification* interfaces supports bidirectional communication with citizens. The *Social* interface enables the posting of achievements on social networks and the creation of peer-to-peer invitations to tasks. The *Gamification* interface supports the publication of leaderboards and achievements, and the redemption of rewards. The *Data* interface supports the publication of observations and the upload of new content.

7.2 SnowWatch Implementation

The SnowWatch project [51] targets the need of low cost analysis of environmental and ecological phenomena, made extremely pressing by climate change and shrinking public investments in monitoring infrastructures. It tackles the problem of mountain environment monitoring with a Citizen Science application for the collection of public images depicting Alpine mountains and the extraction of snow indexes usable in environmental models. The SnowWatch project is a fusion between the content processing algorithms presented in this thesis (photograph acquisition, Section 3; photograph geo-registration, Section 4; photograph snow cover analysis, Section 5) and the citizen science applications described in this section.
Crowdsourcing is employed for four tasks: validating the classification of images that contain visible mountain profiles (Section 3.1.2); validating the geo-registration w.r.t. the terrain computed automatically (Section 4.1); adding/adjusting the GPS geotag of a photograph; collecting images on demand, e.g., portraying mountain for which there is not enough user-generated conent and no webcams are available. Figure 7.5 shows the pipeline of service and crowd tasks that compose the SnowWatch process.
The pipeline is implemented by instantiating the general architecture of Figure 7.1 reusing standard components and adding, where necessary, domain-specific services. The data model is specialized too, e.g., the generic Observation entity is sub-classed to

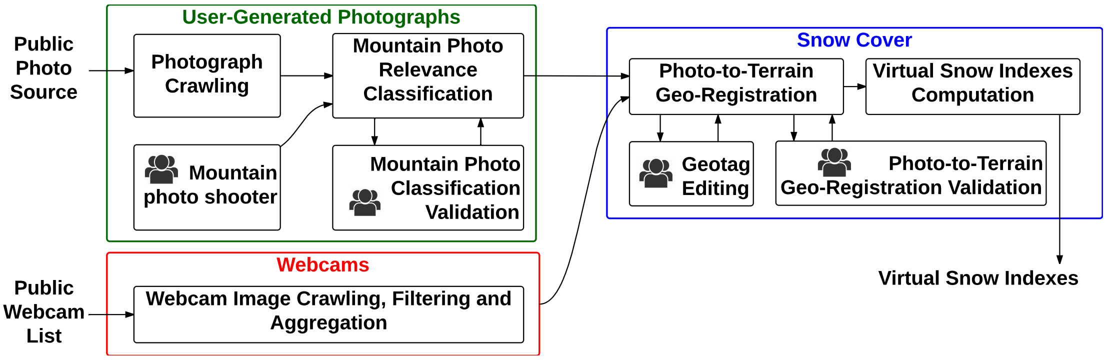

Public Photo Source
User-Generated Photographs
Photograph Crawling
Mountain Photo Relevance Classification
Mountain photo shooter
Mountain Photo Classification Validation
Snow Cover
Photo-to-Terrain Geo-Registration
Virtual Snow Indexes Computation
Geotag Editing
Photo-to-Terrain Geo-Registration Validation
Virtual Snow Indexes
Public Webcam List
Webcams
Webcam Image Crawling, Filtering and Aggregation

Figure 7.5: *Pipeline of service and human tasks in SnowWatch*

accommodate properties needed to represent the technical features of photos (geo-tag, sensor model, optical parameters, field of view, etc). Figure 7.6 shows how the general architecture proposed in Figure 7.1 has been applied to the SnowWatch architecture.

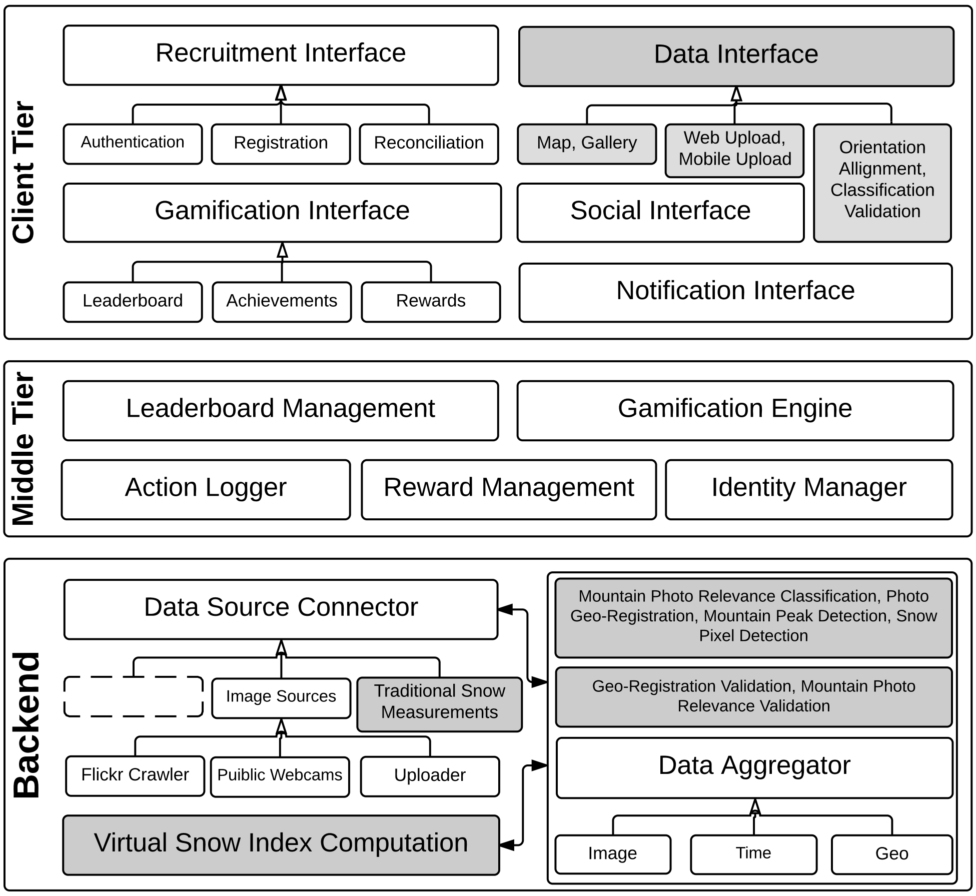


Figure 7.6: *Overview of the SnowWatch architecture, a specialization of the one proposed in Figure 7.1. Specialized components are highlighted in gray.*

The back-end data acquisition and processing interfaces have been instantiated as follows. Two *Image source connectors* and one *Data Aggregator* have been configured, for acquiring images from photgraph sharing websites, webcams, and users, and improving their quality. The *Flickr Crawler* described in Section 3.1 is used in *Photo Crawling* automatic task of Figure 7.5; it specializes the generic, keyword-search, image source connector with three filtering criteria: the photo must be geo-tagged; located within a rectangular region provided in input; the altitude of the shooting location must be higher than a minimum threshold. The *Mountain webcam crawler* described in Section 3.2, used in the *Webcam Image Crawling* and *Weather condition filtering* (Section 3.2.2) automatic tasks, specializes the generic webcam source connector with a filtering step. Since cloudy meteorological conditions are very common at high altitudes, the connector discards images with bad weather conditions. The *Image aggregator*, used in the *Daily image aggregation* automatic task (Section 3.2.3), specializes the Data Aggregator interface to collapse a set of input mountain images, taken in good weather conditions, into one daily median image.

The *Data Processor* interface is the one most heavily specialized, to incorporate the domain-specific algorithms for mountain image analysis. The realized specializations include the *Mountain photo relevance classification* (Section 3.1.2) and the *Photo-to-terrain geo-registration* (Chapter 4).

The *Crowd Connector* has been instantiated to support the human tasks of Figure 7.5, as follows: a task GUI has been added in the client tier, enabling the execution of the task. And a crowd connector has been allocated in the back-end, to support the recruitment of the contributors.

Finally, the *Knowledge Extractor* has been instantiated with a *Virtual snow indexes computation* service (Section 5.3.2).

The Middle Tier (Gamification Engine) has been reused without modification. Its administrative interface (shown Figure 7.1) has been used to configure the Gamification Data Model, by creating: *i)* actions that can be performed by a user (e.g. upload a photo, share a photo with friends, comment a photo, validate a photo, manually align a photo, etc); *ii)* achievements and actions required to obtain them; *iii)* rewards that can be provided to users for their achievements.

The Client Tier of the application has been customized by adding the GUIs needed for supporting the execution of human tasks and an exploratory web portal interface for the general public:

- *Exploratory web portal*: it customizes the Exploration Data Interface (Figure 7.7 top) to support browsing the geolocated image collection, in two ways: with a map view, placing the images on a map in the positions they were shot; and with a gallery view that publishes all images into a scrollable grid. The web portal supports also the crowd tasks: *Photo upload*; *Geotag edit*; *Mountain photo classification validation*, the user can label as negative (does not contain mountains) a photo that was erroneously classified as positive; *Photo-to-terrain geo-registration validation*, the user can adjust the automatically computed alignment of the photo w.r.t. the rendered terrain view (Figure 7.7 bottom).

- *Mobile application*: the user can upload own photographs using a mobile application supporting the human task whereby the user can take photos of mountains with the peak names automatically overlaid onto the image, described in details in Section 7.3.

Overall, the development of the SnowWatch application required 10 % of the total ef-

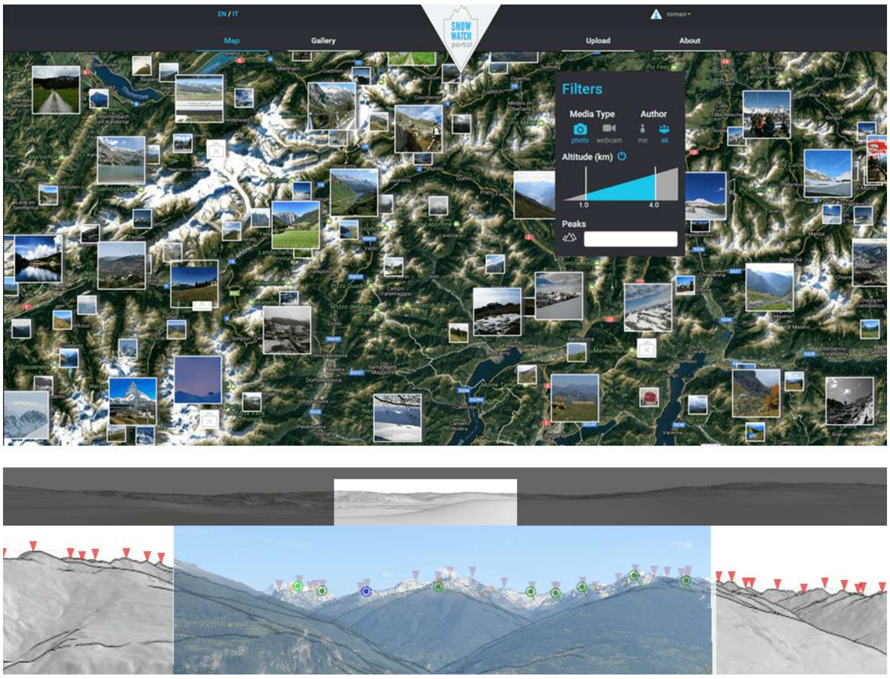


Figure 7.7: *GUIs of the SnowWatch human tasks: exploratory web portal home page (top) and manual photo-to-terrain geo-registration (bottom).*

fort, with 90 % of the effort spent in the research of the computer vision and machine learning algorithms in the back-end. The effort has been spent in the integration of the services in the pipeline and in the realization of the human task and data exploration GUIs. Negligible effort has been requested by the integration of the crowd support: alternative crowdsourcing approaches have been used for different purposes: from small captive crowds reached by email in the early validation phase to the deployment of a Social Network crowd connector for posting peak-labeled images in one's wall and inviting friends to test the application. The gamification aspects have been added a posteriori; the proposed generic architecture has sensibly speed-up the integration of such functionality: once the task GUIs have been completed and the gamification rules configured, the only technical effort has been the addition of one service call from the task GUI to the Gamification Engine for action notification.

In retrospective, the main contribution of the proposed rapid prototyping architecture resulted in the evolutive maintenance phase. Substantial change requests in a variety of areas, such as data analysis services, workflow of the data processing tasks, human task interface and device, recruitment and engagement methods have been tackled effectively thanks to an approach based on a mix of a common centralized data model, standard service interfaces, reusable components, and declarative business rules, factored out of the code. These benefits are especially welcome in applications directed to volunteer crowds, which must exhibit a persuasive nature, attract people, engage them in (sometimes not so exciting) activities, retain their attention over long periods of time, and even change their habits. This entails the need of adapting the crowd engagement methods, task interfaces, and gamification rule continuously, to fit them to the current level of people's engagement.

7.3 Augmented Reality Mobile Application

In this section we describe the mobile application, which is **a part of the SnowWatch client tier**. The application supports the crowdsourcing task of adding new photographs to the SnowWatch platform whereby the user can take photographs of mountains and upload them automatically. Differently from the other crowdsourcing tasks, the mobile application borrows key concepts from Games With A Purpose (GWAP) paradigm: the idea is to engage the user into using the application due to a personal entertaining instead of a volunteer participation to a citizen science campaign. Specifically, the entertaining aspect of the application is the capability of identifying mountains that are framed with the mobile device in real-time.

For this purpose, we devised a novel framework for the development of outdoor mobile Augmented Reality (AR) applications. Section 7.3.1 describes the framework, which is generic and can be applied to any outdoor AR scenario that requires precise object identification through computer vision algorithms. Section 7.3.2, instead, describes the experience of implementing the mountain identification application through the proposed framework and discusses engagement milestones achieved by the application.

7.3.1 Mobile Augmented Reality Applications Framework

Outdoor augmented reality applications exploit the position and orientation sensors of mobile devices in order to estimate the location of the user and her field of view so as to overlay such view with information pertinent to the user's inferred interest. These solutions are finding a promising application in the tourism sector, where they replace traditional map-based interfaces with a more sophisticated user experience whereby the user automatically receives information based on what he is looking at, without the need of manual search. Examples of such AR apps include, e.g, Metro AR and Lonely Planet's Compass Guides[1]. The main challenge of such applications is providing an accurate estimation of the user's current interest and activity, adapted in real-time to the changing view. Commercial applications, which operate mostly in the tourism field, simplify the problem by estimating the user's interest based only on the information provided by the device position and orientation sensors, irrespective of the content actually in view. Examples are sky maps, which show the names of constellations, planets and stars based on the GPS position and compass signal. An obvious limit of these approaches is that they may provide information that does not match well what the user is seeing, due to errors in the position and orientation estimation or to the presence of objects partially occluding the view. These limitations prevent the possibility for the AR application to create *augmented content* usable for monitoring purposes. If the overlay of the meta-data onto the view is imprecise, it is not possible for the user to save a copy of the augmented view, e.g., in the form of an image with captions associated to the objects. Such augmented content could be useful for several purposes: archiving the augmented outdoor experience, indexing visual content for supporting search and retrieval of the annotated visual objects, and even for the extraction of semantic information from the augmented content.

AR is a well established research topic within the Human Computer Interaction field, which has recently attracted new attention due to the announcement by major hardware

[1] http://www.lonelyplanet.com/guides

vendors of low-cost, mass-market AR devices. In particular, the recent trend of mobile devices as AR platforms benefits from the improved standardization (most AR software can now be used without ad hoc hardware), increased computational power and sensor precision [93]. The survey in [9] overviews the history of research and development in AR, introduces the definitions at the base of the discipline, and positions it within the broader landscape of other technologies. The authors also propose design guidelines and examples of successful AR applications and give an outlook on future research directions. An important branch of the discipline is the outdoor AR. Several works address the problem, usually to identify [34] and track [148] points of interest in urban scenarios. Although standard solutions for mobile AR already exist (e.g. Wikitude[2]), they rely only on compass sensors or the a priori known appearance of the objects. We present a novel framework for the fusion of the two techniques: refining the compass-based AR performance without knowing a priori the appearance of the objects.
The problem addressed in this framework is the design of mobile AR applications for the enrichment of outdoor natural objects. Restricting the focus to devices that support a bi-dimensional view, a generic architecture must be realized that receives as a first input a representation of the reality - in which the user is embedded - captured by the device sensors; such representation typically comprises a sequence of camera frames captured at a fixed rate, and the position and orientation of the device, captured by the GPS and orientation sensors; the second input is the information about the possible objects present in a region of interest. The output is the on-screen position of relevant objects and the association of relevant meta-data to such objects, computed at the same frequency of the input capture. Besides the near real-time execution time, the system must also cope with the following requirements:

- *Uncontrolled viewing conditions*: the objects to be identified have no fixed, known a priori, appearance, because the viewing conditions can drastically change due to weather, illumination, occlusions, etc.

- *Uncertain positioning*: position and orientation sensor errors make the location estimation potentially noisy; thus the identification of the relevant objects from these signals alone cannot be assumed to be fully reliable and must be corrected with information from the camera view.

- *Bi-dimensional reduction*: although the objects' position in the real world is estimated in the 3D space, the on-screen rendition requires a projection onto the 2D surface of the camera view, based on a model of the camera.

- *Uncertain internet connection*: especially for rural and mountain regions.

Figure 7.8 shows a representation, through an UML component diagram, of the reference architecture of a mobile outdoor AR application. The key idea is to enable the near real-time reality augmentation process thanks to a proper partition of functionality and a mix of synchronous and asynchronous communications among the modules. The architecture consists of four sub-systems: the Sensor Manager, the Data Manager, the Position Alignment Manager and the Bi-dimensional Graphical User Interface, which draws objects and their metadata in provided on-screen coordinates.

[2] http://www.wikitude.com/app/

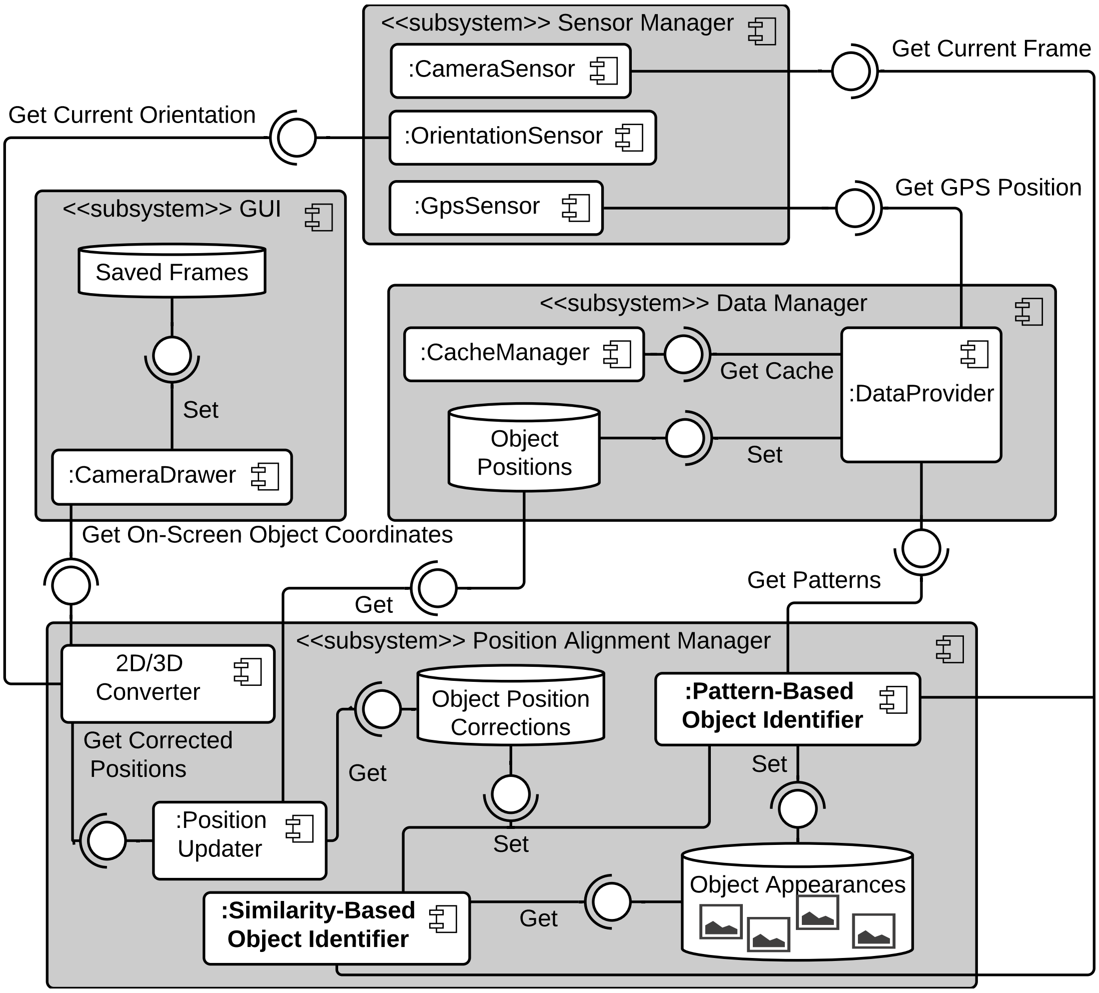


Figure 7.8: *The proposed architecture of a mobile outdoor AR application.*

Sensor Manager

The *Sensor Manager* coordinates data acquisition from the device sensors. It typically comprises one module per each signal processed by the application; the typical configuration comprises the GPS Sensor Manager, the Orientation Sensor Manager and the Camera Sensor Manager. The modules work asynchronously and provide input to the Position Alignment Manager and Data Manager, which subscribe to their interface and are notified when a new signal arrives from a sensor.

Data Manager

The *Data Manager* is responsible for providing to the other sub-systems the initial positions of the objects in view and the meta-data for enriching them. It receives as input the specification of an area of interest (typically, inferred from the user's position, which defines the region the user may be looking at, or may be moving within), and interacts with an external repository containing a virtual representation of the world (e.g, a sky map or a DEM). It produces as output *Object Positions*, which specify the (initially approximate) 3D coordinates of the candidate objects to display. Within the *Data*

Manager, a *Data Provider* component queries one or more external geo-referenced data sources, with the current user's location, and extracts the coordinates of the objects that are likely to lie within the view of the user. For example, in a sky observation app, it queries the sky map for the celestial coordinates, plus meta-data such as type, name, distance, etc., of the potentially visible objects. The *Cache Manager* implements data pre-fetching and synchronization policies, based on information about current cache content, network availability, and cost of data transfer. Since data about the objects can be large the Cache Manager realizes a trade-off between on-demand transfer from external data sources and caching in the local storage of the device. Furthermore, it enables disconnected usage, as needed in the outdoor scenario, in which internet connection may not be always granted.

Position Alignment Manager

The Data Manager provides a *fast* computation of the initial Object Positions, to enable the immediate update of the GUI. But its output may be noisy, because the estimated user's position, the camera orientation and the virtual world representation may all contain errors. It is well-know that the GPS and orientation signal of mobile devices may be inaccurate; on the other hand, also the virtual world representation, e.g., a Digital Elevation Model (DEM) describing the earth surface, may be affected by errors, e.g., due to low resolution. Therefore, the Position Alignment Manager comprises components for updating the positions of the objects, adapting them to the actual content of the camera view, and projecting them to the device's view. It takes in input the initial object positions provided by the Data Manager and produces in output the *corrected* on screen object coordinates. To support the trade-off between accuracy and speed, the (demanding) computations required for improving accuracy are delegated to separate modules, which provide asynchronous corrections to the initial candidate positions, by applying content-based object detection techniques. These modules feed the *Object Position Corrections* store (see Figure 7.8) with the adjustments computed asynchronously, which the *Position Updater* and *3D/2D Converter* components exploit to correct the on screen coordinates used by the GUI. Examples of components for the content-based refinement of object positions are *Pattern-Based* and *Similarity-Based* Object Identifiers.

A *Pattern-Based Object Identifier* performs a frame-based match. It uses the virtual world representation as a pattern to search within the real world image. It takes in input the virtual representation of the world (e.g., the synthetic rendition of a constellation or of a piece of mountain skyline) and computes a ranked list of approximate matches between the virtual image and the real one, with respect to some similarity function. As a collateral output, the Pattern-Based Object Identifier can also extract from the real world image the regions that correspond to the identified objects, according to the best match. Such artifacts, cached in the *Object Appearance Store* of Figure 7.8, denote the visual appearance of the objects of interest in the current view and can be used for accelerating the correction of objects' positions when the view changes.

A *Similarity-Based Object Identifier* performs object-based similarity search; it takes in input the object appearance artifacts and searches them in the frame, using computer vision techniques.

Finally, the *2D/3D Converter* projects 3D positions onto the bi-dimensional screen

space. It takes in input the device position, orientation, and Field Of View (FOV), applies a prospective projection, determines the on-screen coordinates of the candidate objects and discards those out-of-view, e.g, due to micro-movements of the device. For example, it projects the celestial coordinates of the relevant sky objects into on-screen coordinates. The on-screen coordinates are used by the GUI for rendering the augmented reality view.
The asynchronous communication between the components that compute position corrections and those that project positions and render the virtual reality view aims at enabling a best effort, near real-time adjustment of the view. The prospective projection is a constant-time procedure, so that the total response time of the Position Updater and of the 3D/2D Converter is linear w.r.t. to the number of candidate objects. Since this number is reasonably bound, the resulting time complexity is constant, which allows the mobile device to call the Position Updater and the 3D/2D Converter *synchronously* at every frame arrival and redraw the view in near real-time based on the best available approximation of the object positions.

Capture and Replay Testing Framework

Testing an outdoor AR application is a complex task that requires evaluating simultaneously the precision of object positioning and the response time, two competing objectives, in a realistic setting that considers the sensor inputs (not available in the lab). The assessment criteria must also take into account usage conditions: if the user keeps the device steady, low error is the prominent goal, while higher execution time due to re-positioning after micro-movements is less relevant; conversely, if the device is subject to movement (e.g, during walking), fast execution can be more important than object positioning precision. Therefore, testing should be supported by an auxiliary architecture that helps achieve the following objectives:

- Perform lab testing in conditions equivalent to real outdoor usage.
- Contrast different designs in the same operating conditions and assess the same designs under different operating conditions.
- Use the performance metrics best suited to a specific application and operating condition.

To support such requirements, we have extended the architecture of Figure 7.8 with a testing framework based on a *Capture & Replay* approach:

- A *Capture application*: it is a mobile application that can be used to record an outdoor usage session, complete with all sensor data (camera, GPS and orientation) and user's activity (start, stop, video record, snapshot, etc.).
- An *Annotation application*: it is an application that allows one to annotate the frames of a usage session with the position of the visible objects, so to create a gold standard for evaluating the accuracy of object positioning.
- A *Replay test driver*: it is an application that can attach to the Position Alignment Manager sub-system of the architecture of Figure 7.8 and measure its performance based on a plug-in metrics.

The Capture application collects execution traces. A trace consists of a sequence of entries that record all the events occurred during a usage session, including: information about the device manufacturer and model; the set of frame images taken at frequency F, with their acquisition timestamp; and the sequence of time-stamped sensor readings, i.e., the values of the position and orientation sensors acquired at the maximum frequency supported by the device. The above mentioned information, logged by default, is normally sufficient to reproduce the user activity for a typical outdoor AR application; however, the Capture application can be extended to support additional logging, if needed by a specific application. Note that the described Capture & Replay approach allows lab tests to assess the Position Alignment Manager in the same operating conditions that occur in an outdoor session, because it exploits the same frame acquisition rate and sensor sampling frequency experimented in the real time use.

7.3.2 Mountain Identification Augmented Reality Application

The SnowWatch mobile AR application specializes the architecture described in Figure 7.8. In this section we describe the application-specific concepts and component refinements introduced for the mountain identification context.

The *objects* to be identified are mountain peaks and the *object positions* are 3D global system coordinates laying on a unit sphere centered in the device location.

An application-specific *Cache Manager* has been implemented, responsible for pre-fetching and caching the Digital Elevation Model fragments corresponding to the geographical region the user is visiting. Pre-fetching is enabled when the WiFi connection of the device is on and cache data are used by the *Data Provider* component to compute the Object Positions during outdoor usage. When the user moves out of the region for which data are in the cache, a cache miss triggers the download of a new fragment, which, in case of cache full, replaces the fragment relative to the region visited earliest. The user can also manually select regions (defined by country borders) to be permanently downloaded offline.

The *Similarity-Based Object Identifier* component is implemented with a state-of-the-art cross-correlation patch recognition technique [15], which has been ported to the mobile execution environment.

The component where the most relevant adaptations have been introduced is the Pattern-Based Object Identifier, which uses the set of algorithms that implement the pattern matching between the skyline extracted from the DEM and the skyline visible in the camera view, and computes *Object Position Corrections* based on the outcome of such procedure. The algorithms are described in detail in Section 4.2.

During the application development the Capture and Reply Testing Framework (Section 7.3.1) has been widely used to simulate registered outdoor scenarios. This facilitated both the improvement of the mountain identification precision and the GUI development (since the developer was able to test the new GUI in a challenging mountain scenario). Furthermore, the framework automated regression testing of the mobile application, by computing quality metrics to trace regression errors. The regression testing was used to ensure application quality prior to every version release.

Engagement Experiments

In order to test the engagement potential of the mobile crowdsourcing, we developed the aforementioned application as a fully market-ready Android product. With a help of professional graphic designers and usability experts we finalized the app with several functionalities, such as: settings that allow to customize the mountain information displayed on the screen, photo shooting and social sharing buttons. We ran a beta testing program with 100 users for one month and fixed numerous stability and compatibility issues. Figure 7.9 shows several screenshots of the application.

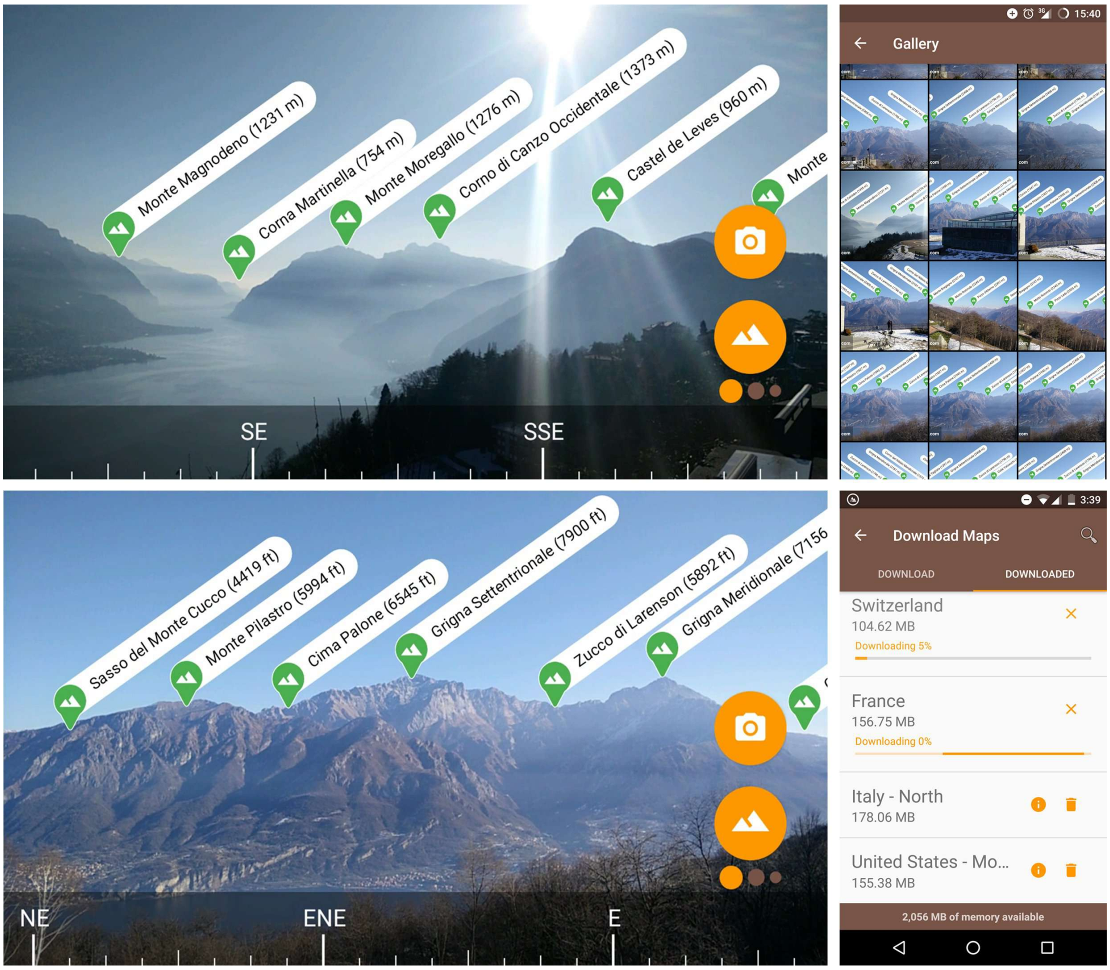


Figure 7.9: *Several screenshots of the mobile AR application.*

The experiment aimed at the sole evaluation of the engagement that can be reached with the application, thus, in order to avoid dealing with privacy policies and issues, the functionality of automatic photograph uploading to the SnowWatch website was disabled in the public version of the application.

We deployed the application publicly on Google Play Store[3], listed as a free app. The only advertisement consisted in ~ 10 posts on Facebook groups of mountain lovers in Italy. No professional or paid marketing campaigns were performed.

[3] http://peaklens.com

At the end of the monitored period of 7 months (from February 1, 2017 to August 24, 2017) the following engagement results were achieved:

- 87 k total downloads of the application and 57 k installs on active devices. Figure 7.10 shows the trend of the downloads and active devices in time.
- Average 2200 active daily users in the last month (average 2700 active daily users during the weekends). The active users are all the distinct devices that communicated with the application Web server during the day (this is a pessimistic estimation since the application is also able to work offline). Figure 7.11 shows the day-by-day active daily users.
- Top 25 position of the Google Play Store applications within the *Travel & Local* category in Italy, Switzerland, Austria and Slovenia.
- More than 50 spontaneous blog and forum posts reviewing and promoting the application.

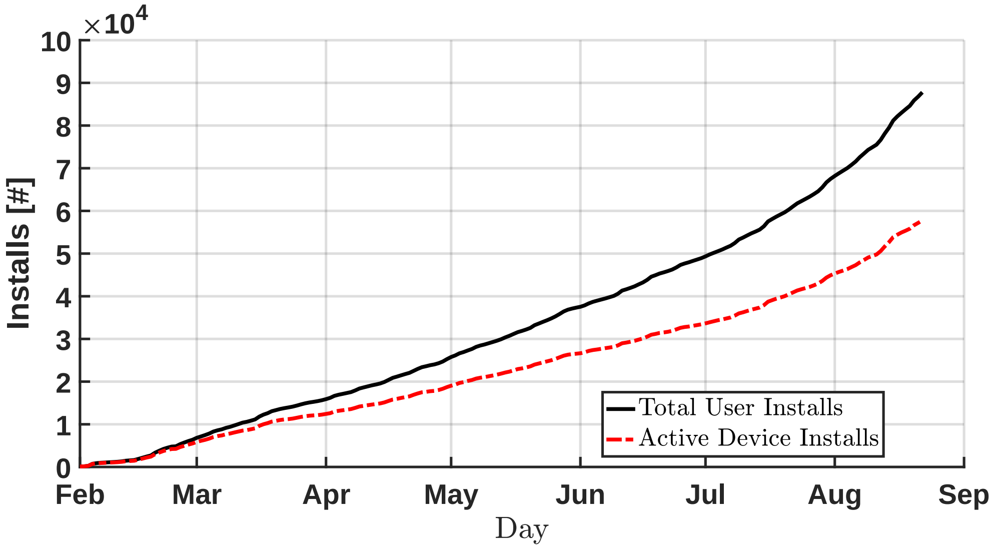


Figure 7.10: *Day-by-day total user installs and active device installs of the mountain identification application.*

The geographical distribution of the application is concentrated in Italy (54 %), since we announced it only on Italian social pages. However, it naturally expanded to the whole Alpine area, including Switzerland (12 %), Germany (7 %), France (6 %) and Austria (5 %). The United Stated market has also been involved with 5 % of the downloads.

Conclusions

In this section we described the experiment of deploying on the market the entertainment AR application that motivate users to take mountain photographs. The experiment showed that such application (that uses the algorithms developed for the automatic processing pipelines) has a great public engagement power and has the potential to contribute with massive amount of user-generated photographs. Next experiments in this

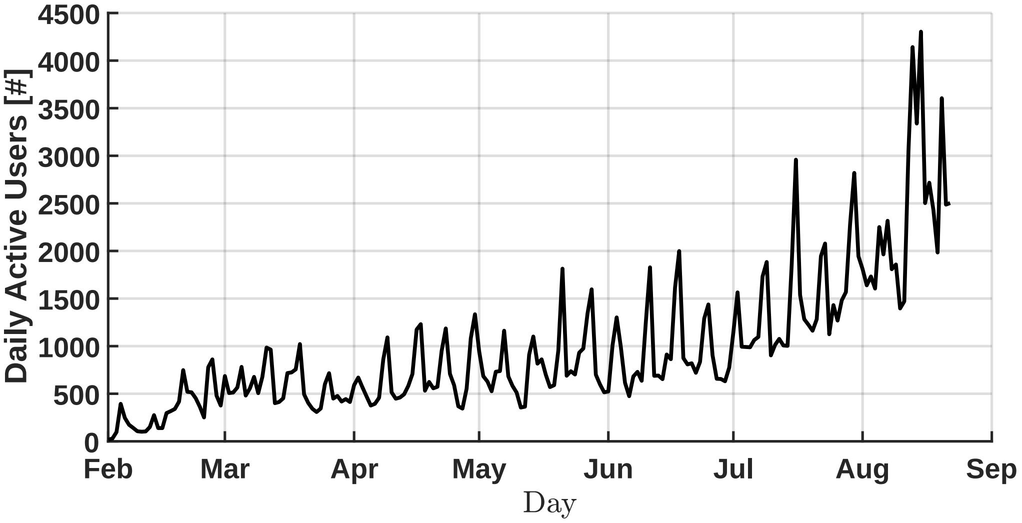


Figure 7.11: *Day-by-day daily active users of the mountain identification application.*

direction will include the collection of the photographs from the users and their validation.

CHAPTER 8

Conclusions and Future Work

In this thesis we explored the feasibility of using multimedia content publicly available on the web to enhance environmental monitoring. We addressed the problem through the use case of snow cover estimation using public mountain images.

We described an acquisition pipeline that automatically retrieves new images containing mountain slopes from photo-sharing platforms and public webcams. We discovered that the amount of user-generated photographs depicting relevant mountain slopes is insufficient to get spatio- or temporally-consistent snow cover trends. Webcam streams, on the other hand, proved to guarantee a good spatial and excellent temporal frequency of the visual observations.

We also presented algorithms for processing the acquired photographs: $i)$ mountain image geo-registration that, given a geolocated photograph as input, infer photograph geographical properties; $ii)$ virtual snow cover computation that, given a photograph identifies which areas are covered by snow and produces virtual snow indexes.

We tested the environmental usefulness of these indexes by feeding them into a real environmental resource-management model developed in collaboration with environmental researchers. We proved that, in the described use case, the virtual snow indexes obtained from a single public webcam were able to replace the original authoritative snow measurements (provided by the Italian Environmental Protection Agency, Agenzia Regionale per la Protezione dell'Ambiente) without a performance drop and improve the performance if complemented to the authoritative snow measurements. At the best of our knowledge, this is the first successful effort in moving beyond proving the data correctness and **objectively prove the practical usefulness** of the public visual web content in a real world environmental monitoring use case. We hope that this work will encourage more computer science scholars to explore the environmental potential of the public web content and stimulate more environmental researchers to experiment with web-originating inputs in their models.

The environmental impact of the obtained virtual data was, however, estimated through *simulations*. We would like to foster a closer collaboration with environmental researches to assess the performance of different control policies with real tests using live models.

Given the insufficient volumes of user-generated photographs, only webcam imagery was used for the environmental experimental setting. However, all the described processing algorithms are compatible both with webcam images and user-generated photographs. The volume of the user-generated content continues to grow and we acquire longer time series of mountain photographs. Our future work will include environmental tests at a larger scale, using hundreds or thousands of public webcams and user-generated photos. Our long-term challenges include the derivation of the statistical snow cover models using predictive models that rely on other environmental variables. Furthermore, the proposed approaches for content acquisition, relevance classification and pixel-level estimation of the desired phenomena are generic and can be applied to other environmental problems. We would like to port the proposed architecture to such problems, including sediment monitoring in river beds and vegetation monitoring in mountain regions.

In this thesis we also proposed a preliminary study on how the crowdsourcing can benefit the automatic web data acquisition and analysis pipelines. We described a web portal that allows users to contribute with own content and correct the mistakes made by automatic processing. Furthermore, we explored how the users can be engaged into providing photographs through an entertaining experience: we developed a mobile augmented reality application that is able to perform a real-time precise mountain identification. The deployment of the application on the mobile market resulted in 16 k users in the first 2 months, confirming its engagement potential. We plan to carry on with this research track and perform a longitudinal study on the effectiveness of the alternative recruitment and gamification policies, and investigate what works and what does not in the design of citizen science applications that seek the voluntary contribution of people.

Bibliography

[1] D. Anghileri, N. Voisin, A. Castelletti, F. Pianosi, B. Nijssen, and D.P. Lettenmaier. Value of long-term streamflow forecasts to reservoir operations for water supply in snow-dominated river catchments. *Water Resources Research*, 52, 2016.

[2] Mikhail J Atallah. A linear time algorithm for the hausdorff distance between convex polygons. *Information processing letters*, 17(4):207–209, 1983.

[3] Georges Baatz, Olivier Saurer, Kevin Köser, and Marc Pollefeys. Large scale visual geo-localization of images in mountainous terrain. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part II*, pages 517–530. Springer-Verlag, 2012.

[4] Lionel Baboud, Martin Čadík, Elmar Eisemann, and Hans-Peter Seidel. Automatic photo-to-terrain alignment for the annotation of mountain pictures. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 41–48. IEEE, 2011.

[5] Lars Backstrom, Eric Sun, and Cameron Marlow. Find me if you can: improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th international conference on World wide web*, pages 61–70. ACM, 2010.

[6] Tim P Barnett, Jennifer C Adam, and Dennis P Lettenmaier. Potential impacts of a warming climate on water availability in snow-dominated regions. *Nature*, 438(7066):303–309, 2005.

[7] Carlo Bernaschina, Ilio Catallo, Eleonora Ciceri, Roman Fedorov, and Piero Fraternali. Towards an unbiased approach for the evaluation of social data geolocation. In *9th Workshop on Geographic Information Retrieval*. ACM, 2015.

[8] Mark Bilandzic, Michael Banholzer, Deyan Peev, Vesko Georgiev, Florence Balagtas-Fernandez, and Alexander De Luca. Laermometer: a mobile noise mapping application. In *Proceedings of the 5th Nordic Conference on Human-computer interaction: building bridges*, pages 415–418. ACM, 2008.

[9] Mark Billinghurst, Adrian J. Clark, and Gun A. Lee. A survey of augmented reality. *Foundations and Trends in Human-Computer Interaction*, 8(2-3):73–272, 2015.

[10] Günter Blöschl. Scaling issues in snow hydrology. *Hydrological processes*, 13(1415):2149–2175, 1999.

[11] Chris A Boulton, Humphrey Shotton, and Hywel TP Williams. Using social media to detect and locate wildfires. In *Proceedings of the 1st international workshop on Social Web for Environmental and Ecological Monitoring*. AAAI Publications, 2016.

[12] Alessandro Bozzon, Piero Fraternali, Luca Galli, and Roula Karam. Modeling crowdsourcing scenarios in socially-enabled human computation applications. *Journal on Data Semantics*, 3(3):169–188, 2014.

[13] Alessandro Bozzon, Piero Fraternali, Luca Galli, and Roula Karam. Modeling crowdsourcing scenarios in socially-enabled human computation applications. *Journal on Data Semantics*, 3(3):169–188, 2014.

[14] Eliza S Bradley and Keith C Clarke. Outdoor webcams as geospatial sensor networks: Challenges, issues and opportunities. *Cartography and Geographic Information Science*, 38(1):3–19, 2011.

[15] Kai Briechle and Uwe D Hanebeck. Template matching using fast normalized cross correlation. In *Aerospace/Defense Sensing, Simulation, and Controls*, pages 95–102. International Society for Optics and Photonics, 2001.

[16] Ross D Brown and Robert O Braaten. Spatial and temporal variability of canadian monthly snow depths, 1946–1995. *Atmosphere-Ocean*, 36(1):37–54, 1998.

[17] Geisa Bugs, Carlos Granell, Oscar Fonts, Joaquín Huerta, and Marco Painho. An assessment of public participation gis and web 2.0 technologies in urban planning practice in canela, brazil. *Cities*, 27(3):172–181, 2010.

[18] L. Busoniu, D. Ernst, B. De Schutter, and R. Babuska. Cross–Entropy Optimization of Control Policies With Adaptive Basis Functions. *IEEE Transactions on systems, man and cybernetics–Part B: cybernetics*, 41(1):196–209, 2011.

[19] Daniel E Canfield Jr, Claude D Brown, Roger W Bachmann, and Mark V Hoyer. Volunteer lake monitoring: testing the reliability of data collected by the florida lakewatch program. *Lake and Reservoir Management*, 18(1):1–9, 2002.

[20] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[21] A. Castelletti, S. Galelli, M. Restelli, and R. Soncini-Sessa. Tree-based reinforcement learning for optimal water reservoir operation. *Water Resources Research*, 46(W09507), 2010.

[22] A. Castelletti, S. Galelli, M. Restelli, and R. Soncini-Sessa. Tree-based feature selection for dimensionality reduction of large-scale control systems. In *Proceedings of the IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, Paris, France, 11–15 April 2011.

[23] Andrea Castelletti, Roman Fedorov, Piero Fraternali, and Matteo Giuliani. Multimedia on the mountaintop: Using public snow images to improve water systems operation. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 948–957. ACM, 2016.

[24] Federica Cena, Antonina Dattolo, Ernesto William De Luca, Pasquale Lops, Till Plumbaum, and Julita Vassileva. Semantic adaptive social web. In *Advances in User Modeling*, pages 176–180. Springer, 2012.

[25] Andrea Ceron, Luigi Curini, Stefano M Iacus, and Giuseppe Porro. Every tweet counts? how sentiment analysis of social media can improve our knowledge of citizens' political preferences with an application to italy and france. *New Media & Society*, 16(2):340–358, 2014.

[26] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 1–14. ACM, 2007.

[27] Hau-wen Chang, Dongwon Lee, Mohammed Eltaher, and Jeongkyu Lee. @ phillies tweeting from philly? predicting twitter user locations with spatial word usage. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, pages 111–118. IEEE Computer Society, 2012.

[28] Zhiyuan Cheng, James Caverlee, and Kyumin Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 759–768. ACM, 2010.

[29] Richard Comerford. Interactive media: an internet reality. *IEEE Spectrum*, 33(4):29–32, 1996.

[30] Cathy C Conrad and Krista G Hilchey. A review of citizen science and community-based environmental monitoring: issues and opportunities. *Environmental monitoring and assessment*, 176(1-4):273–291, 2011.

[31] Caren B Cooper, Janis Dickinson, Tina Phillips, and Rick Bonney. Citizen science as a tool for conservation in residential ecosystems. *Ecology and Society*, 12(2):11, 2007.

[32] Jeff Cox and Beth Plale. Improving automatic weather observations with the public twitter stream. *IU School of Informatics and Computing*, 2011.

[33] David J Crandall, Lars Backstrom, Daniel Huttenlocher, and Jon Kleinberg. Mapping the world's photos. In *Proceedings of the 18th international conference on World wide web*, pages 761–770. ACM, 2009.

[34] Patrick Dähne and John N Karigiannis. Archeoguide: System architecture of a mobile outdoor augmented reality system. In *null*, page 263. IEEE, 2002.

[35] Stefan Daume. Mining twitter to monitor invasive alien species - an analytical framework and sample information topologies. *Ecological Informatics*, 31:70–82, 2016.

[36] Stefan Daume, Matthias Albert, and Klaus von Gadow. Forest monitoring and social media–complementary data sources for ecosystem surveillance? *Forest Ecology and Management*, 316:9–20, 2014.

[37] Stefan Daume and Victor Galaz. "anyone know what species this is?"–twitter conversations as embryonic citizen science communities. *PloS one*, 11(3):e0151387, 2016.

[38] Milad Davari and Haleh Amintoosi. A survey on participant recruitment in crowdsensing systems. In *Computer and Knowledge Engineering (ICCKE), 2016 6th International Conference on*, pages 286–291. IEEE, 2016.

[39] Ranieri de Brito Moreira, Lıvia Castro Degrossi, and Joao Porto de Albuquerque. An experimental evaluation of a crowdsourcing-based approach for flood risk management. In *Paper presented at the Conference: 12th Workshop on Experimental Software Engineering (ESELAW), at Lima, Peru*, 2015.

[40] Bertrand De Longueville, Robin S Smith, and Gianluca Luraschi. OMG, from here, I can see the flames!: a use case of mining location based social networks to acquire spatio-temporal data on forest fires. In *Proceedings of the 2009 international workshop on location based social networks*, pages 73–80. ACM, 2009.

[41] Lisette De Vries, Sonja Gensler, and Peter SH Leeflang. Popularity of brand posts on brand fan pages: An investigation of the effects of social media marketing. *Journal of Interactive Marketing*, 26(2):83–91, 2012.

[42] Christopher M DeBeer and John W Pomeroy. Modelling snow melt and snowcover depletion in a small alpine cirque, canadian rocky mountains. *Hydrological processes*, 23(18):2584–2599, 2009.

[43] Lívia Castro Degrossi, JP Albuquerque, Maria Clara Fava, and Eduardo Mario Mendiondo. Flood citizen observatory: a crowdsourcing-based approach for flood risk management in brazil. In *26th Int. Conf. on Software Engineering and Knowledge Engineering*, 2014.

[44] Dong-Po Deng, Guan-Shuo Mai, Tyng-Ruey Chuang, Rob Lemmens, and Kwang-Tsao Shao. Social web meets sensor web: From user-generated content to linked crowdsourced observation data. In *LDOW*, 2014.

[45] Andreas Juergen Dietz, Claudia Kuenzer, Ursula Gessner, and Stefan Dech. Remote sensing of snow–a review of available methods. *International Journal of Remote Sensing*, 33(13):4094–4134, 2012.

[46] Céline Dizerens. Georectification and snow classification of webcam images: potential for complementing satellite-derrived snow maps over switzerland. Master's thesis, University of Bern, 2015.

[47] Luca Egli. Spatial variability of new snow amounts derived from a dense network of alpine automatic stations. *Annals of Glaciology*, 49(1):51–55, 2008.

[48] Eric Enge. Hard numbers for public posting activity on google plus.[online] available https://www.stonetemple.com/real-numbers-for-the-activity-on-google-plus. 2015.

[49] Daniel Farinotti, Jan Magnusson, Matthias Huss, and Andreas Bauder. Snow accumulation distribution inferred from time-lapse photography and simple modelling. *Hydrological processes*, 24(15):2087–2097, 2010.

[50] R. Fedorov, A. Camerada, P. Fraternali, and M. Tagliasacchi. Estimating snow cover from publicly available images. *IEEE Transactions on Multimedia*, PP(99), 2016.

[51] Roman Fedorov, Alessandro Camerada, Piero Fraternali, and Marco Tagliasacchi. Estimating snow cover from publicly available images. *arXiv preprint arXiv:1508.01055*, 2015.

[52] Roman Fedorov, Darian Frajberg, and Piero Fraternali. A framework for outdoor mobile augmented reality and its application to mountain peak detection. In *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, pages 281–301. Springer, 2016.

[53] Roman Fedorov, Piero Fraternali, and Chiara Pasini. Snowwatch: a multi-modal citizen science application. In *International Conference on Web Engineering*, pages 538–541. Springer, 2016.

[54] Roman Fedorov, Piero Fraternali, and Marco Tagliasacchi. Mountain peak identification in visual content based on coarse digital elevation models. In *Proceedings of the 3rd ACM International Workshop on Multimedia Analysis for Ecological Data*, 2014.

[55] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[56] Emilio Ferrara, Roberto Interdonato, and Andrea Tagarelli. Online popularity and topical interests through the lens of instagram. In *Proceedings of the 25th ACM conference on Hypertext and social media*, pages 24–34. ACM, 2014.

[57] Colin J Ferster and Nicholas C Coops. A review of earth observation using mobile personal communication devices. *Computers & Geosciences*, 51:339–349, 2013.

[58] Eileen Fischer and A Rebecca Reuber. Social interaction via new social media:(how) can interactions on twitter affect effectual thinking and behavior? *Journal of business venturing*, 26(1):1–18, 2011.

[59] William Floyd and Markus Weiler. Measuring snow accumulation and ablation dynamics during rain-on-snow events: innovative measurement techniques. *Hydrological processes*, 22(24):4805–4812, 2008.

[60] Piero Fraternali, Andrea Castelletti, Rodolfo Soncini-Sessa, C Vaca Ruiz, and Andrea Emilio Rizzoli. Putting humans in the loop: Social computing for water resources management. *Environmental Modelling & Software*, 37:68–77, 2012.

[61] Piero Fraternali, Marco Tagliasacchi, Davide Martinenghi, Alessandro Bozzon, Ilio Catallo, Eleonora Ciceri, Francesco Nucci, Vincenzo Croce, Ismail Sengor Altingovde, Wolf Siberski, et al. The cubrik project: human-enhanced time-aware multimedia search. In *Proceedings of the 21st International Conference on World Wide Web*, pages 259–262. ACM, 2012.

[62] S. Galelli, G.B. Humphrey, H.R. Maier, and et al. An evaluation framework for input variable selection algorithms for environmental data-driven models. *Environmental Modelling & Software*, 62:33–51, 2014.

[63] LUCA Galli, PIERO Fraternali, CHIARA Pasini, GIORGIA Baroffio, A Dos Santos, ROBERTO Acerbis, and VALENTINA Riva. A gamification framework for customer engagement and sustainable water usage promotion. In *36th IAHR World Congress, Delft, Holland*, 2015.

[64] J Garvelmann, S Pohl, and M Weiler. From observation to the quantification of snow processes with a time-lapse camera network. *Hydrology and Earth System Sciences*, 17:1415–1429, 2013.

[65] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006.

[66] M. Giuliani and A. Castelletti. Is robustness really robust? how different definitions of robustness impact decision-making under climate change. *Climatic Change*, 135:409–424, 2016.

[67] M. Giuliani, A. Castelletti, F. Pianosi, E. Mason, and P.M. Reed. Curses, tradeoffs, and scalable management: advancing evolutionary multi-objective direct policy search to improve water reservoir operations. *Journal of Water Resources Planning and Management*, 142(2), 2016.

[68] M. Giuliani, J.D. Herman, A. Castelletti, and P.M. Reed. Many-objective reservoir policy identification and refinement to reduce policy inertia and myopia in water management. *Water Resources Research*, 50:3355–3377, 2014a. doi: 10.1002/2013WR014700.

[69] M. Giuliani, E. Mason, A. Castelletti, F. Pianosi, and R. Soncini-Sessa. Universal approximators for direct policy search in multi-purpose water reservoir management: A comparative analysis. In *Proceedings of the 19th IFAC World Congress*, Cape Town (South Africa), 24-29 August 2014.

[70] M. Giuliani, F. Pianosi, and A. Castelletti. Making the most of data: an information selection and assessment framework to improve water systems operations. *Water Resources Research*, 2015b. (under review).

[71] Matteo Giuliani, Andrea Castelletti, Roman Fedorov, and Piero Fraternali. Using crowdsourced web content for informing water systems operations in snow-dominated catchments. *Hydrology and Earth System Sciences*, 20(12):5049, 2016.

[72] K.E. Gleason, A.W. Nolin, and T.R. Roth. Developing a representative snow monitoring network in a forested mountain watershed. *Hydrology and Earth System Sciences - Discussion*, 2016.

[73] Michael F Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221, 2007.

[74] Michael F Goodchild. Citizens as voluntary sensors: Spatial data infrastructure in the world of web 2.0. *International Journal of Spatial Data Infrastructures Research*, 2(2), 2007.

[75] Hugh S Gorman and Erik M Conway. Monitoring the environment: Taking a historical perspective. *Environmental monitoring and assessment*, 106(1):1–10, 2005.

[76] Cristina Gouveia, Alexandra Fonseca, António Câmara, and Francisco Ferreira. Promoting the use of environmental data collected by concerned citizens through information and communication technologies. *Journal of environmental management*, 71(2):135–154, 2004.

[77] Eric A Graham, Erin C Riordan, Eric M Yuen, Deborah Estrin, and Philip W Rundel. Public internet-connected cameras used as a cross-continental ground-based plant phenology monitoring system. *Global Change Biology*, 16(11):3014–3023, 2010.

[78] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1458–1465. IEEE, 2005.

[79] Linda Green, Elizabeth Herron, and Arthur J Gold. Uri watershed watch.

[80] D. Hadka and P.M. Reed. Borg: An Auto–Adaptive Many–Objective Evolutionary Computing Framework. *Evolutionary Computation*, 21(2):231–259, 2013.

[81] Dorothy K Hall, George A Riggs, Vincent V Salomonson, Nicolo E DiGirolamo, and Klaus J Bayr. Modis snow-cover products. *Remote sensing of Environment*, 83(1):181–194, 2002.

[82] Samiul Hasan, Xianyuan Zhan, and Satish V Ukkusuri. Understanding urban human activity and mobility patterns using large-scale location-based data from online social media. In *Proceedings of the 2nd ACM SIGKDD international workshop on urban computing*, page 6. ACM, 2013.

[83] T. Hashimoto, J.R. Stedinger, and D.P. Loucks. Reliability, resilience, and vulnerability criteria for water resource system performance evaluation. *Water Resources Research*, 18(1):14–20, 1982.

[84] Jørgen Hinkler, Steen Birkelund Pedersen, Morten Rasch, and Birger Ulf Hansen. Automatic snow cover monitoring at high temporal and spatial resolution, using images taken by a standard digital camera. *International Journal of Remote Sensing*, 23(21):4669–4682, 2002.

[85] Tin Kam Ho. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE, 1995.

[86] Nadav Hochman and Raz Schwartz. Visualizing instagram: Tracing cultural visual rhythms. In *Proceedings of the Workshop on Social Media Visualization (SocMedVis) in conjunction with the Sixth International AAAI Conference on Weblogs and Social Media (ICWSM–12)*, pages 6–9, 2012.

[87] Ove Hoegh-Guldberg. Climate change, coral bleaching and the future of the world's coral reefs. *Marine and freshwater research*, 50(8):839–866, 1999.

[88] Ladislav Holko, Liudmyla Gorbachova, and Zdeněk Kostka. Snow hydrology in central europe. *Geography Compass*, 5(4):200–218, 2011.

[89] Otto Hyvärinen and Elena Saltikoff. Social media as a source of meteorological observations. *Monthly Weather Review*, 138(8):3175–3184, 2010.

[90] D Ings et al. Web services–human task (ws-humantask) specification version 1.1. *OASIS Committee Specification (August 2010).*

[91] Alan Irwin. *Citizen science: A study of people, expertise and sustainable development*. Psychology Press, 1995.

[92] Nathan Jacobs, Walker Burgin, Nick Fridrich, Austin Abrams, Kylia Miskell, Bobby H Braswell, Andrew D Richardson, and Robert Pless. The global network of outdoor webcams: properties and applications. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 111–120. ACM, 2009.

[93] Puneet Jain, Justin Manweiler, and Romit Roy Choudhury. Overlay: Practical mobile augmented reality. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pages 331–344. ACM, 2015.

[94] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65. ACM, 2007.

[95] Adrianne Jeffries. The man behind flickr on making the service 'awesome again'. *The Verge*, 20, 2013.

[96] Andrej Karpathy. Cs231n: Convolutional neural networks for visual recognition. *Neural networks*, 1, 2016.

[97] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[98] Felix Keis and Kevin Wiesner. Participatory sensing utilized by an advanced meteorological nowcasting system. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pages 1–6. IEEE, 2014.

[99] Sunyoung Kim, Jennifer Mankoff, and Eric Paulos. Sensr: evaluating a flexible framework for authoring mobile data-collection tools for citizen science. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 1453–1462. ACM, 2013.

[100] Asanobu Kitamoto and Takeshi Sagara. Toponym-based geotagging for observing precipitation from social and scientific data streams. In *Proceedings of the ACM multimedia 2012 workshop on Geotagging and its applications in multimedia*, pages 23–26. ACM, 2012.

[101] Max König, Jan-Gunnar Winther, and Elisabeth Isaksson. Measuring snow and glacier ice properties from satellite. *Reviews of Geophysics*, 39(1):1–27, 2001.

[102] Paul L Krapivsky and Sidney Redner. A statistical physics perspective on web growth. *Computer Networks*, 39(3):261–276, 2002.

[103] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[104] John Krumm, Nigel Davies, and Chandra Narayanaswami. User-generated content. *IEEE Pervasive Computing*, 7(4):10–11, 2008.

[105] K.E. Kunkel, D.A. Robinson, S. Champion, X. Yin, T. Estilow, and R.M. Frankson. Trends and extremes in northern hemisphere snow characteristics. *Current Climate Change Reports*, 2(2):65–73, 2016.

[106] Dominique Laffly, E Bernard, Madeleine Griselin, Florian Tolle, Jean-Michel Friedt, G Martin, and Christelle Marlin. High temporal resolution monitoring of snow cover using oblique view ground-based pictures. *Polar Record*, 48(01):11–16, 2012.

[107] Shyamal Lakshminarayanan. Using citizens to do science versus citizens as scientists. *Ecology and Society*, 12(2):2, 2007.

[108] AS Laliberte, Albert Rango, JE Herrick, Ed L Fredrickson, and Laura Burkett. An object-based image analysis approach for determining fractional cover of senescent and green vegetation with digital plot photography. *Journal of Arid Environments*, 69(1):1–14, 2007.

[109] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.

[110] Stefan Lee and David Crandall. Learning to identify local flora with human feedback. In *IEEE Computer Vision and Pattern Recognition. Workshop on Computer Vision and Human Computation*, 2014.

[111] Stanley Lemeshow and David W Hosmer. A review of goodness of fit statistics for use in the development of logistic regression models. *American journal of epidemiology*, 115(1):92–106, 1982.

[112] Daniel Leung and Shawn Newsam. Proximate sensing: Inferring what-is-where from georeferenced photo collections. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2955–2962. IEEE, 2010.

[113] Yunpeng Li, Noah Snavely, and Daniel P Huttenlocher. Location recognition using prioritized feature matching. In *European conference on computer vision*, pages 791–804. Springer, 2010.

[114] Wen-Nung Lie, Tom C-I Lin, Ting-Chih Lin, and Keng-Shen Hung. A robust dynamic programming algorithm to extract skyline in images for navigation. *Pattern recognition letters*, 26(2):221–230, 2005.

[115] Yu-Pin Lin, Dongpo Deng, Wei-Chih Lin, Rob Lemmens, Neville D Crossman, Klaus Henle, and Dirk S Schmeller. Uncertainty analysis of crowd-sourced and professionally collected field data used in species distribution models of taiwanese moths. *Biological Conservation*, 181:102–110, 2015.

[116] Jack Lindamood, Raymond Heatherly, Murat Kantarcioglu, and Bhavani Thuraisingham. Inferring private information using social network data. In *Proceedings of the 18th international conference on World wide web*, pages 1145–1146. ACM, 2009.

[117] Kathleen E Little, Masaki Hayashi, and Steve Liang. Community-based groundwater monitoring network using a citizen-science approach. *Groundwater*, 2015.

[118] Hai-Ying Liu, Irene Eleta, Mike Kobernus, and Tom Cole-Hunter. Analysis of public interest in environmental health information: Fine tuning content for dissemination via social media. In *International Workshop on the Internet for Financial Collective Awareness and Intelligence*, pages 129–146. Springer, 2016.

[119] Wei-Han Liu and Chih-Wen Su. Automatic peak recognition for mountain images. In *Advanced Technologies, Embedded and Multimedia for Human-centric Computing*, volume 260, pages 1115–1121. Springer Netherlands, 2014.

[120] Xiaoye Liu, Zhenyu Zhang, Jim Peterson, and Shobhit Chandra. Lidar-derived high quality ground control information and dem for image orthorectification. *GeoInformatica*, 11(1):37–53, 2007.

[121] JI López-Moreno and D Nogués-Bravo. Interpolating local snow depth data: an evaluation of methods. *Hydrological processes*, 20(10):2217–2232, 2006.

[122] Christopher S Lowry and Michael N Fienen. Crowdhydrology: crowdsourcing hydrologic data and engaging citizen scientists. *Ground Water*, 51(1):151–156, 2013.

[123] H.R. Maier, Z. Kapelan, J. Kasprzyk, and et al. Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions . *Environmental Modelling & Software*, 62(0):271–299, 2014.

[124] Nicolas Maisonneuve, Matthias Stevens, Maria E Niessen, and Luc Steels. Noisetube: Measuring and mapping noise pollution with mobile phones. In *Information technologies in environmental engineering*, pages 215–228. Springer, 2009.

[125] Irene Garcia Martí, Luis E Rodríguez, Mauricia Benedito, Sergi Trilles, Arturo Beltrán, Laura Díaz, and Joaquín Huerta. Mobile application for noise pollution monitoring through gamification techniques. In *International Conference on Entertainment Computing*, pages 562–571. Springer, 2012.

[126] Nargess Memarsadeghi. Citizen science [guest editors' introduction]. *Computing in Science Engineering*, 17(4):8–10, July 2015.

[127] N Michelsen, H Dirks, S Schulz, S Kempe, M Al-Saud, and C Schüth. Youtube as a crowd-generated water level archive. *Science of the Total Environment*, 568:189–195, 2016.

[128] Alvaro Moreno, Bas Amelung, and Lorena Santamarta. Linking beach recreation to weather conditions: a case study in zandvoort, netherlands. *Tourism in Marine Environments*, 5(2-1):111–119, 2008.

[129] Philip W Mote, Alan F Hamlet, Martyn P Clark, and Dennis P Lettenmaier. Declining mountain snowpack in western North America. *Bulletin of the American meteorological Society*, 86(1):39–49, 2005.

[130] Anastasia Moumtzidou, Symeon Papadopoulos, Stefanos Vrochidis, Ioannis Kompatsiaris, Konstantinos Kourtidis, George Hloupis, Ilias Stavrakas, Konstantina Papachristopoulou, and Christodoulos Keratidis. Towards air quality estimation using collected multimodal environmental data. In *International Workshop on Internet and Social Media for Environmental Monitoring*. Springer, 2016.

[131] Jonathan Muñoz, Jose Infante, Tarendra Lakhankar, and et al. Synergistic use of remote sensing for snow cover and snow water equivalent estimation. *British Journal of Environment & Climate Change*, 3(4):612–627, 2013.

[132] Calvin Murdock, Nathan Jacobs, and Robert Pless. Webcam2satellite: Estimating cloud maps from webcam imagery. In *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, pages 214–221. IEEE, 2013.

[133] Thomas Gru Newald and Michael Lehning. Altitudinal dependency of snow amounts in two small alpine catchments: can catchment-wide snow amounts be estimated via single snow or precipitation stations? *Annals of Glaciology*, 52(58):153–158, 2011.

[134] PR Newswire. Facebook reports fourth quarter and full year 2013 results. *MENLO PARK, Calif., Jan*, 29:2014, 2014.

[135] Anne W Nolin and Jeff Dozier. Estimating snow grain size using aviris data. *Remote Sensing of Environment*, 44(2):231–238, 1993.

[136] Timothy Nyerges and Michael Barndt. Public participation geographic information systems. In *Proceedings, Auto-Carto 13*. Citeseer, 1997.

[137] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

[138] J Parajka and G Blöschl. Validation of MODIS snow cover images over Austria. *Hydrology and Earth System Sciences Discussions*, 3(4):1569–1601, 2006.

[139] J Parajka and G Blöschl. Spatio-temporal combination of MODIS images–potential for snow cover mapping. *Water Resources Research*, 44(3), 2008.

[140] Juraj Parajka, Peter Haas, Robert Kirnbauer, Josef Jansa, and Günter Blöschl. Potential of time-lapse photography of snow for hydrological purposes at the small catchment scale. *Hydrological Processes*, 26(22):3327–3337, 2012.

[141] M. Pepe, P.A. Brivio, A. Rampini, F.R. Nodari, and M. Boschetti. Snow cover monitoring in alpine regions using ENVISAT optical data. *International Journal of Remote Sensing*, 26(21):4661–4667, 2005.

[142] P. Perona, E. Daly, B. Crouzy, and A. Porporato. Stochastic dynamics of snow avalanche occurrence by superposition of poisson processes. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 468(2148):4193–4208, 2012.

[143] Lara Piccolo, Miriam Fernández, Harith Alani, Arno Scharl, Michael Föls, and David Herring. Climate change engagement: Results of a multi-task game with a purpose. In *Proceedings of the 1st international workshop on Social Web for Environmental and Ecological Monitoring*. AAAI Publications, 2016.

[144] Lorenzo Porzi, Samuel Rota Buló, Paolo Valigi, Oswald Lanz, and Elisa Ricci. Learning contours for automatic annotations of mountains pictures on a smartphone. In *Proceedings of the International Conference on Distributed Smart Cameras*, page 13. ACM, 2014.

[145] Alexander Prokop. Assessing the applicability of terrestrial laser scanning for spatial snow depth measurements. *Cold Regions Science and Technology*, 54(3):155 – 163, 2008.

[146] Roy A Rappaport. *Pigs for the ancestors: Ritual in the ecology of a New Guinea people*. Waveland Press, 2000.

[147] Sasank Reddy, Katie Shilton, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava. Evaluating participation and performance in participatory sensing. *UrbanSense08*, page 1, 2008.

[148] Gerhard Reitmayr and Tom Drummond. Going out: robust model-based tracking for outdoor augmented reality. In *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2006.

[149] Dominic Rüfenacht, Matthew Brown, Jan Beutel, and Sabine Süsstrunk. Temporally consistent snow cover estimation from noisy, irregularly sampled measurements. In *Proc. 9th International Conference on Computer Vision Theory and Applications*, 2014.

[150] Mark A Ruzon and Carlo Tomasi. Color edge detection with the compass operator. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference On.*, volume 2, pages 160–166. IEEE, 1999.

[151] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010.

[152] Vincent V Salomonson and I Appel. Estimating fractional snow cover from modis using the normalized difference snow index. *Remote sensing of environment*, 89(3):351–360, 2004.

[153] Rosamaria Salvatori, Paolo Plini, Marco Giusto, and et al. Snow cover monitoring with images from digital camera systems. *Italian Journal of Remote Sensing*, 43:137–145, 2011.

[154] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245, 2013.

[155] Olivier Saurer, Georges Baatz, Kevin Köser, Marc Pollefeys, et al. Image based geo-localization in the alps. *International Journal of Computer Vision*, pages 1–13, 2015.

[156] E Schnebele et al. Improving remote sensing flood assessment using volunteered geographical data. *Natural Hazards and Earth System Sciences*, 13(3):669, 2013.

[157] Martin Schuler, E Stucki, Oliver Roque, and Manfred Perlik. Mountain Areas in Europe: Analysis of mountain areas in EU member states, acceding and other European countrie. Technical Report 2002.CE.16.0.AT.136, Nordic Centre for Spatial Development, 2004.

[158] J. Schweizer, C. Mitterer, and L. Stoffel. On forecasting large and infrequent snow avalanches. *Cold Regions Science and Technology*, 59(2):234–241, 2009.

[159] Pavel Serdyukov, Vanessa Murdock, and Roelof Van Zwol. Placing flickr photos on a map. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 484–491. ACM, 2009.

[160] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[161] Jiancheng Shi, J. Dozier, and H. Rott. Snow mapping in alpine regions with synthetic aperture radar. *Geoscience and Remote Sensing, IEEE Transactions on*, 32(1):152–158, Jan 1994.

[162] Jonathan Silvertown. A new dawn for citizen science. *Trends in ecology & evolution*, 24(9):467–471, 2009.

[163] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[164] Vivek K Singh, Mingyan Gao, and Ramesh Jain. Social pixels: genesis and evaluation. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 481–490. ACM, 2010.

[165] Craig Smith. How many people use the top social media, apps & services. *Digital Marketing Ramblings. Recuperado de http://expandedramblings. com/index. php/resource-how-many-people-use-the-top-social-media*, 2013.

[166] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.

[167] Richard Stafford, Adam G Hart, Laura Collins, Claire L Kirkhope, Rachel L Williams, Samuel G Rees, Jane R Lloyd, and Anne E Goodenough. Eu-social science: the role of internet social networks in the collection of bee biodiversity data. *PloS one*, 5(12):e14381, 2010.

[168] Severin Stähly, Heide Friedrich, and Martin Detert. Size ratio of fluvial grains' intermediate axes assessed by image processing and square-hole sieving. *Journal of Hydraulic Engineering*, 143(6):06017005, 2017.

[169] Maria Staudinger, Kerstin Stahl, and Jan Seibert. A drought index accounting for snow. *Water Resources Research*, 50(10):7861–7872, 2014.

[170] Brian L Sullivan, Christopher L Wood, Marshall J Iliff, Rick E Bonney, Daniel Fink, and Steve Kelling. ebird: A citizen-based bird observation network in the biological sciences. *Biological Conservation*, 142(10):2282–2292, 2009.

[171] Mengfan Tang, Pranav Agrawal, Siripen Pongpaichet, and Ramesh Jain. Geospatial interpolation analytics for data streams in eventshop. In *Multimedia and Expo (ICME), 2015 IEEE International Conference on*, pages 1–6. IEEE, 2015.

[172] A Murat Tekalp. *Digital video processing*. Prentice Hall Press, 2015.

[173] Dallen J Timothy and David L Groves. Research note: webcam images as potential data sources for tourism research. 2001.

[174] Goldee Udani. An exhaustive study of twitter users across the world.[online] available http://www. beevolve. com/twitter-statistics. 2012.

[175] Johan Ugander, Brian Karrer, Lars Backstrom, and Cameron Marlow. The anatomy of the facebook social graph. *arXiv preprint arXiv:1111.4503*, 2011.

[176] Willem W Verstraeten, Bart Vermeulen, Jan Stuckens, Stefaan Lhermitte, Dimitry Van der Zande, Marc Van Ranst, and Pol Coppin. Webcams for bird detection and monitoring: A demonstration study. *Sensors*, 10(4):3480–3503, 2010.

[177] Jingya Wang, Mohammed Korayem, Saul Blanco, and David J Crandall. Tracking natural events through social media and computer vision. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 1097–1101. ACM, 2016.

[178] Jingya Wang, Mohammed Korayem, and David J Crandall. Observing the natural world with flickr. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 452–459. IEEE, 2013.

[179] Alexandra Weilenmann, Thomas Hillman, and Beata Jungselius. Instagram at the museum: communicating the museum experience through social photo sharing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1843–1852. ACM, 2013.

[180] Graham Whitelaw, Hague Vaughan, Brian Craig, and David Atkinson. Establishing the canadian community monitoring network. *Environmental monitoring and assessment*, 88(1-3):409–418, 2003.

[181] A.W. Wood and D.P. Lettenmaier. A test bed for new seasonal hydrologic forecasting approaches in the western united states. *Bulletin of the American Meteorological Society*, 87(12):1699, 2006.

[182] Jianxiong Xiao, Krista A Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, pages 1–20, 2014.

[183] Jun Yang, Yu-Gang Jiang, Alexander G Hauptmann, and Chong-Wah Ngo. Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 197–206. ACM, 2007.

[184] F. Yokota and K.M. Thompson. Value of information analysis in environmental health risk management decisions: past, present, and future. *Risk analysis*, 24(3):635–650, 2004.

[185] J. Zatarain-Salazar, P.M. Reed, J.D. Herman, M. Giuliani, and A. Castelletti. A diagnostic assessment of evolutionary algorithms for multi-objective surface water reservoir control. *Advances in Water Resources*, 92:172–185, 2016.

[186] Haipeng Zhang, Mohammed Korayem, David J Crandall, and Gretchen LeBuhn. Mining photo-sharing websites to study ecological phenomena. In *Proceedings of the 21st international conference on World Wide Web*, pages 749–758. ACM, 2012.

[187] Tong Zhang. An introduction to support vector machines and other kernel-based learning methods. *AI Magazine*, 22(2):103, 2001.

[188] Gabe Zichermann and Christopher Cunningham. *Gamification by design: Implementing game mechanics in web and mobile apps*. " O'Reilly Media, Inc.", 2011.