

Registration of single landscape photographs with 3D landscape models

THÈSE N° 6650 (2015)

PRÉSENTÉE LE 26 JUIN 2015

À LA FACULTÉ DE L'ENVIRONNEMENT NATUREL, ARCHITECTURAL ET CONSTRUIT
LABORATOIRE DE SYSTÈMES D'INFORMATION GÉOGRAPHIQUE
PROGRAMME DOCTORAL EN GÉNIE CIVIL ET ENVIRONNEMENT

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Timothée PRODUIT

acceptée sur proposition du jury:

Prof. A. Berne, président du jury
Prof. F. Golay, Prof. D. Tuia, directeurs de thèse
Dr C. Strecha, rapporteur
Prof. F. Remondino, rapporteur
Prof. J.-Ph. Thiran, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2015

Der Liebhaber-Photograph wird Kenner und Freund der Heimat,
die er auf der Suche nach fesselnden Bildern durchstreift;
er wird auch bei manchen andern durch eine Aufnahme,
die in der Wahl des Standpunktes und der Belichtung glücklich ist. (...)
Man hat es oft bedauert, dass von den Tausenden trefflicher Photographien,
die wohl Jahr für Jahr allein in der Schweiz entstehen,
so wenig bekannt und Allgemeingut wird.

Jules Coulin,
Amateur-Photographie, Zur Postkarten-Ausstellung des Schweiz. Amateur-Photographen-Verbandes.
Zeitschrift der «Schweizer. Vereinigung für Heimatschutz»
Oktober 1916.

Acknowledgements

First and foremost, I would like to thank my research supervisors, Prof. François Golay and Prof. Devis Tuia. During our discussions, François always helped me to step back to ensure the coherence of this research. Devis taught me the research profession. His arrival in the LASIG and his advice was pivotal for this thesis. I would like to thank you both very much for your support and understanding over these past four years.

The original impulse for the subject of this thesis comes from Prof. Pierre Frey who shared with us the problematic of the georeferencing of a postcard archive. I would like to thank Prof. Frey and his team of the and the *Archive de la Construction Moderne* to have provided us pictures used to test and illustrate our monoplottter.

I would also like to show my gratitude to the people who introduced me to the computer vision field. Dr. Christoph Strecha helped me find the thesis topic. His pertinent suggestions for the initial technical directions were decisive. Prof. Vincent Lepetit took the time to discuss with me and raised many precious points. Finally, the contribution of Julien Rebetez is essential for a chapter of this thesis. At the time of Dr. Jan Skaloud lessons, I did not suspect that his teaching would be so relevant in the near future. I also thank Jan for having provided me with the facilities to acquire my validation dataset and for our discussions.

I would also like to thank those who helped me to fill my knowledge gaps in statistics: Claudio Semadeni and Dr. Frank De Morsier.

Finally, I feel very fortunate that Gillian Milani completed his master thesis in our lab. His amazing programmer capabilities enabled me to present a functional and shareable monoplottter. The monoplottter evaluation also benefited from the expert advices of Claudio Bozzini, who I would like to thank for our valuable discussions.

Remerciements

Au-delà des collaborations scientifiques, quatre ans de doctorat représentent aussi le soutien de nombreux amis que je souhaiterais remercier ici.

Il y a les amis qui m'accompagnent en montagne, Hugues, Florence, Claire, Arnaud, Gabriel et Virgile. Je leur dois de nombreux week-end exténuants mais ressourçants et de tout aussi nombreuses discussions de geek de haute-montagne.

Pour pouvoir les suivre le week-end, de nombreux entraînements étaient nécessaires. Merci à Florence et Kevin de m'avoir forcé à m'extirper de derrière l'écran pour nos courses de midi. Merci aux partenaires de nocturnes hivernales aux Paccots : Hugues, Claire, Tristan, Xavier et Kevin.

Quatre ans c'est de nombreuses soirées à remplir (mais des verres à vider). Merci Cédric, Kevin, Hugues, Sylvain, Claire, Tristan, Xavier, Florence, Valérie, Bastien, André pour ces bons moments.

C'est quand les collègues deviennent des amis que l'on a du plaisir à venir travailler et il y en a eu plein au LASIG : Kevin, Sylvie, Devis, André, Stéphane, Ivo, Matthew, Ema, Estelle, Solange, Jessie, Nicolas, Marc, François, Jens et tous les étudiants, stagiaires et personnes de passage.

Merci à ma famille et aux amis du Mazot: Cédric, Gabriel et François.

Le lecteur attentif aura noté quelques *top scorers* dans ces remerciements :

Avec quatre merci, Kevin, tu arrives en tête. Pourtant, malgré le désavantage du terrain et ta passion du houblon, tu as su te hisser sur le haut du podium. Tu partages cette place avec une habituée de la performance sportive, culinaire et scientifique : Florence, tu es propulsée tout en haut par tes relectures de mes bafouillages en anglais. Merci à vous deux d'avoir été aussi proches pendant cette thèse.

Puis sur la deuxième place quantitative mais pas qualitative, je te remercie Claire. Merci à toi de partager mon chemin au propre comme au figuré.

Abstract

Terrestrial imagery found in public databases is an alternative and complementary data source for environmental studies. Compared to usual data sources, such as airborne images, terrestrial pictures may have higher temporal and spatial resolutions. They are especially valuable in studying the past, when airborne acquisitions were more rare or inexistent. Nevertheless, terrestrial image databases are typically imperfectly georeferenced, if at all. Hence, it is difficult to retrieve images of interest in a database. Moreover, georeferencing processes proposed for these images are either totally manual or rely on the existence of a collection of similar and overlapping images. The first is time-consuming and the second is appropriate only for particular sets of images. An alternative and more general georeferencing procedure is the registration of a picture with a landscape model. In this thesis, we propose to use as the reference a 3D virtual globe and our aim is to retrieve the orientation and the location of the camera.

The location and orientation of a picture can be computed with correspondences that a user recognizes and clicks in a picture and in an orthoimage. However, these two distinct types of images are difficult to match because of the distortions resulting from the different viewpoints. Hence, we provide a navigation tool to scroll through a 3D model to ease this task. Indeed, to find the orientation of a picture, users typically exploit the skyline which appears only in a 3D view. In imitation, we develop a novel skyline-matching technique, based on Dynamic Time Warping, to compute the fine orientation of a picture. In a second phase, we assess whether the similarities between a 3D synthetic image and real images can be used to compute the orientation of a picture, such as illustrated in Figure 1. The matching of real images with a model is challenging. We show how images shared on the web can be inserted in the landscape model to ease the georeferencing. Finally, all of the presented pose estimation schemes assume that every location of the map is as likely to be a shooting spot. However, we show that geographic indicators related to accessibility and morphology are correlated with a database of picture locations. Exploiting these relations, our last contribution is a map of the attractiveness.

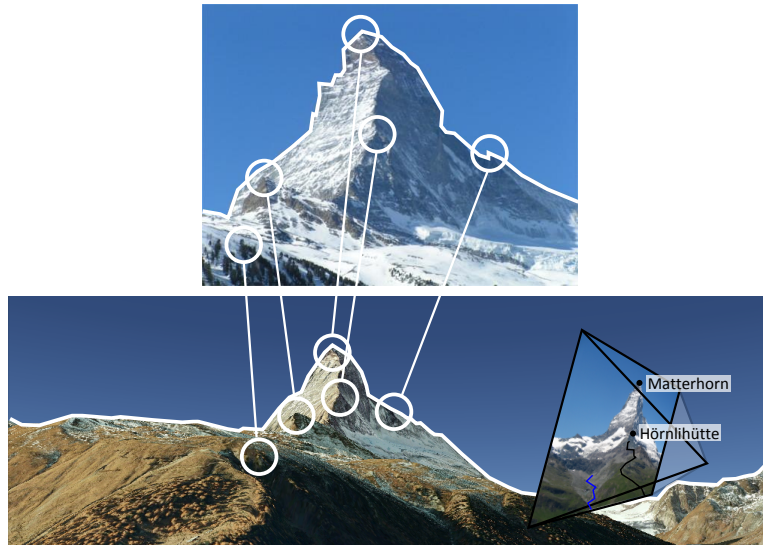


Figure 1 – Orientation of an image (top) with a landscape 3D model (bottom): skyline or visual 2D-3D correspondences are detected. Once oriented, the image interacts with geographic data.

In this thesis, we observed that pose estimation from the automatic registration of a real image with a landscape model cannot compete with the accuracy reached by an operator or a standard photogrammetric workflow. However, the proposed methods can greatly reduce the operator workload for the manual georeferencing by finding automatically correspondences with a 3D landscape model. Second, they are a valid alternative if the required conditions for Structure-from-Motion are not respected. Finally, the accuracy obtained is sufficient to explore image databases with spatial queries and to add geographic information in the pictures.

Registration of landscape images with landscape models has previously received limited attention. The tools developed in this thesis can ease the work of archivists (in the management and provision of efficient image databases), environmental scientists (for the extraction of georeferenced information from pictures) and the general public (to browse collections of augmented images inserted in a virtual globe).

Key words: 2D-3D Orientation, Monoplotting, Skyline Matching, Dynamic Time Warping, HOG, Landscape Model Alignment, Kalman Filter, Collective photography, Geographic One-Class Problem, Attractiveness.

Résumé

Les photographies terrestres qui se trouvent dans les bases de données publiques sont une représentation de l'état du paysage avec une haute résolution spatiale et temporelle. Elles peuvent donc servir de source de données alternatives pour des études de l'environnement. Elles sont ainsi complémentaires aux données plus classiques que sont les images aériennes, en représentant par exemple des régions à des périodes où les acquisitions aériennes n'étaient pas encore possibles. Cependant, le géoréférencement de photographies obliques de paysages est encore mal maîtrisé. Le géoréférencement permet de lier une image avec des attributs géographiques et aussi d'en extraire des paramètres environnementaux quantifiables.

Il existe principalement deux méthodes pour le géoréférencement d'images de paysage. La première est l'aéro-triangulation et n'est adaptée qu'à des collections d'images similaires se chevauchant largement. La seconde est le géoréférencement manuel qui demande beaucoup de temps à un opérateur. Alternativement, des études, dont fait partie cette thèse, proposent de trouver la position et l'orientation d'une photographie en la comparant à une modélisation géographique du territoire. Cette technique permet potentiellement de traiter de nombreux types d'images sans avoir à recourir à un opérateur. Les recherches qui vont dans ce sens proposent de comparer la ligne d'horizon d'une photographie (et autres lignes de rupture dues au relief) avec celle que l'on peut extraire d'un modèle 3D. Ces études supposent en général que la position de la caméra est connue, elles calculent alors l'orientation de la caméra. Dans cette thèse, nous proposons d'utiliser comme référence des images synthétiques réalistes générées depuis un Modèle Numérique de Terrain (MNT) drapé avec une image aérienne. Ces images synthétiques sont donc des images telles que celles produites par les globes virtuels. De plus, nous voulons calculer non-seulement l'orientation de la prise de vue mais aussi son lieu si celui-ci n'est qu'approximativement renseigné.

Un utilisateur peut calculer l'orientation et le lieu d'une prise de vue en cliquant sur des lieux similaires dans la photographie et dans une image aérienne. Cette tâche est cependant rendue difficile par la différence de point de vue entre une photographie acquise en perspective oblique et une image aérienne orthorectifiée. Il apparaît que la tâche de l'utilisateur est facilitée s'il peut orienter l'image dans une interface 3D et donc, par exemple, comparer les lignes d'horizon. Nous avons donc développé une interface 3D pour le géoréférencement d'images, celle-ci a été mise à disposition des utilisateurs de QGIS.

De plus, afin d'automatiser certaines étapes de ce processus, nous proposons trois méthodes. Nous avons développé dans un premier temps une méthode qui permet d'aligner les lignes d'horizon automatiquement. Cette méthode se base sur l'algorithme de déformation temporelle dynamique (Dynamic Time Warping, DTW) et permet d'orienter une image par rapport à un modèle 3D du territoire. Deuxièmement, tel qu'illustré dans la Figure 1, nous avons évalué la possibilité d'utiliser une image synthétique (générée avec un Modèle Numérique de Terrain et une orthoimage) comme source de correspondances. Troisièmement, nous proposons d'intégrer au modèle géographique des images partagées sur le web et qui peuvent faciliter le géoréférencement. Finalement, ces méthodes de calcul de l'orientation et du lieu de prise de vue supposent que tous les lieux sont autant probables d'être

un lieu de prise de vue. Nous avons démontré que ce n'est pas le cas et proposé une méthode pour calculer la probabilité d'un lieu en fonction de ses caractéristiques géographiques. Ceci nous permet de dessiner une carte d'attractivité.

Dans cette thèse, par les méthodes proposées, nous avons observé que la précision de l'orientation d'une photographie à l'aide d'un modèle du paysage ne permet ni d'atteindre la précision d'un opérateur ni celle de l'aéro-triangulation. Cependant, ces méthodes ont quand même des avantages. D'abord, elles sont la seule alternative si l'aérotriangulation n'est pas possible. Elles peuvent aussi permettre de limiter le travail d'un opérateur en trouvant des correspondances automatiquement entre la photographie et le modèle 3D. De plus, la précision atteinte est suffisante pour ajouter des fonctions géographiques aux bases de données d'images telles que des requêtes spatiales ou l'augmentation des photographies avec des données géographiques.

Le géoréférencement de photographies de paysage grâce à un modèle du territoire n'est encore que peu étudié bien qu'il ait de nombreuses applications potentielles. Les méthodes développées dans cette thèse pourraient par exemple faciliter le travail des archivistes (pour améliorer l'organisation et la recherche dans leurs bases de données), pour des chercheurs en environnement (pour leur faciliter l'accès et l'extraction de données depuis des photographies), pour le grand public (pour qu'il puisse visiter des collections d'images dans un contexte géographique tel qu'un globe virtuel). Dans cette thèse, nous avons voulu explorer le potentiel des données géographiques comme source de référence pour le géoréférencement de photographies. En effet, ces données permettent de considérer le géoréférencement d'images sur l'entier du globe alors que les méthodes classiques requièrent des sets d'images et ne sont donc applicables que pour des collections particulières ou autour de points d'intérêt.

Contents

Acknowledgements	i
Abstract (English/Français)	v
1 Introduction	1
1.1 Georeferencing landscape images	1
1.2 Computation of geographic attributes	3
1.3 Registration with a landscape model	4
1.3.1 Supervised georeferencing	4
1.3.2 Automatic registration	5
1.3.3 Metadata and case studies	6
1.4 Objectives	7
1.4.1 Monoplotter	7
1.4.2 Registration with a 3D landscape model	8
1.4.3 Learning priors	9
1.5 Organization	10
2 State-of-the-Art	11
2.1 History of the acquisition of georeferenced landscape images	11
2.1.1 Acquisition and georeferencing of photogrammetric pictures	11
2.1.2 Georeferencing general public photography	13
2.2 User interaction-based pose estimation	13
2.2.1 Rephotography, repeated-photography	14
2.2.2 Monophotogrammetry	15
2.3 Automatic pose estimation	15
2.3.1 Coarse localization of pictures	16
2.3.2 SfM-based localization	16
2.3.3 GIS data as reference	18
2.4 Landscape image registration	19
2.4.1 3DOF Orientation	19
2.4.2 6DOF orientation	19
2.5 Discussion	20
2.6 Contributions	21
3 Theory	23
3.1 Camera model	23
3.2 Camera orientation	25
3.2.1 Homogeneous representation	26

Contents

3.3	2D-3D Image Orientation or Space Resection	27
3.3.1	Least-Square Methods	27
3.3.2	Direct Linear Transform	28
3.3.3	Perspective-n-Points	29
3.4	Feature detection and matching	29
3.4.1	Feature detection	29
3.4.2	Feature description	33
3.4.3	Feature matching	33
3.5	Simultaneous pose estimation and matching	34
3.5.1	Noisy correspondences: RANSAC	34
3.5.2	No correspondences, but pose prior: Blind-PnP	36
3.5.3	Pose Estimation with Pose and Landscape Model Priors, PEP-ALP	37
4	Monoplotter: Pic2Map, a plugin for the integration of pictures in QGIS	41
4.1	Introduction	41
4.2	Description of a monoplotter	42
4.3	Description of the implementation	43
4.3.1	Structure of the database	43
4.3.2	Pose estimation	45
4.3.3	3D-rendering on the GPU	45
4.3.4	Map-to-image functions	46
4.3.5	Image-to-map functions	47
4.3.6	Integration with an open source GIS	48
4.4	Illustration of the functions and case study (Aletsch)	50
4.5	Discussion	53
4.6	Conclusion	55
5	Registration of a landscape image with a landscape model	57
5.1	Camera orientation with DTW-based Skyline alignment	59
5.1.1	Introduction	59
5.1.2	Review	59
5.1.3	Proposed method	62
5.1.4	Experiments	68
5.1.5	Discussion	75
5.1.6	Conclusion	75
5.2	Detection of correspondences between real and synthetic images with HOG	77
5.2.1	Introduction	77
5.2.2	Review	78
5.2.3	Method	79
5.2.4	Experiments	82
5.2.5	Discussion	85
5.2.6	Conclusion	88
5.3	Georeferencing an image collection	89
5.3.1	Introduction	89
5.3.2	Review	90
5.3.3	Method	91
5.3.4	Experiment	94
5.3.5	Discussion	99

5.3.6	Conclusion	102
5.4	Discussion of the proposed methods	104
5.4.1	Skyline and HOG registration, possible interaction	104
5.4.2	Focal length	106
5.4.3	Datasets and benchmark datasets	107
5.4.4	Reference data (DEM and orthoimages)	107
5.4.5	Alternative usage of landscape models as the reference	108
6	Learning prior:	
	Measuring attractiveness of a location	111
6.1	Introduction	111
6.2	Review	114
6.3	Problem formulation	115
6.3.1	One-Class Data and Geographic One-Class Data	116
6.4	Method	118
6.4.1	Geographic indicators computation	120
6.4.2	Geographic indicators preprocessing and decorrelation	120
6.4.3	Labelled data sets separation	120
6.4.4	Distribution estimation with kernel density estimation	120
6.4.5	Probability estimation and classification	121
6.4.6	Performance evaluation and setup	122
6.4.7	Validation	123
6.5	Data	123
6.6	Results	124
6.6.1	Distribution of the locations with respect to the geographic indicators	124
6.6.2	Analysis of a PCA of every geographic features	127
6.6.3	Numerical results for various combination of geographic features	129
6.6.4	Evaluation of the map	132
6.7	Discussion	136
6.7.1	Method	136
6.7.2	Geographic indicators	136
6.8	Conclusion	137
7	Conclusion	139
7.1	Goals	139
7.2	Results: discussion of the objectives	140
7.2.1	Monoplotter	140
7.2.2	Registration with a 3D landscape model	140
7.2.3	Learning location priors	143
7.3	Future research and applications	143
7.3.1	Further research	143
7.3.2	Applications	144
7.4	Concluding words	145
	Bibliography	153

Contents

Appendix	155
A Pic2map snapshots	155
B Les Diablerets dataset	158
C Skyline matching pseudo-codes	161
C.1 3DOF orientation	161
C.2 6DOF orientation	162
D Skyline matching results	163
E HOG matching pseudo-codes	165
E.1 Global matching	165
E.2 Multi-scale matching	166
F Georeferencing a collection of images.	167
Curriculum Vitae	168

Accronyms

3DOF Three Degrees of Freedom, camera having a known location. 7	KPCA Kernel Principal Component Analysis. 125
6DOF Six Degrees of Freedom, camera having unknown location and orientation. 8	NCC Normalized Cross Correlation. 37
DEM Digital Elevation model. 5	NNDR Nearest Neighbor Distance Ratio. 38
DLT Direct Linear Transform. 32	OCSVM One-Class Support Vector Machine. 125
DTW Dynamic Time Warping. 69	PCA Principal Component Analysis. 127
GCP Ground Control Point, 2D-2D or 2D-3D correspondences for georeferencing. 4	PDF Probability Density Function. 129
GIS Geographic Information System. 2	PEP-ALP Pose Estimation with Pose and Landscape model Priors. 41
GOCD Geographic One-Class Data. 125	PnP Perspective-n-Point. 32
GPS Global Positioning System, used in the sense of Global Navigation Satellite System. 4	PUL Positive and Unlabelled Learning. 126
GPU Graphical Processing Unit. 50	QGIS Quantum GIS, an Open Source GIS. 53
GUI Graphic User Interface. 53	RANSAC Random Sample Consensus. 39
HOG Histogram of Oriented Gradients. 87	SfM Structure from Motion. 19
ICP Iterative Closest Points. 66	SIFT Scale-Invariant Feature Transform. 37
IMU Inertial Measurement Unit, consists of accelerometers, gyroscopes and sometimes also magnetometers. 4	SSD Sum of Squared Difference. 37
KDE Kernel Density Estimation. 129	SVM Support Vector Machine. 88
	TLM Topographic Landscape Model. 132
	UAV Unmanned Aerial Vehicle. 1

1 Introduction

1.1 Georeferencing landscape images

From the very beginning of the photography, scientists noticed the potential of pictures to record and measure the environment. First photogrammetrists used cameras just like they used theodolites and acquired images from the ground. However, as soon as the technology allowed it, they brought the cameras in the air and thus enlarged the sensed area, obtaining pictures more similar to maps. With the advent of aerial sensors (mounted on planes, satellites and recently Unmanned Aerial Vehicles, UAV) terrestrial photogrammetry to record the landscape seems to have lost its attractiveness.

However, today, with the rise of photo-sharing websites and consumer cameras with processing abilities (i.e. smartphones and tablets), terrestrial photogrammetry is experiencing a renewed interest and faces the need of new strategies to meet current challenges. A particularly interesting challenge is the use of public databases of images as a sensor of the environment. On one hand, since the first World War, classic aerial sensors record the landscape at regular time steps and were engineered to ease the georeferencing of the images. On the other hand, in databases of general public images, complementary collections can be found: older and historical pictures, images acquired at higher frequency or between aerial campaigns. However, consumer images are generally not or poorly georeferenced and beyond the limits of standard georeferencing workflows. The georeferencing is mandatory to organize the database, retrieve images according to geographic criteria and extract environmental data from the photographs. We need then to develop efficient and general georeferencing methods.

This thesis focuses on landscape photographs that we will define as follows: pictures of scenery in open country, in which the focus is on natural features. In the specific family of landscape images, we can list: pictures shot to inform scientific campaigns or natural disasters, postcards, tourist pictures etc. Unless recorded with recent cameras or processed with state-of-the-art georeferencing methods, these images are generally not associated with metadata describing the picture location and orientation. Thus, the organization of the database is textual rather than spatial. For this reason, these photographs lose most of their value for collective memory or environmental studies.

Indeed, landscape images without metadata have a very low value for the community. Imagine that we want to study an image of the East face of the Matterhorn, this famous mountain in Switzerland. In a database of mountain images without metadata, as illustrated in the first column of Figure 1.1, we would have to look at every single image and select those recording the Matterhorn (provided that we know its shape). Fortunately, images are usually associated with some textual data (or tags in the

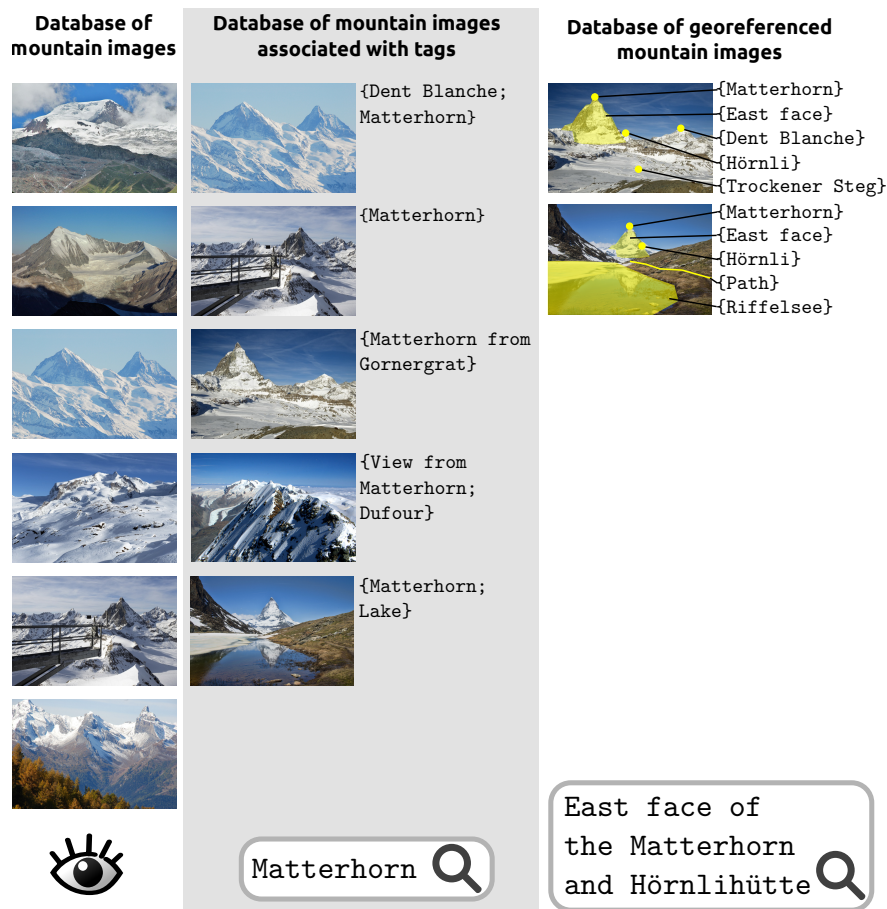


Figure 1.1 – Searching for the Matterhorn in a database of mountain images. First column, the images are not tagged. Second column, the images are tagged by the users. Last column, the images are georeferenced, the complete set of attributes enable the database with geographic queries as well as geographic data overlay.

web jargon), which describe the image content or location. With such information, we can restrict the scan to images associated with the word "Matterhorn", represented in the second column. During this operation, we will lose images tagged inadequately or get pictures for which the Matterhorn is not the subject but the location. Yet, if the images are georeferenced, we can select images representing the Matterhorn and even the orientation of the picture. We can also ask the name of a summit or overlay geographic layers on the picture, such as illustrated in the last column. Hence, the knowledge of the precise image orientation opens opportunities such as:

- Geographic selection in databases (not only with tags but also with the geographic content or location of the picture);
- Picture augmentation: geographic layers are overlaid on the images (summit names, points of interest, trails...);
- Integration of the images in a 2D (after ortho-rectification) or 3D Geographic Informations System (GIS);

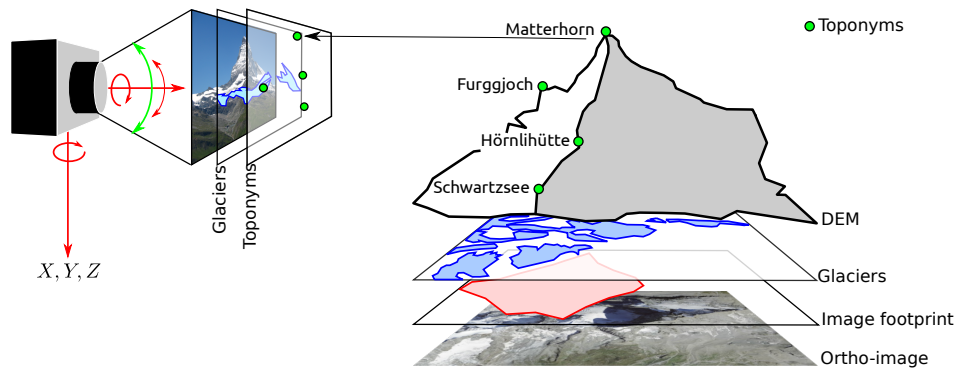


Figure 1.2 – If the location of the camera and its orientation are known, each pixel is related to its geographic location and thus with geographic layers. Location and rotation of the camera are illustrated in red and the field of view in green.

- Measurement of environmental parameters (change detection from historical photographs or collection of tourist's images).

These advanced functions require a direct relation between the pixels of the picture and geographic data, as illustrated in Figure 1.2. Implicitly, they require the knowledge of the picture location (where the camera is placed), orientation (in which direction it looks, both represented in red) and field of view (the zoom, in green). Nevertheless, the accurate *in situ* measurement of these parameters relies on specific instruments. Hence, most of the images collected are only associated with approximations of some of these parameters, if at all.

1.2 Computation of geographic attributes

In the previous section, we described why the georeferencing of landscape images is crucial to provide databases with efficient access to the pictures and subsequently use these pictures to analyze the evolution of the environment. Usually, the georeferencing of an image is modelled with its exterior orientation (in red in Figure 1.3) composed of the location and orientation of the camera and its interior orientation describing the internal geometry of the shooting (here the field-of-view in green).

The interior and exterior orientation are compulsory to link an image with geographic data. The interior orientation can be approximated from the focal length and the sensor size whereas the exterior orientation can be measured *in situ* for instance with a GPS, compass and IMU (namely direct georeferencing). With the recent advent of smartphones and tablets, equipped with these sensors, everyone can record pictures and their exterior orientation. However, accurate sensors are still very expensive and values measured with consumer sensors can only be considered as approximations. Moreover, the majority of images found in current databases were not recorded with such cameras.

The traditional workflow to georeference a collection of images is based on Structure-from-Motion (SfM). During SfM, similar key-points are recognized across the images. The camera orientations and 3D coordinates are retrieved from the relation of the 2D key-points. The absolute orientation is obtained by inserting geographic reference in the process (geographic coordinates of the camera location or Ground Control Points (GCP, locations in the image with known 3D geographic coordinates)). This process, which is entirely automated, is particularly efficient for sets of images with a large overlap

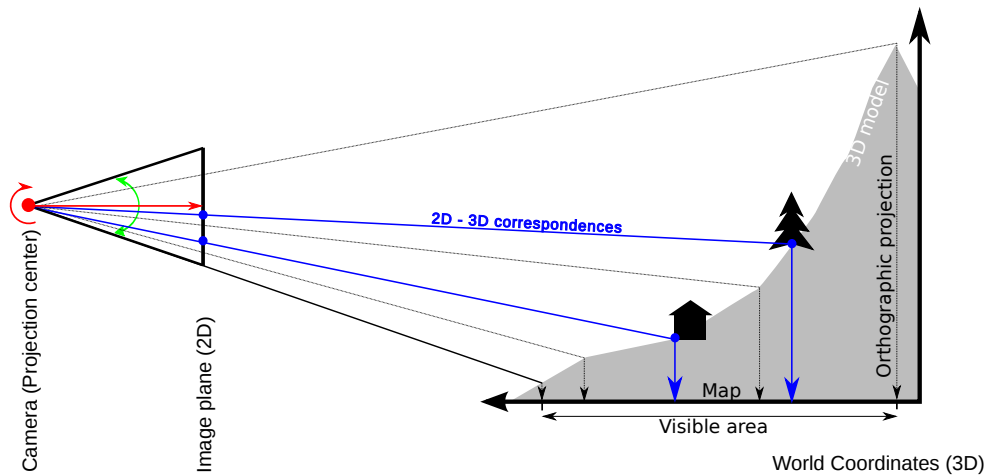


Figure 1.3 – Geometry of the problem: With an oriented camera (exterior orientation in red, interior in green), each pixel is projected on the 3D model of the landscape to compute their 3D coordinates. The orthographic projection is obtained with their horizontal components. 2D-3D correspondences are point features for which both the image coordinates and the world coordinates are known. They can be used to orient a camera.

and shot under the same conditions. This is why it is now the standard for the processing of UAV images. However, in the context of pictures recorded with many different cameras and under various environmental conditions, it can only be applied to particular sets of images. Furthermore, SfM is not directly applicable for the georeferencing of a single image, an important condition to answer the need for evolving databases, online georeferencing, and pictures without overlap. For a single photograph, either a new SfM is computed or it is matched with the 3D key-points resulting from the SfM.

If SfM cannot be applied, the remaining and laborious solution is to involve the user to compute exterior and interior orientation of each image. For instance, a user can browse a virtual globe and fix the image at a likely location and orientation. This tool is for instance provided by Google Earth (*Google Earth, Google Inc., Mountainview, 2015*). The user can also recognize and click on similar locations found in the picture and in an orthoimage. The interior and exterior orientation of the picture can be established from these GCP or 2D-3D correspondences (visible in blue in Figure 1.3).

1.3 Registration with a landscape model

1.3.1 Supervised georeferencing

In Figure 1.4, we illustrate the steps of the supervised georeferencing of a picture. Firstly, the region is identified (**A**). This step is simplified if the user recognizes the region represented in the picture or if the picture is linked with toponyms reducing the potential locations. Secondly, the user browses a virtual globe to orient roughly the image (**B**). Thirdly, he clicks on easily recognizable landmarks in the image (summits etc.) and in the 3D view (**C**). These approximate 2D-3D correspondences improve the orientation. At this stage, the photograph is well aligned with the landscape model, specifically for the background part. Finally, he detects precise GCP on the orthoimage, which is more precise than the 3D view, to improve again the orientation and reach the accuracy required to extract georeferenced

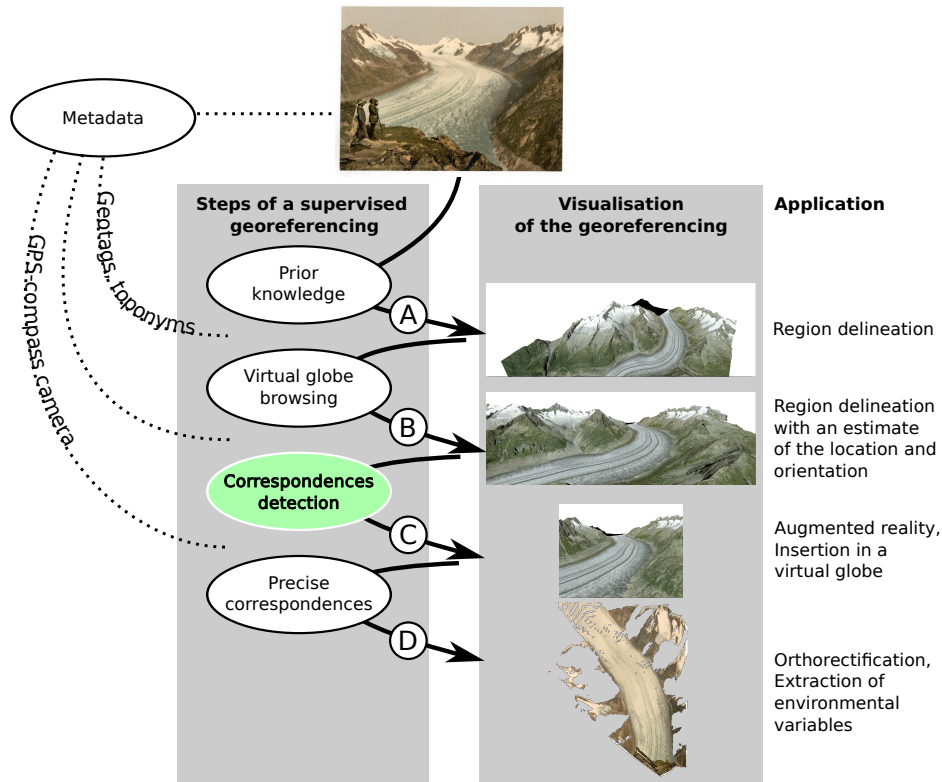


Figure 1.4 – Steps of the manual georeferencing of an image.

information from the picture (D). The metadata joined to the picture can short-cut this workflow by providing prior location or orientation. In this thesis, we will provide a, so called, monoplottor, which is used for the georeferencing of a picture following these steps. Unlike other implementations, it integrates a 3D interface for the browsing and digitization of GCP.

1.3.2 Automatic registration

In this process, we would like to ease the task of the operator (or even replace him). In this thesis, we will specifically act on the step which is highlighted in green in Figure 1.4. The goal of this step, such as illustrated in Figure 1.5, is to detect correspondences between the 3D model and the image in order to align them. We want to explore image processing methods to register a real image with a 3D landscape model. We will discuss three methods based respectively on the skyline, on a Digital Elevation Model (DEM) textured with an orthoimage and finally on a DEM textured with other terrestrial images. Despite their interest for many applications going from augmented reality to the organization of image databases, these processes are currently the focus of few investigations.

The correspondences detected automatically are less accurate than the GCP that an operator would provide. Thus, the camera positions estimated with these correspondences are typically unlikely: located in the air or underground, in a cliff rather than on a path. Therefore, in our last contribution, we will explore how we can estimate the probability of a location to be a good location to shoot pictures given its geographic characteristics.

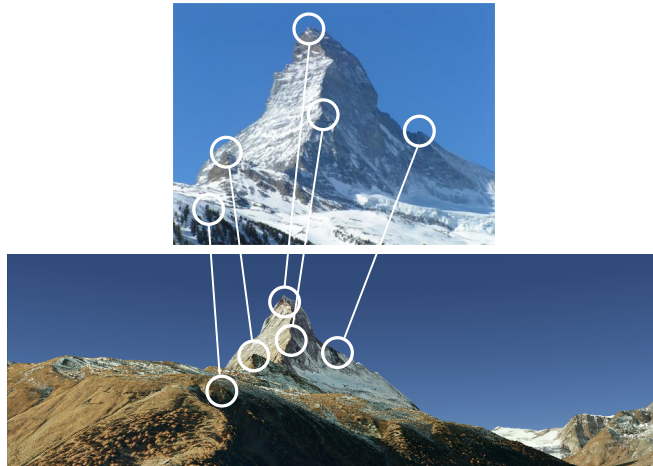


Figure 1.5 – Bottom a synthetic image (*Google Earth*), top a real image. In our approach, we want to find image processing methods able to extract correspondences between these images.

1.3.3 Metadata and case studies

The retrieval of an image orientation within the entire globe is yet utopian. Fortunately, a significant amount of images is associated with geographic attributes, which can restrict the search space to a smaller part of the globe:

- **The position and orientation of the camera are measured:** Recently, mobile phones and tablets have been enabled with GPS, IMU and compass. However, if the position provided by the GPS can be trustful in open landscapes ($<10\text{m}$), usually the azimuth provided is coarse and perturbed by the proximity of metallic objects. Values measured with consumer sensors are neither precise enough to augment pictures with geographic data nor to extract geographic information from the picture. These values have to be refined.
- **Only the position of the camera is known:** If the location of a camera is provided, the camera has three Degree Of Freedom (3DOF): three angles to be recovered. Such images are recorded with GPS-cameras without compass or provided by photo-sharing platforms having geotagging utilities (the user indicates the location of the picture on a map). In databases of more ancient pictures, a position can also have been measured by a surveyor or the position can be deducible from a description (for instance the name of a summit).
- **An approximate location of the camera is provided:** Currently, this is the most common case in databases of pictures. The approximate location is either provided by the user via a photo-sharing platform or found as a toponym in the textual tags. The camera has 6DOF: three coordinates for the location and three angles for the orientation.
- **Only the image is available:** As stated before, unless the user is able to recognize the location, this problem faces a much too large search space and will not be encompassed here.

Usually, image databases consist of a mix of these types of images. The geographic attributes are priors that our methods will take into account to ease the georeferencing by limiting the possible locations and orientations of the picture.

Many authors (Schlieder and Matyas, 2009; Xie and Newsam, 2011) notice that the spatial distribution of images is inhomogeneous. Indeed, unlike photogrammetry campaigns, pictures shot by the general public are biased by the attractiveness of the locations. Hence, easily accessible, scenic areas are subject of many pictures whereas the other regions are neglected. Thus, we can define two types of areas:

- **Popular areas:** In scenic and famous areas, the landscape is densely photographed. Pictures overlap temporally and spatially: they may represent a same region at a same time. The overlap between the images can be exploited for the pose estimation based on the detection and matching of key-points between the images.
- **Less popular area:** In other areas, the only reference available for the georeferencing of a picture is geographic data. Unlike the detection and matching of key-points in similar images, no robust solutions for the matching of an image with geographic data have been proposed yet.

1.4 Objectives

In this section, we will formulate the main objectives of this thesis. If appropriate, we will also announce the associated hypotheses which, in our context, are the drivers of the method chosen to implement the objectives. We also give the conditions of validation of the objectives. Hence, at the end of the manuscript (in Section 7.2, p. 140), we will ensure that the objectives are reached by comparing the conditions of validation and the results of our experiments.

The main objective of this thesis is to provide methods to georeference high-oblique images by comparing them with a landscape model.

1.4.1 Monoplotter

First, we will develop a computer program dedicated to the manual georeferencing of single images (a monoplotter). Thus, the first objectives of this thesis are :

Objective A: Improve existing supervised methods to georeference single (oblique) images (monoplotters):

A. Obj. 1. Facilitate the acquisition of 2D-3D correspondences:

- Provide a 3D browser (to render geographic data with a natural viewpoint);
- Provide GCP digitization in the 3D environment;
- Provide Ground Control Lines as an alternative to GCP;

A. Obj. 2. Provide a direct interaction between a picture and the GIS.

1.4.2 Registration with a 3D landscape model

Second, we want to limit the intervention of the operator and thus detect automatically 2D-3D correspondences (in the image and in the landscape model). The second objective is then:

Objective B: Exploit a landscape model as the reference for the automatic georeferencing of an image having a pose prior:

- B. Obj. 1.** Find metric(s) to measure the similarity of (features of) real and synthetic 3D images;
- B. Obj. 2.** Detect correspondences between a real and a synthetic image;
- B. Obj. 3.** Estimate the pose of a picture from 2D-3D correspondences by taking into account a pose prior;

Hypotheses :

- B. Hyp. 1.** Skyline-matching can provide an estimate of the 3DOF and 6DOF orientation of a camera;
- B. Hyp. 2.** Synthetic views of a DEM textured with an orthoimage are sufficiently similar to natural images for image matching techniques to detect correspondences;
- B. Hyp. 3.** A pose prior limits potential correspondences between the 3D landscape model and the picture (geometric constraint). This constraint improves the image matching.

Validation :

- B. Val. 1.** Skylines extracted from a real image and a landscape model are aligned and correspondences detected. The correspondences are used to retrieve the 3DOF and 6DOF orientation of a database of images having a ground-truth orientation.
- B. Val. 2.** A metric to measure the similarity of (patches of) real and synthetic images is provided. This similarity measure provides 2D-3D correspondences exploited to retrieve the 3DOF and 6DOF orientation of a database of images having a ground-truth orientation.
- B. Val. 3.** The methods are integrated to georeference a collection of various (accuracy of the geotag, appearance) and overlapping images. The quality of the georeferencing is assessed (accuracy of the localisation of a pixel).

1.4.3 Learning priors

For the objectives presented above, we assume a pose prior (provided by an operator or an instrument) limiting the possible locations and orientations of the camera. As a last objective, we want to learn such priors from a collection of images.

Objective C: A method to learn the attractiveness and repulsiveness of a location is provided. It results in a map measuring for each location its likeliness to be a shooting spot.

Hypotheses :

C. Hyp. 1 Geographic indicators describing the accessibility and morphology of a location are related to the location of pictures.

Validation :

C. Val. 1 An independent validation set is used to verify the pertinence of the method, the chosen indicators and the resulting map.

1.5 Organization

This thesis will follow the following structure. First, in Chapter 2, we will do a review of the state-of-the-art solutions for the georeferencing of landscape images. The review has two distinct parts, presenting respectively manual and automatic methods to derive an image orientation.

Next, in Chapter 3, we will define theoretically the main tools which are used in this thesis (and are common) for the georeferencing of pictures. Hence, we will define the camera model, the interior and exterior orientation. Based on a camera model, the camera orientation can be retrieved from 2D-3D correspondences. Several algorithms have been developed for the pose estimation from 2D-3D correspondences and we will describe some of them. In automatic methods, 2D-3D correspondences are detected with image matching algorithms. The main steps of these algorithms are described. Finally, we will also review some methods to insert geometric constraints to improve the feature matching.

The following chapters present our contributions. In Chapter 4, we will describe our monoplottter, which estimates the orientation of a picture from 2D-3D correspondences detected by an operator and relates the picture with geographic data in a GIS. It gives the basis of this work by showing which correspondences are chosen by a user, but also which accuracy of the georeferencing can be reached. In Chapter 5, we will present three methods to register a picture with a landscape model: automatic methods for the detection of the correspondences are divided into two parts, according to the reference data they use: those based strictly on the DEM (Section 5.1) and those based on the DEM textured with an orthoimage (Section 5.2). These two contributions are subdivided into the retrieval of the orientation if the location of the camera is known (3DOF orientation) and the retrieval of the position and orientation if a pose prior is provided (6DOF orientation). Next, in Section 5.3, we will present how we can benefit from the insertion of georeferenced pictures (downloaded from photo-sharing web-sites or georeferenced previously) in the landscape model. This Section shows how key-point detectors and descriptors developed in computer vision can be inserted in our workflow if pictures are overlapping. In our last contribution (Chapter 6), we will show that we can learn trends from existing databases of image locations. Specifically, we will learn which are the preferred spots to shoot pictures.

Finally, we will conclude this thesis in Chapter 7 with a general discussion of our contributions, the achievement of our objectives, and how the methods presented could be further developed.

2 State-of-the-Art

In this chapter, we review the state-of-the-art methods for the georeferencing of oblique images with a focus on landscape images, which are those considered in this thesis. Detailed and more technical reviews are presented in the sections dedicated to our contributions.

Manual georeferencing of landscape pictures is tedious. In general, it is justified for images with a high value, such as historical photographs. Hence, we will start with a brief history of landscape photography and of methods developed to georeference them. In this section, we will especially compare photogrammetric and consumer photographs. In the second section, monophotogrammetry and monoplotters, as well as their applications are presented. Finally, in the last section, we review automatic methods for the detection of the **pose** (orientation parameters: location and rotation of the camera) of a landscape image.

2.1 History of the acquisition of georeferenced landscape images

We can divide the photography into two sub-fields. The first is **photogrammetry**, which is the use of photographs for measurement. A large part of the algorithms and methods used and presented in this thesis come from this field. The second is **photography** in its ordinary meaning: photographs shot for artistic or documentation goals. These images are not thought to be used for measurement, but may contain valuable information, for instance, about our environment. In the two next sections, we will present a historical background of both fields giving emphasis to the orientation of pictures.

2.1.1 Acquisition and georeferencing of photogrammetric pictures

First easily usable cameras were developed by L. Daguerre around 1837-1839 and named daguerreotype. Before that time, some cameras had been developed but can be simply regarded as prototypes considering their cumbersome time of exposure and the poor image quality. Shortly after the commercialization of the daguerreotype, geodesists saw the potential of using photographs for mapping. At this time, mapping was done with plane-tables, illustrated in Figure 2.1 (a), a horizontal table localized either by graphical intersection of lines drawn in the direction of objects with known coordinates or determined mathematically from angular measures to these objects, namely *resection*. Hence, if a same object is observed from several plane-table positions, it can be mapped, as illustrated in Figure 2.1 (b). First cameras used for mapping, the "photogrammetric plane-tables", were used in a similar way: multiple

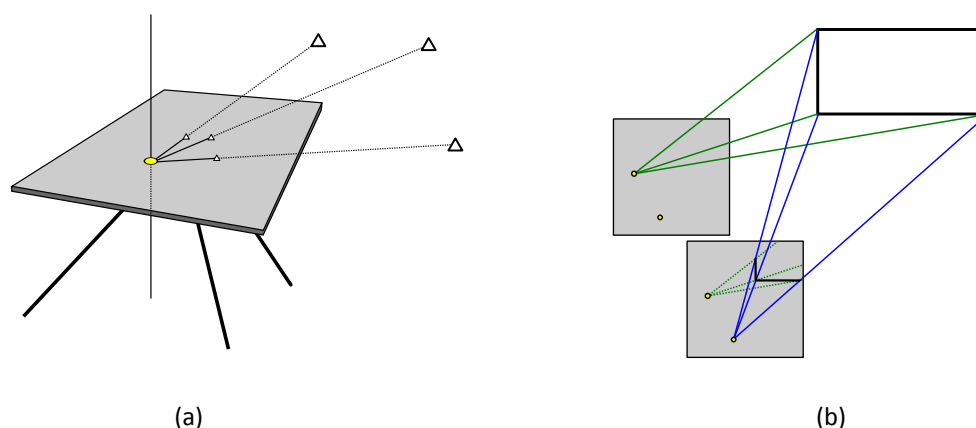


Figure 2.1 – (a) Three known points are plotted on the map and used to determine the position of the plane table (resection). (b) A similar object is observed from two locations. The intersection of the lines of sight determines the location on the map.

oriented photographs of an object were shot from several stations localized by resection. The map was then created from the intersection of the bearings. This method supposes a horizontal camera which is localized on the horizontal plane. Shortly after, it was discovered that two images, distanced with a short basis, can be used to determine the depth of the scene with a stereoscope. Phototheodolites were developed for this particular task.

In early 1900s, photogrammetry really took off with the first airplanes. First campaigns and results were published in 1913 (Tardivo, 1913) and this technique was directly used during WWI to detect the location of the enemy lines. However, at that time, there were no instruments to measure directly the camera location and orientation in the air. These parameters needed to be computed indirectly from corresponding locations in the image space and object space, namely 2D-3D correspondences (Gruen and Huang, 2001). Indeed, the resection from three points used to compute the planar location of a photogrammetric plan-table cannot provide the 3D orientation of a camera. Once in the air, the camera can turn around two more axes and its height must also be determined. Church (1936) developed an analytical solution to this problem called, in imitation, *space resection* (see Section 3.3).

Later, camera and planes evolved and allowed photogrammetrists to shoot sets of overlapping, close to horizontal images. For these particular sets of images, a new georeferencing approach was set up. First, a relative orientation was done from 2D-2D correspondences (tie points) found between the images. Second, the absolute orientation was achieved with 3D [image] - 3D [object] correspondences (control points) (Gruen and Huang, 2001). Currently, an accurate pose of the aerial cameras is usually computed with a Differential GPS and an IMU. These values are used to ease the detection of tie points and as prior for the absolute orientation.

Lately, computer scientists and photogrammetrists noticed that they had developed similar methods to solve similar tasks (Förstner, 2009), but that computer scientists focused on broader type of images, cameras and applications, included uncalibrated cameras from the consumer market. One of the great contributions of the computer vision is the development of robust key-point detectors and descriptors. Key-point detection and matching completely replaces operators for the detection of tie points and makes the relative orientation fully automatic.

2.1.2 Georeferencing general public photography

In parallel to photogrammetry, pictures were also shot by photographers for documentation and artistic goals. With the rise of films, replacing the inconvenient plates, cameras appeared in the customer market at the beginning of the 20th century. Pictures shot with these cameras were generally not geolocalized at all. However, professional photographers or well-organized amateurs recorded a coarse picture location and date in field notes. Hence, some images may be linked to a date and a toponym, which could be written on the back of the picture or in a tidy form. Some years later, colored films and shortly afterwards, digital cameras appeared and revolutionized the customer market, but did not directly influence the georeferencing. However, it should be noted that digital cameras do not only record the image, but also associate it with some standardized metadata (camera model, date, focal length etc.). Nevertheless, we had to wait the opening of the GPS to civilian use (year 2000) and the miniaturization of the GPS sensors, progressing with the advent of smartphones, to record the GPS location in the picture metadata.

The World Wide Web had a major impact on the way pictures are stored and shared. Photo-sharing websites developed structured databases enabling quick access to the pictures and efficient selection from textual or geographic tags. Hence, most photo sharing websites allow users to provide the picture description and location. The geotagging, which relates a picture with a location, was firstly done manually with a click on a map. The camera location is yet automatically extracted from the pictures shot with cameras equipped with a GPS. The combination of the WWW and smartphones, having a direct connection with the internet, is increasing drastically the amount of images uploaded on the Web and accurately located with GPS. Computer vision scientists took the opportunities of these large sets of shared pictures for several research purposes, among them the localization of un-geotagged pictures (Crandall et al., 2009) or the 3D reconstruction of famous monuments (Snavely et al., 2006; Strecha et al., 2010; Agarwal et al., 2011) (see Section 2.3).

Landscape photographs are the subject of this thesis. These images are mainly images shot with consumer cameras. Several type of georeferencing can be found in the databases: accurate locations measured with GPS, manual geotags, toponyms as a location indicator or no geographic information at all. These images require a particular procedure to compute their pose and consequently relate them with geographic data. In the next section, we will review supervised methods developed to retrieve the location and orientation of a photograph.

2.2 User interaction-based pose estimation

Images not associated with their orientation require a specific processing to extract geographic information from their content. GIS provide georeferencing functions, not adapted to oblique images. Indeed, they generally compute a planar transformation between the image plane and the map plane without considering the relief and the perspective. For the collection of images discussed earlier, and specifically for terrestrial and high-oblique images, the perspective and the relief generate substantial deformations, which cannot be retrieved from 2D-2D transformations.

The literature reviewing the localization of terrestrial landscape pictures is mainly dedicated to historical pictures. However, we can also find more and more studies using fixed cameras to sense continuously a landscape and measure the evolution of an environmental parameter. Indeed, these images have a high value (historical or scientific), which justifies the time spent by an operator for their georeferencing. Two trends are discussed in the next sections. The first is repeated-photography

which requires an operator in the field to detect visually the location similar to the picture and shoots the same picture. The second is monophotogrammetry using 2D-3D correspondences detected by an operator to retrieve the camera orientation.

2.2.1 Rephotography, repeated-photography

Rephotography or repeated-photography is the search of the location of an existing photograph in order to shoot a new similar image. The image pairs can then be easily compared. Rephotography is often used as a visual support to show landscape change to the general public. Despite its numerous drawbacks, it is also used by scientists. A review is presented below.

Repeated photography is a time-consuming way of computing the georeference of a picture which is surprisingly widespread. The first known scientific who used rephotography was Finsterwalder in 1888 (Webb, 2010). Finsterwalder was an active photogrammetrist who not only repeated his surveys of glaciers in the Alps, but also provided the basis of the space resection. Several studies, and even recent ones, apply this setting in natural sciences. Webb (2010) lists in his book a wide range of scientific applications of repeated photography (geoscience, population ecology, ecosystem change, cultural application). The main advantage of the repeated photography is that it is a straightforward way to demonstrate landscape changes to the public. The presentation of successive images with a natural perspective allows everyone to evaluate the changes instinctively. Hence, there are innumerable exhibitions, websites and photography books illustrated with pictures acquired using this technique.

Often, scientists using rephotography make use of GIS data and Virtual Globes to detect an unknown camera orientation and refine it once in the field. Bae et al. (2010) published an image matching-based method for the refinement of the camera orientation in the field. This solution is based on a 3D reconstruction from two images and a guidance of the photographer. This method supposes that the query image and the reference images, shot in the field, are sufficiently similar to be matched with a key-point detector and descriptor (SIFT, see Section 3.4). However, among the studies using rephotography, few exploits a DEM to extract georeferenced data. For example, in the reference book for rephotography one can read: *"While repeat photography cannot provide the same type of quantitative data as can be obtained from satellite imagery and aerial photographs, it serves to complement them providing (...) a greater time range, in many cases going back to the nineteenth century"* (Webb, 2010). This statement is only partially true, because, as we will see in the next section, if a DEM or a Digital Surface Model (DSM) of the area is available and if the camera is oriented with GCP, georeferenced data can be extracted from a photograph. It is true that data extracted this way are usually not as accurate as those extracted from aerial images georeferenced with standard techniques.

At the time of precise GPS localization and accurate DEM, repeated-photography seems nicely outdated, but has the advantage to show that some technologies common in a field (2D-3D pose estimation in photogrammetry and computer vision) may have difficulties to reach applicative studies. This is probably due to a lack of easily available and user-friendly software providing the connection between a picture and the GIS. This software is commonly called monoplottter and is discussed in the next section.

2.2.2 Monophotogrammetry

The monoplottting functions use the orientation of a picture to project each pixel of an image on a DEM. In contrast, stereoplottting uses two photographs to determine the depth and thus does not rely on a DEM. As stated in Kraus (2007), monoplottting is generally not as accurate as stereoplottting but has some advantages.

- First, monoplottting can be used without expensive stereoscopic workstations: using monoplottting an operator can measure and digitize objects directly from one single image.
- If the orientation of the image is not computed with aero-triangulation, it can be computed from 2D-3D correspondences - called Ground Control Points (GCP). In this case, we will call the software having the functions for the pose estimation and the monoplottting a **monoplotter**. Hence, the second advantage is that monoplotters can be used to extract georeferenced information from images that do not comply with the requirements of the aero-triangulation framework (images not shot for photogrammetric purpose or image collections with little or no overlap).
- Finally, the orientation from GCP does not require particular skills or knowledge, which can make the monoplottting useful for everyone aware of GIS and more accessible than stereo-photogrammetry.

Currently, there are three providers of monoplotters. Bozzini et al. (2012) develop a standalone monoplotter used for several researches (Bozzini et al., 2013; Scapozza et al., 2014; Wiesmann et al., 2012) and benefit from a large community of scientific users who guide the development of the required functions. This program is mostly used for the analysis of historical photographs. Lately, Messerli and Grinsted (2014) has provided a Matlab toolbox for the monophotogrammetry. This toolbox is mainly dedicated to environmental studies from successive images. Finally, a pose estimation function is found in a GIS environment (ILWIS GIS, 2012, 52° North ILWIS Community, ilwis.org 🌐), but this GIS is poorly represented in the community.

Monophotogrammetry is much more accessible than stereo-photogrammetry and everyone familiar with GIS should be able to compute a camera pose easily. For instance, Google Earth (*Google Earth*, 2015, *Google Inc.*, *Mountain View*) provides a function to browse and insert an image in a virtual globe. However, none of the existing software mixes the conviviality of the virtual globe and the accuracy of the GCP. Currently, most of the users of monoplotters are scientists, which have to find a way to extract quantitative data from landscape images. Scientists who are at ease with programming can use the Matlab toolboxes developed by Messerli and Grinsted (2014). However, with a more accessible monoplotter (more user-friendly, free and part of a GIS software), the monoplottting could be used more extensively by the GIS community and even bring new people into the GIS community (archive managers...).

2.3 Automatic pose estimation

Monophotogrammetry requires a large investment of the operator for the GCP digitization. Recently, georeferencing methods that do not involve the user have been developed. They exploit the availability of images georeferenced with GPS and inertial sensors, pictures geotagged by the users, accurate GIS data and recent development in computer vision. In the following paragraphs, we will first review methods using databases of georeferenced images as reference for the world-wide localization of query images. Second, we discuss more accurate georeferencing methods based on the 3D reconstruction

from a collection of images. Third, we provide some examples of methods involving a database of geographic objects as the reference.

2.3.1 Coarse localization of pictures

Among the billions of images shared on the web, only a small (but increasing) amount of images is georeferenced. Some researchers aim at computing the location of a picture by comparing it with databases of georeferenced images. For instance, this task is proposed to the researchers of the multimedia community (multimediaeval.org 🐼) with benchmark datasets. For a query image, the general idea is to determine the most similar geotagged image or the most similar cluster of images. Hays and Efros (2008) described the set of geotagged reference images with popular features descriptors from the literature. An easy way to improve this result is to consider also textual descriptors. Indeed, most of the images found on the web are also associated with some textual data (namely tags). Hence, Crandall et al. (2009) extract spatial clusters of pictures by computing a spatial density. Next, they train a classifier based on textual and visual features for each cluster. Hence, a picture and its tags can be associated with the most similar cluster which is an indication of its location. The discontinuous and heterogeneous spatial distribution of the images, driven by the presence of point of interest, is the main drawback of the databases of geolocated images. Schlieder and Matyas (2009) show that the distribution follows a power law: few locations are very often taken in photo while most of the other locations are subject of few or no shots. The difficulty is then to extend the georeferencing methods outside these dense areas. Lee et al. (2015) propose a method to derive "geo-informative" attributes (such as the elevation gradient, population density, percentage of pasture land) from an image. The recognition of these attributes can reduce the locations likely to be represented in a picture. Tsung-Yi et al. (2013) pursue the same objective, but rather relate visual features of the query images with visual features of an orthoimage and a land cover map. The relations between the image and map features are learned from a training set of georeferenced images but provide an estimate of the picture location everywhere on the map.

These approaches do not compute a location precise enough to meet our objectives. However, they can be computed world-wide or at least at large scale. They show that the available databases of geolocated images can be used to extract statistical information about the geolocation of pictures. Thereby, we will propose a method to learn the type of locations preferred to shoot pictures in Chapter 6. By involving geographic data, our method is not biased by the spatial distribution of the reference images and gives an estimate at local scale.

2.3.2 SfM-based localization

In densely photographed areas, the literature related to the pose estimation of images is currently widely dominated by Structure-from-Motion (SfM) methods. SfM is the name used in the computer vision community for the simultaneous orientation (inner and outer) and 3D reconstruction from overlapping images. A usual SfM follows these steps (Remondino et al., 2012):

1. Key-point detection (SIFT, SURF, etc. see Section 3.4.1.2),
2. Key-point matching (see Section 3.4.3),
3. Outlier rejection (RANSAC, see Section 3.5.1 and relative orientation of two or three images (Hartley and Zisserman, 2003)),

4. Relative orientation of the block of images,
5. Bundle adjustment (least-square based refinement of the orientation).

For some applications, the absolute orientation (georeferencing) is not always required. The georeferencing can be obtained by providing GCP to the bundle adjustment (the most rigorous approach which avoids drifts) or by georeferencing the final product of the SfM with a similarity transform (for instance with GPS locations of the images or GCP). During the three first steps, tie-points linking overlapping images are detected and used to simultaneously reconstruct the camera orientations and the 3D location of the tie-points. Hence, SfM requires a set of images overlapping widely and covering completely the scene.

This process has reached the consumer market and is found in desktop or web-applications to process aerial and terrestrial images, for professional and amateur needs. In the literature, it has proved its efficiency for the reconstruction of scenic or historical locations from sets of images uploaded by tourists (Agarwal et al., 2011; Snavely et al., 2006). However, general methods based on SfM have four limitations:

- First, they require a complete and overlapping set of images. These sets are typically found only around points of interest such as tourist attractions or obtained during a photogrammetric campaign.
- Second, if the image distribution is heterogeneous and the GCP are not inserted in the resolution of the SfM, drifts occur and generate distorted 3D models.
- Third, if the images forming a set are not sufficiently similar (illumination, quality, viewpoints), several unlinked blocks, corresponding to homogeneous subsets, are reconstructed.
- Finally, SfM only partially answers the problem of growing databases. For a new image to be georeferenced, a new SfM has to be run.

For the reconstruction of a city model from web-shared images, Strecha et al. (2010) propose a method to overcome some of these limitations. They involve geotags and GIS data to compute the camera orientation directly in the geographic coordinate system and thus avoid unlinked clusters of images. They first use geotagged images as a loose constraint for the camera orientation. These orientations are then refined further by forcing each 3D reconstruction to lie on the ground and match a database of building footprints. Moreover, the overall process is based on cluster growing. A new image inserted in the database is associated with a cluster and hence only a subset of the database needs to be processed to orient the new image.

Several authors propose to localize directly a query image by matching it with the 3D key-points generated from the SfM. In these approaches, there is a huge amount of 3D key-points that must be stored and compared with the key-points of the query image. Hence, robust and efficient matching methods have to be developed (Sattler et al., 2011; Svärm et al., 2014). Li et al. (2012) recover the pose of a single picture at "Worldwide Scale" by matching the key-points detected in a query image with those found in several reconstructed models. The models are undoubtedly distributed all around the world, but the available reconstructed scenes are only parts of cities or famous rural areas.

Hence, the main limitations of the SfM-based methods are that they are applicable only to densely photographed and reconstructed locations. Specifically, to start the orientation of a cluster of images they need a sufficient amount of similar and overlapping images. This conclusion stresses the need

for other methods using as reference geographic data, and thus available everywhere or in regions where the images overlap and similarity is more heterogeneous. In particular, geographic data is an opportunity to register images also away from tourist attractions and to develop matching strategies less sensitive to various image appearances.

2.3.3 GIS data as reference

Orthoimages and query images are of a same nature (typically color images), but are issued from very different viewing angles inducing substantial deformations. Their direct matching is then challenging. Thus, Li et al. (2014) ask the user to draw straight and horizontal lines in the query image to match those recovered in orthoimages of urban areas. Cham et al. (2010) propose an inverted process, the query images are matched with a database of building footprints.

Recently, oblique aerial views, much more similar to terrestrial views, have also started to be distributed. Bansal et al. (2012); Schindler et al. (2008) use respectively features descriptors developed for façades, and façade recognition to match a query image with a 3D database of reference façades. Currently, terrestrial views are also acquired systematically to provide street-view maps or 3D city models. Hence, some cities and some famous buildings are modelled with detailed and textured 3D models. Aubry et al. (2014) create a database of 3D views generated from several locations and orientations distributed around the model of an architectural site. In these reference images, they determine patches likely to be matched with patches extracted from query drawings, paintings or ancient pictures. A patch of the query image matched with a patch of the database provides a 2D-3D correspondence used to compute the picture pose. To overcome the differences induced by the different types of images, the patches are matched with a classifier.

Currently, numerous cameras are equipped with a GPS and compass and record a pose prior. With a prior, the matching of a picture with a GIS model is simplified. Some authors match an image with 3D models of buildings (Haala and Böhm, 2003; Sourimant et al., 2012). Other authors apply a texture on the city model to render synthetic views more similar to the real pictures (Coors et al., 2000; Reitmayr and Drummond, 2006; Dawood et al., 2011), this process is also called *analysis-by-synthesis* in the literature (Schumann et al., 2009; Koeser et al., 2007).

Lately, object recognition has also improved considerably. Lee et al. (2015); Ardeshir et al. (2014) propose to detect objects such as road signs in the query image and find the corresponding records in a GIS database. However, most of the objects stored in geographic databases have more various shapes than road signs and are more difficult to recognize. Thus, Hammoud et al. (2013); Koperski et al. (2013) ask the user to label objects in the picture and match them with a fusion of GIS data (DEM, land use, building height etc.).

By necessity, most of these examples are associated with urban datasets. Indeed, in cities, GIS models and reference data are more detailed. Moreover, a large part of these methods involve directly or indirectly straight lines and right angles which are predominant and easily detected in images and models of buildings. Finally, urban areas have also more commercial implications (augmented reality, GIS model update, vehicles navigation with deteriorated or loss of GPS signal). The most inspiring work in this review is proposed by Aubry et al. (2014). The matching of pictures of various type with a synthetic model is very close to our objectives. In the next section, we will focus on the georeferencing of landscape images representing rural areas, which is specifically the target of this thesis.

2.4 Landscape image registration

Rural pictures, just like urban pictures, can be separated into images with or without location priors. We will first review methods to recover the orientation of geolocalized pictures (3DOF orientation). We will then take an interest in methods dedicated to images without location priors (6DOF orientation).

2.4.1 3DOF Orientation

In particular databases or case studies, images are associated with their location and (sometimes) an estimate of their orientation. For instance, images can be shot from a particular summit or recorded location. Moreover, more and more cameras are equipped with GPS capable of measuring an accurate location ($<10\text{m}$ in open landscapes). For these specific images, if the focal length is known or approximated, only three camera angles have to be recovered.

In hilly regions, horizon and silhouette edges are widely used for this task as they are hardly affected by seasonal and illumination changes. For instance, Behringer (1999) suggests using the skyline to retrieve the tilt and azimuth, while the roll is assumed to be null. In this method, the horizon line is extracted from the query image with edge detection, and the reference horizon is computed with a DEM. Peaks and dips are then extracted from these sequences. Several matches of these clues are tested and the one which minimizes the distances between the reference and image skylines defines the camera orientation.

Baboud et al. (2011), for the specific task of naming mountains in a picture, also assume known camera Field of View and position. They propose to work not only with the skyline but with every morphological break-lines (depth discontinuities). On one hand, they generate a 360° panorama around the camera location and on the other hand, they apply edge detection (compass operator, Ruzon and Tomasi (2001)) to the query image. The registration is thus twofold. First, a cross-correlation algorithm, which gives emphasis to the edge orientation, is used to detect potential solutions. Second, a particular metric is developed to select the best solution. Chippendale et al. (2008) have a very similar approach for a similar goal. Their registration algorithm is based on key-points detection both in a rendered view of the DEM (emphasizing the morphological break lines) and on the edges of the query image. Similarly to RANSAC (see Section 3.5.1), pose hypotheses are evaluated with an edge-features comparison. Finally, the key-points of the best solution are used to refine the pose.

A different approach is that of Baatz et al. (2012b). Here, the relief is involved indirectly. The 360° panorama, generated with the 3D model, is rendered with land-cover classes (lakes, forest, grass and built-up areas). The same classes are detected in the query image based on a classifier trained with a set of pictures. Thus, the orientation is reduced to a cross-correlation of the classified image with the land-cover panorama.

2.4.2 6DOF orientation

Unfortunately, many pictures are not associated with a precise location. The literature covering the 6DOF pose estimation of landscape images is much more disparate than the one dedicated to urban areas. Existing methods are generally based on SfM and were discussed earlier. An interesting exception is the work of Baatz et al. (2012a). Coming back to the skyline, they extract "contourlets" of the horizon in a query image and in panoramas generated on a regular grid over the entire surface of Switzerland ($40'000\text{km}^2$). Then, using a geometric constraint to add robustness, they match the skyline features of

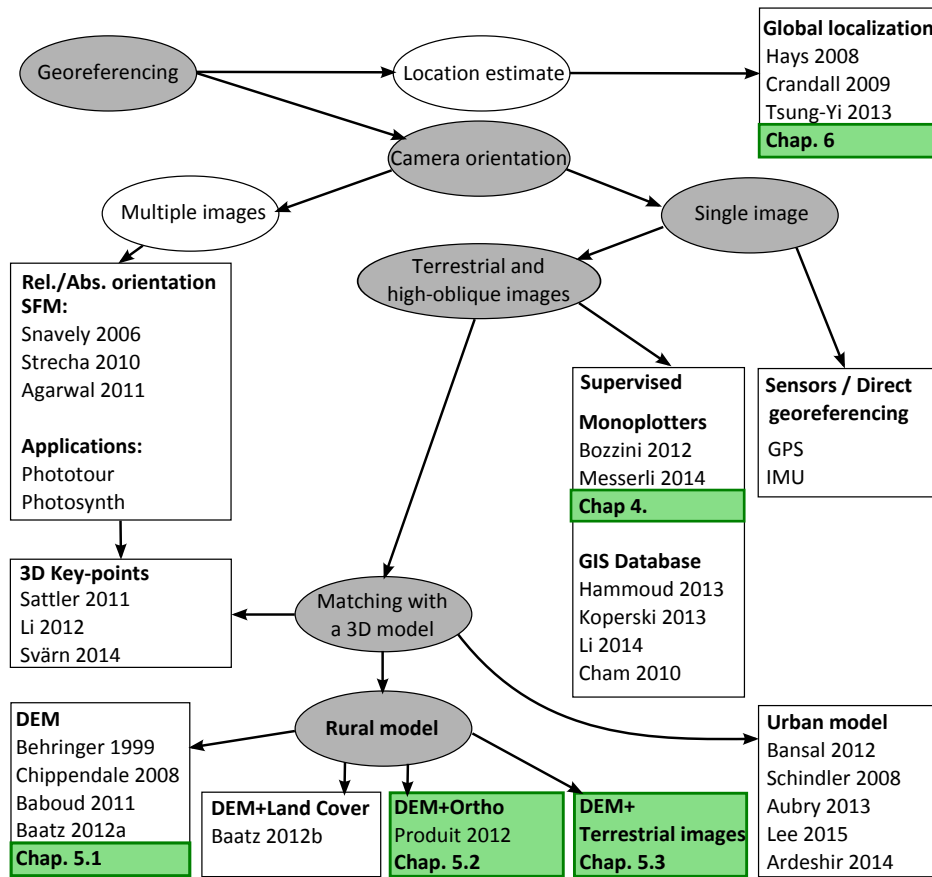


Figure 2.2 – Summary of the state of the art. Our contributions are highlighted with a green frame.

the query images with the reference database. Assuming that a location is correctly estimated if it falls in a 1km radius around the ground truth location, they reach an accuracy of 88%. As far as we know, this is the only method dedicated to 6DOF orientation of landscape images which is not based on SfM or the existence of a database of 3D key-points generated by SfM. This method is a good indicator of *a priori* pose but is not very accurate.

2.5 Discussion

To summarize this Chapter, a scheme showing the fields of study discussed previously is presented in Figure 2.2. As illustrated, there are two kinds of georeferencing. The green frames indicate our contributions which are detailed in the next section.

First, there are applications that aim at recovering an approximate localization of a picture (an estimate of the 2D location). Based on a training database of geotagged images, these authors learn spatially discriminative features (visual and/or textual). These estimates are often biased by the inhomogeneous spatial distribution of the training images (clustered around points of interest).

Second, there are studies which recover the camera orientation. Here, we enter the field of photogrammetry (or photogrammetric computer vision). There are two main trends. On the one hand,

there are methods reconstructing the 3D coordinates of key-points from a collection of pictures (in a relative or absolute coordinate system). They can thus be applied only in regions densely covered by pictures. They can georeference only clusters of images sufficiently similar (viewpoint, appearance). On the other hand, there are methods recovering the orientation of a single picture by relating it with georeferenced data. This is the direction that we chose for this thesis since it can potentially answer the georeferencing everywhere and be less sensitive to various landscape appearance. We identified then two fields which could benefit from the knowledge coming from the GIS community. In the supervised methods, we can find two families of solutions. Using a monoplottter, the operator has to provide corresponding locations (GCP) in the picture and in the georeferenced data. This task is laborious because the user must show abstraction skills to match an orthogonal top view with the oblique viewpoint of a photograph. Studies going a step further decrease the user involvement. The user is only asked to detect high-level objects in the query images (such as buildings and associated number of floors, roads, lakes etc.). The spatial relations between these objects are then compared and matched with a fusion of GIS databases to provide an estimate of the camera pose.

In the unsupervised methods, the goal is to completely replace the operator for the detection of the correspondences. We divide here the contributions according to the type of 3D model they consider. First, some authors match a picture with a database of 3D key-points computed with SfM. These methods suffer from the same limitations than SfM. Second, other authors are specifically focusing on urban environments and thus rely on detailed models and remarkable features such as façades, straight lines etc. specific to this environment. Finally, few studies focus on rural images. Here also, we subdivided the contributions according to the type of reference data: a DEM only, a DEM textured with land cover classes, a DEM rendered with an orthoimage, and finally a DEM associated with other terrestrial images (which can be images downloaded from the internet or images already georeferenced belonging to the same collection). In these two last fields, to our best knowledge, we provide currently the only contributions.

2.6 Contributions

Figure 2.3 presents in more details our contributions. On its top, we can see that the pose of an image can be separated into its location and orientation parameters. Accordingly, our contributions (encircled and referencing on the corresponding chapter) can act on the location **and/or** on the orientation of the image. On its bottom, the geographic data available as the reference are represented. We will not work directly with the geographic data, rather with rendered 3D views.

The first contribution (in black) is the development of a monoplottter integrated with a GIS which enables the user with 3D navigation in a virtual globe to ease the GCP acquisition. The second (purple) is a novel method to match the skyline detected in a query image with the skyline generated from a 3D model. It is especially successful to retrieve the orientation of an image with a given localization. However, we will also discuss under which conditions it can be used to compute the 6DOF orientation of an image. The third (yellow) contribution covers the same case study, but uses a descriptor to match query image patches with a realistic synthetic image. Its extension (dashed yellow) to pictures which have only a prior of the location (and not an exact location) are discussed in the same chapter. The fourth contribution (green) considers existing overlapping images, already georeferenced, to infer the georeference of query images with approximate orientation and localization. Finally, the last contribution (red), acts only on the localization. For each location of a map, we will compute its likelihood to be a picture location. This value is learned from a database of real picture locations

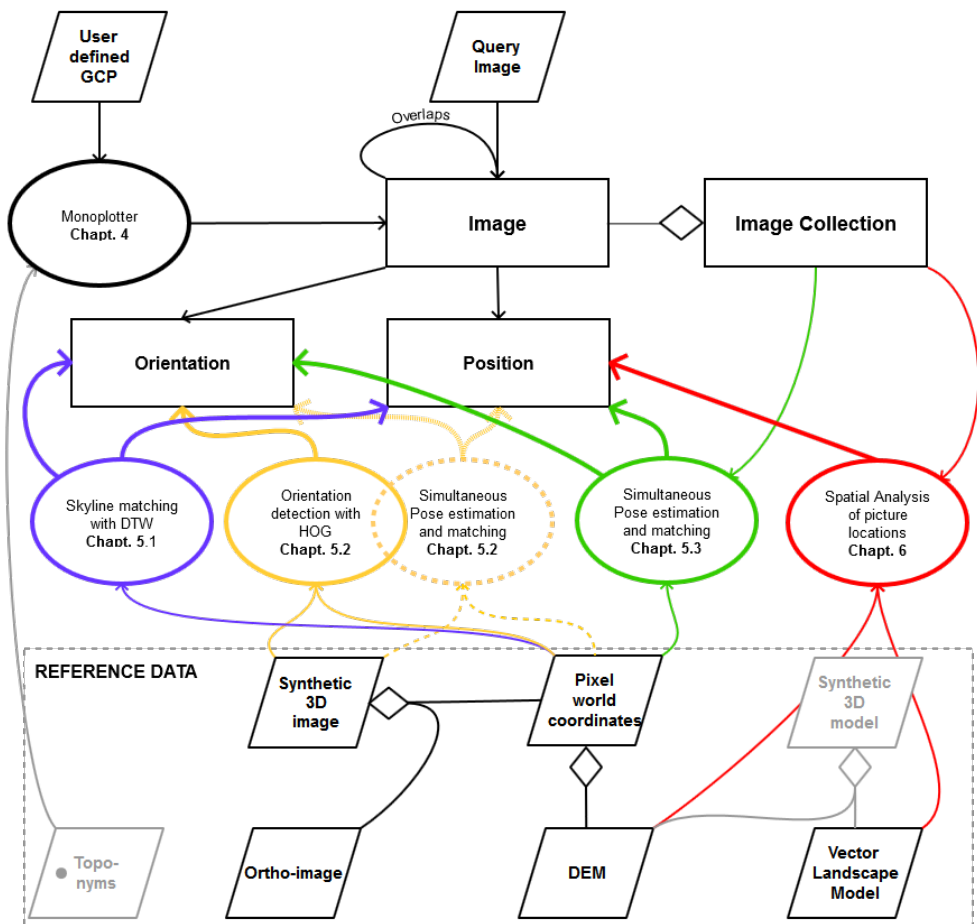


Figure 2.3 – Diagram of our contributions. They can act on the position or orientation of the camera or on both of them. The dashed line encloses available reference data, in gray the data which are not used in this work.

downloaded from a photo-sharing website.

Our contributions are based on techniques and models which are widely used in photogrammetry and computer vision. In the next Chapter, we will give brief introductions to these methods.

3 Theory

Most pose estimation algorithms rely on similar methods. In this chapter, we will present these methods with an emphasis on those used in this thesis. Thus, we will explain how the geometry of a picture is modelled (Section 3.1). Based on this model, we can relate each pixel of a picture with a corresponding object having real world coordinates. To do so, the picture must be oriented according to the world coordinate system (Section 3.2). The camera orientation can be retrieved with Ground Control Points, corresponding locations in the image (2D) and in the world coordinate system (3D). Namely, we will discuss some existing algorithms for **2D-3D pose estimation** or **space resection** (Section 3.3). The detection and matching of key-points across images is currently a central part of many photogrammetry and computer vision applications. We will give a general discussion of its implementation (Section 3.4). The detection of correspondences with these means is generally imperfect, some of the correspondences are wrong and may perturb the resulting registration. Hence, finally, we will present how noisy correspondences can be discarded to estimate an improved pose of a camera (Section 3.5).

3.1 Camera model

The forerunners of the cameras were black boxes drilled with a small hole. Light coming from the real world enters the box passing through the hole and is projected on the back wall of the box. The projected image is then inverted upside-down and mirrored on the horizontal direction. Using the example of this box, we can define a very simple camera model, called a pinhole camera model, illustrated in Figure 3.1. In this model, the hole is called the **projection center**, its projection on the wall is the **principal point**. The distance between them is the **focal length** f . If the focal length is increased (or decreased), the image and the **field-of-view** are scaled accordingly.

The box defines the 3D camera coordinate system where the projection center is the origin, z is along the viewing direction (optical axis) and x and the y are parallel to the sides of the front wall. Finally, the image has also a 2D coordinate system centred on the principal point (u_0, v_0) and u and v parallel to the back wall. The relation between an object located in $\mathbf{x} = [x, y, z]$ and its projection

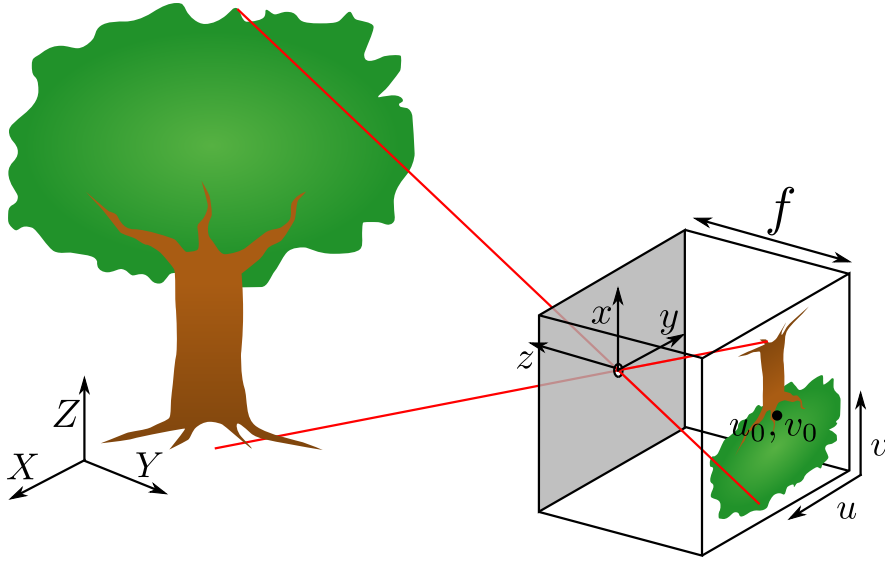


Figure 3.1 – Pinhole camera model and relation between the world coordinate system (XYZ), the camera coordinate system (xyz where z points in the viewing direction) and the image coordinate system (uv). (CC, Wikimedia, Author: Bob Mellish)

$\mathbf{u} = [u, v]$ is:

$$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} = -f \begin{bmatrix} x/z \\ y/z \end{bmatrix} \quad (3.1)$$

Generally, the image coordinate origin is not centred in the middle of the image, rather in the upper left corner. If the central point is located at the center of the image, H is the height of the image and W its width, the central point has the coordinates $u_0 = W/2$ and $v_0 = H/2$ and the relation becomes:

$$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} = -f \begin{bmatrix} x/z \\ y/z \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \quad (3.2)$$

However, this model does not describe exactly real cameras. First, lenses replace the hole and may cause distortions, especially in the borders of the image. Moreover, the photo-sensitive surface (plate, film or sensor) is usually not perfectly parallel to the lens. To respond the needs of accuracy required by photogrammetrists and computer scientists, more complex camera models have been developed. They take into account the lens distortions, the shift between the image center and the principal point etc. These parameters are generally computed during a **camera calibration**.

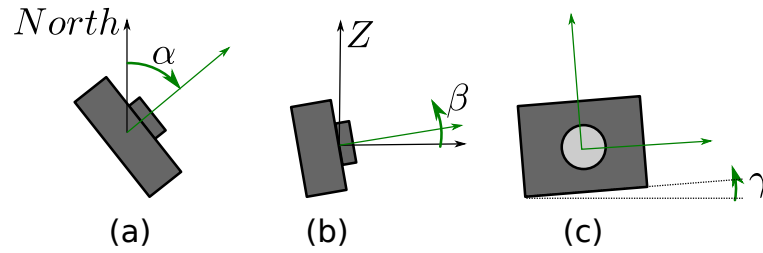


Figure 3.2 – Illustration of the heading (a) top view, tilt (b) side view and roll (c) front view. Green arrows illustrate the camera system, black arrows the world system.

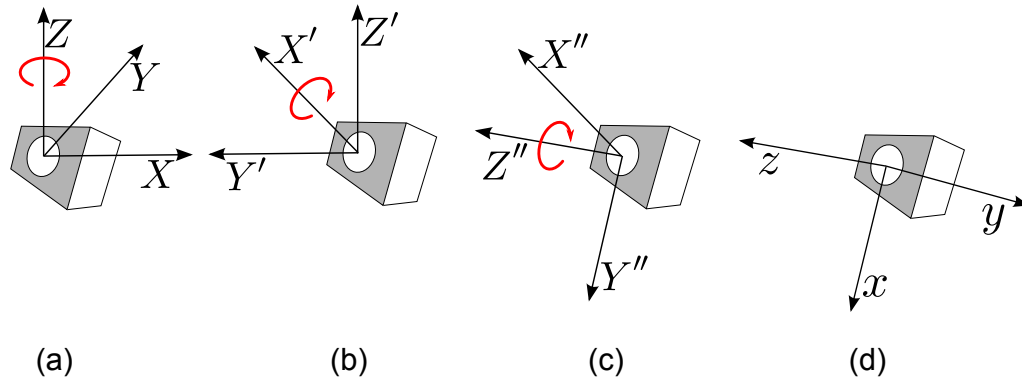


Figure 3.3 – Rotation around Z , X' and Z'' to align the axes of the world system with the camera system.

3.2 Camera orientation

In the previous section, the basic geometry of a camera was described. To relate each image pixel with their corresponding world coordinates $\mathbf{X} = [X, Y, Z]$, as illustrated in Figure 3.1, the transformation from the world coordinate system \mathbf{X} into camera coordinate system \mathbf{x} has to be described. Next, as discussed above, the perspective can be applied to \mathbf{x} to compute the image coordinates \mathbf{u} .

First, we need to be able to describe the position of real world objects in an Euclidean 3D space. Projected coordinates systems are approximations of the curved Earth surface with a plane. This plane has usually its axes parallel to the North and East directions and the Z pointing upwards. We can then describe the camera location, or namely the projection center, by its northing, easting and altitude $(N/X, E/Y, Z)$.

At this stage, the location of the photographer is described but the direction where the camera is pointing, as well as the camera roll, are still unknown. To describe the camera orientation, three Euler angles can be used. These angles are illustrated in Figure 3.2. The azimuth (a) (or heading) represents the angle between the North and the viewing direction. The tilt (b) is how the viewing direction differs from the horizontal plane, it is negative if the camera is pointing downward and positive if it is pointing upward. Finally, the roll (c) (or swing) describes a rotation of the camera around the viewing direction (pointing out of the drawing in the Figure). Within this framework, the sequences of rotations to transform the world coordinate system into the camera coordinate system are $Z - X' - Z''$. For example, in Figure 3.3, the heading is applied around the Z axis (a), the rotation around X' fixes the tilt (b), and finally a rotation around Z'' again sets the roll (c).

Chapter 3. Theory

After these rotations (d), Z is parallel to the viewing direction z and the X and Y are parallel to the image sides (x, y) and thus aligned with the camera system. These rotations can be combined in a unique matrix \mathbf{R} using well-known rotation matrix:

$$\mathbf{R} = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} & 0 \\ r_{2,1} & r_{2,2} & r_{2,3} & 0 \\ r_{3,1} & r_{3,2} & r_{3,3} & 0 \end{bmatrix} = \mathbf{R}_Z(\gamma)\mathbf{R}_X(\beta)\mathbf{R}_Z(\alpha) \quad (3.3)$$

Next, if the camera is located in $\mathbf{X}_C = [X_C, Y_C, Z_C]$, the transformation from the world coordinate system into the camera coordinate system is a 3D rigid body motion (or 3D Euclidean transformation) where \mathbf{R} is the rotation and \mathbf{X}_C the translation:

$$\mathbf{x} = \mathbf{R}(\mathbf{X} - \mathbf{X}_C) \quad (3.4)$$

Hence x, y, z are functions of the Euler angles α, β, γ and the translation parameters X_C, Y_C, Z_C . Now, we can combine this transformation with the camera model defined in Equation 3.2 to obtain the collinearity equations used to compute the image coordinates of 3D objects:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_0 - f \frac{r_{1,1}(X-X_C)+r_{1,2}(Y-Y_C)+r_{1,3}(Z-Z_C)}{r_{3,1}(X-X_C)+r_{3,2}(Y-Y_C)+r_{3,3}(Z-Z_C)} \\ v_0 - f \frac{r_{2,1}(X-X_C)+r_{2,2}(Y-Y_C)+r_{2,3}(Z-Z_C)}{r_{3,1}(X-X_C)+r_{3,2}(Y-Y_C)+r_{3,3}(Z-Z_C)} \end{bmatrix} \quad (3.5)$$

For brevity, we define $\mathbf{p} = [\alpha, \beta, \gamma, X_C, Y_C, Z_C]$ as a vector storing the unknowns of the pose and we can write the collinearity equation as:

$$\mathbf{u} = \mathbf{F}(\mathbf{p}, \mathbf{X}) = \begin{bmatrix} F_0(\mathbf{p}, \mathbf{X}) \\ F_1(\mathbf{p}, \mathbf{X}) \end{bmatrix} \quad (3.6)$$

3.2.1 Homogeneous representation

The collinearity equation are often expressed with homogeneous coordinates where a n -dimensional vector is expressed with an extra component.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ w \end{bmatrix} \quad (3.7)$$

and $x = x'/w, y = y'/w$. This trick is convenient because the coordinate transformation and the perspective transformation can be expressed with matrix multiplication (Mikhail et al., 2001). The rigid body transformation of equation 3.4 becomes:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} R & -R\mathbf{X}_C \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.8)$$

While the central projection of equation 3.2 is:

$$\begin{bmatrix} u' \\ v' \\ z \end{bmatrix} = \begin{bmatrix} fx + zu_0 \\ fy + zv_0 \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = K \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.9)$$

where you can notice the dimensionality reduction. If the central projection is mixed with the rigid body transformation we get:

$$\mathbf{u}' = KR[I] - \mathbf{X}_C \mathbf{X}' \quad (3.10)$$

Often, $\mathbf{x} = R\mathbf{X} + \mathbf{t}$ is used instead of equation 3.4. In this case, $\mathbf{t} = -R\mathbf{X}_C$ and we obtain:

$$\mathbf{u}' = P\mathbf{X} = K[R|\mathbf{t}]\mathbf{X}' \quad (3.11)$$

More details about homogeneous coordinates are found in Mikhail et al. (2001); Hartley and Zisserman (2003).

3.3 2D-3D Image Orientation or Space Resection

In the previous chapter, we presented the resection as a topometry and an early photogrammetry tool to retrieve the planar location of a point from azimuths measured in the direction of other points with known coordinates. The 3D equivalent is named space resection. The camera is oriented with correspondences found in the image (2D) and in the world (3D), namely Ground Control Points (GCP). The goal of the space resection is to estimate the parameters of the rigid body motion aligning the control points expressed in the world coordinate system with their coordinate in the camera system. A minimum of four points is required to estimate unambiguously the six unknowns of the camera orientation (Gruen and Huang, 2001).

Solutions to solve this problem appeared very early in the story of photogrammetry (Church, 1936). Before that time, iterative solutions (Lagrange, 1812) and graphical solutions (Monge, 1798) already existed. *"Since that time, due to its importance for applications of photogrammetry and for the same reason in computer vision in recent time, publications on 2D-3D image orientation may be regarded as a never-ending story"* (Gruen and Huang, 2001).

It is necessary to distinguish solutions developed for calibrated cameras (having 6DOF) from the solutions for uncalibrated camera that estimate also inner parameters. Moreover, some methods are direct while other methods are iterative and require an initial approximate solution. Often direct solution, less accurate but fast, are used to initialize iterative solutions. In this thesis, we will apply already existing 2D-3D image orientation methods. Hence, we will describe briefly methods that are in use in our work.

3.3.1 Least-Square Methods

Photogrammetrists and surveyors have a long tradition of using least-squares methods. In photogrammetry, least-squares are used to solve the bundle adjustment, which is an extension of space resection to a set of images (Gruen and Huang, 2001; Hartley and Zisserman, 2003). Least-square methods

to estimate the orientation parameters aims at minimizing the reprojection error (i.e. the distance between the projection of the 3D coordinates of the GCP and the corresponding 2D coordinates in the image). The least-square method is derived from the Equation 3.6 and minimizes the reprojection error:

$$\min_{p \in \mathbb{R}^6} \sum \|F(\mathbf{p}, \mathbf{X}) - \mathbf{u}\|^2. \quad (3.12)$$

where F is the functional model, \mathbf{p} is the vector of the parameters (or unknowns) and X and u the observations having random errors. In this thesis, we will reduce the dimension of \mathbf{p} when the location of the camera is known and only the camera angles require an estimation. However, it can also be extended if the inner parameters need to be refined.

Least-square methods applied to non-linear problems, such as space resection, are iterative and require an *a priori* value of the pose. F is then linearised around the *a priori* values using the Jacobian. Least-square methods are known to be accurate and this is why they are often used in conjunction with, less accurate, closed-form solutions, aiming at computing a pose without *a priori* values. Among the least-squared based algorithms, the Levenberg-Marquardt (LM) method is often used in computer vision and photogrammetry (Hartley and Zisserman, 2003). It combines the algorithms of Gauss-Newton (converge rapidly close to a minimum) and gradient descent (used to reach the neighborhood of the minimum).

3.3.2 Direct Linear Transform

However, if there is no prior for the pose of the camera, it is necessary to compute a solution in closed-form. Abdel-Aziz and Karara (1971) proceeded to a change of variables and transform the collinearity equation in a homogeneous linear system and the equation 3.11 can be expressed explicitly with the coefficient p_{ij} of P :

$$\begin{bmatrix} uZ \\ vZ \\ z \end{bmatrix} = \mathbf{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.13)$$

$$u = \frac{p_{11}X + p_{12}Y + p_{13}Z + p_{14}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}} \quad v = \frac{p_{21}X + p_{22}Y + p_{23}Z + p_{24}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}} \quad (3.14)$$

If six or more 2D-3D correspondences are provided the coefficients of P can be computed (the 12 unknowns). As the problem is reformatted in a linear form, the resolution is non-iterative. However, the solution of the DLT is the matrix \mathbf{P} which transforms the 3D coordinates in 2D coordinates. The coefficients of this matrix are not directly transposable in the perspective model, but methods exist to retrieve the orientation parameters as well as the focal and the coordinate of the principal point Mikhail et al. (2001); Melen (1994). A drawback of the DLT is that it is not able to compute a pose if the 3D correspondences are placed on a plane. DLT is implemented in our monoplottter, it is useful to compute the (initial) focal and coordinates of the principal point which can be refined with a Least-Square

method afterwards.

3.3.3 Perspective-n-Points

In computer vision, 2D-3D orientation is called PnP for Perspective-n-Points. In OpenCV (Bradski, 2000), the most widespread library for computer vision, the EPnP (Efficient PnP) is implemented (Lepetit et al., 2009). It is a non-iterative method to estimate the orientation of an image if the matrix K is known. It requires at least four correspondences and have a complexity growing linearly with n (the number of correspondences, $O(n)$).

Using the formulation of equation 3.11, they propose to express $n \geq 4$ control points X_i as a weighted sum of four virtual control points:

$$\mathbf{X}_i = \sum_{j=1}^4 \alpha_{ij} \mathbf{c}_j \text{ with } \sum_{j=1}^4 \alpha_{ij} = 1 \quad (3.15)$$

where c_j is a virtual control point. During the resolution, the coordinates of these virtual control points are computed. This method proves to be accurate and fast in comparison with state-of-the-art direct methods. Moreover, it is also general since working with $n \geq 4$ GCP and planar configurations. To improve further the accuracy of the pose, the authors recommend to use the solution of the EPnP as the initial solution of an iterative method such as Gauss-Newton.

3.4 Feature detection and matching

The detection of similar low-level features in overlapping images is an active computer vision field. In photogrammetry, feature detection and matching is widely used for the detection of tie-points, i.e. corresponding points across images. Usually, the matching is divided into three steps. The first is the feature detection, which aims at finding salient locations (features) in an image. The second is the feature description which generally uses a patch of image surrounding a feature to describe it. Finally, the matching retrieves corresponding features based on their description.

For instance, given the two images of Figure 3.4, we would like to estimate the transformation required to superimpose them. An operator to which this task is proposed could imagine two strategies. For the first, illustrated in black, he will try to align salient edges features such as the skyline or glacier limits. The second strategy is illustrated in red, he detects similar locations (or key-points) which define the parameters of the transformation. The goal of the feature detection is then to extract salient features such as edges and key-points from an image.

3.4.1 Feature detection

3.4.1.1 Edge detection

Many discontinuities occurring in the landscape give rise to edge in an image: depth, land cover, orientation, illumination discontinuities (Mikhail et al., 2001). These edges correspond to boundaries between objects recorded in GIS database. The most explicit example is the skyline which is remarkable in pictures and easily extracted from a DEM. Hence, edge detection has an extensive literature, we will

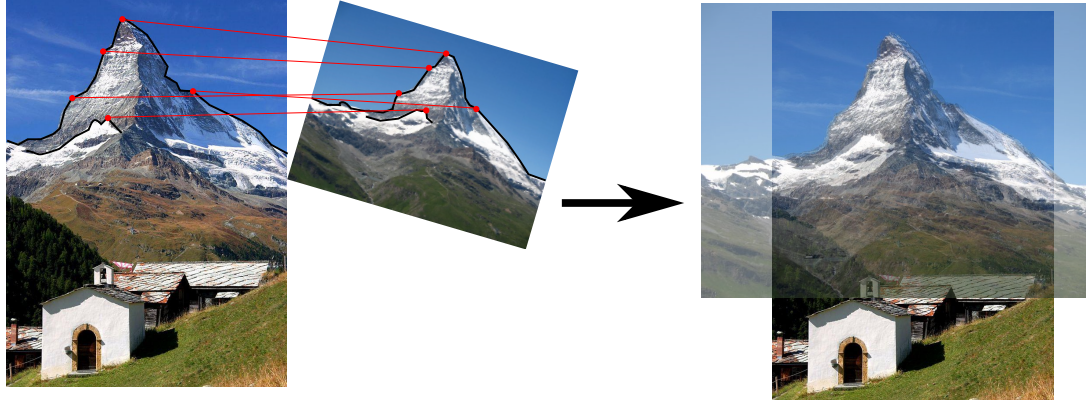


Figure 3.4 – Two images have to be superimposed. Correspondences between edges features (drawn in black), or key-points (red) provide transformation parameters.

only briefly describe its main principle.

An image I is composed of pixel located at x, y . For GIS specialists, it is natural to liken images to DEM and thus the pixel intensities to heights. In this scenario, edges are locations having a steep slopes (sudden change of intensities). Edge detectors are then often derived from the image gradient which is the first derivative in x and y directions and illustrated in Figure 3.5:

$$\mathbf{J} = \nabla I(x, y) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) = (I_x, I_y) \quad (3.16)$$

\mathbf{J} is a vector pointing in the direction of the steepest ascent. Its magnitude $\sqrt{I_x^2 + I_y^2}$ indicates how steep is the slope and its orientation $\arctan(I_y/I_x)$ is perpendicular to the edge. The partial derivatives are computed by convolving the image with an edge detector kernel. A threshold of the magnitudes provides edge features (second line of the Figure 3.5). They may require a further edge linking for some applications.

Various edge detectors have been developed, a review can be found in Ziou et al. (1998). A limitation of edge features is that they are not trivial to insert in algorithms used to retrieve a transformation, such as 2D-3D pose estimation, rather based on point correspondences. Edge detection is also noisy and does not provide directly a segmentation of land cover classes. However, edge detection is generally central for the matching of skylines.

3.4.1.2 Key-point detection

Thus, numerous methods privilege the detection of key-points which must be retrieved in overlapping images. The selection of the appropriate key-points depends on the application and the matching method that is subsequently applied to detect corresponding key-points. The simplest matching criterion used to compare two image patches, I_0 and I_1 , is a weighted Sum of their Squared Difference (SSD, Szeliski (2010)):

$$E_{WSSD} = \sum_i w(\mathbf{x}_i) [I_1(\mathbf{x}_i) - I_0(\mathbf{x}_i)]^2 \quad (3.17)$$

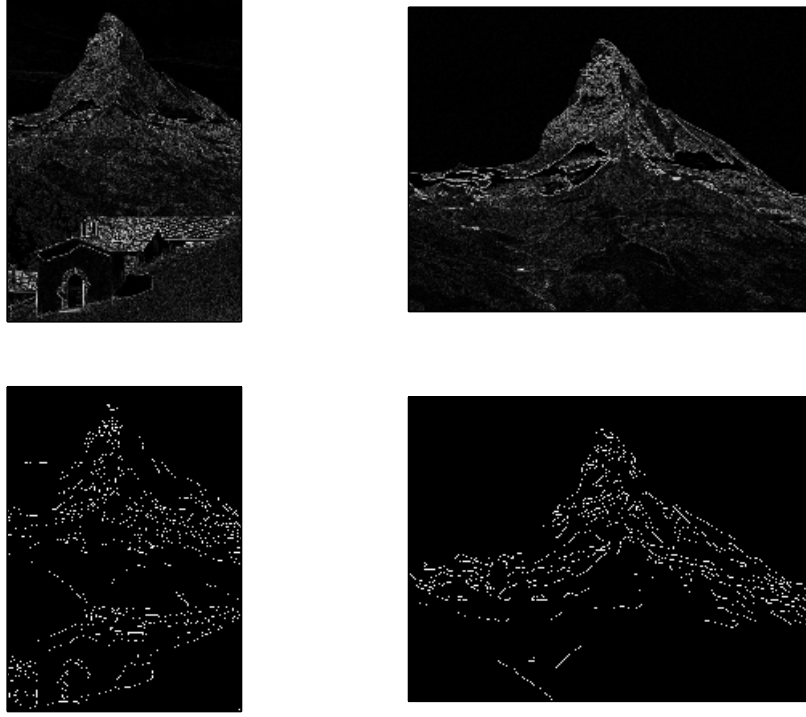


Figure 3.5 – Edge detection in the first line. A threshold on the magnitudes is applied then to extract edge features.

where w stores the weight of each pixel and the summation is over all the pixels i . At the time of the key-point detection, the second image is not available. However, similarly, we can measure the stability of an image patch by comparing it (blue frame) with its neighbourhood (red frame), as illustrated in Figure 3.6:

$$E_{AC}(\Delta \mathbf{u}) = \sum_i w(\mathbf{x}_i) [I(\mathbf{x}_i + \Delta \mathbf{u}) - I(\mathbf{x}_i)]^2 \quad (3.18)$$

for which $\Delta \mathbf{u}$ is a displacement vector. In the Figure 3.6 a patch of image is framed in blue, its similarity is measured with the red patch displaced by $\Delta \mathbf{u}$. If the patch is in the middle of a homogeneous surface, E_{AC} is small and does not variate with $\Delta \mathbf{u}$ (c). Oppositely, for a patch on a corner (a), E_{AC} varies in every direction. Finally, along an edge, E_{AC} varies mainly in the direction perpendicular to the edge (b). The key-points detection aims then at retrieving corners, which are rotationally invariant, that is to say corners vary mainly along two directions. Using a Taylor Series expansion (Lucas et al., 1981), the autocorrelation surface is approximated with:

$$E_{AC}(\Delta \mathbf{u}) \approx \sum_i w(\mathbf{x}_i) [\nabla I(\mathbf{x}_i) \cdot \Delta \mathbf{u}]^2 = \Delta \mathbf{u}^T \mathbf{A} \Delta \mathbf{u} \quad (3.19)$$

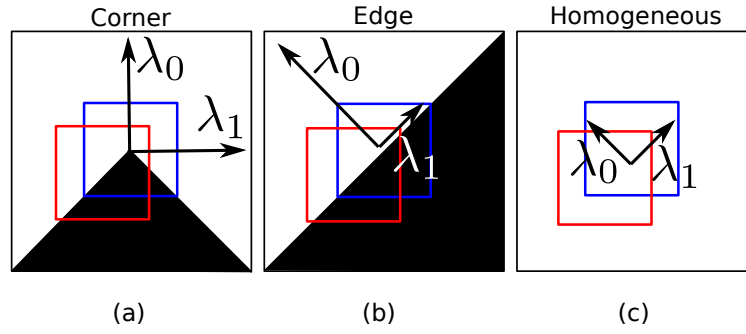


Figure 3.6 – Representation of the eigenvectors of E_{AC} for (a) a corner, (b) an edge, (c) an homogeneous surface.

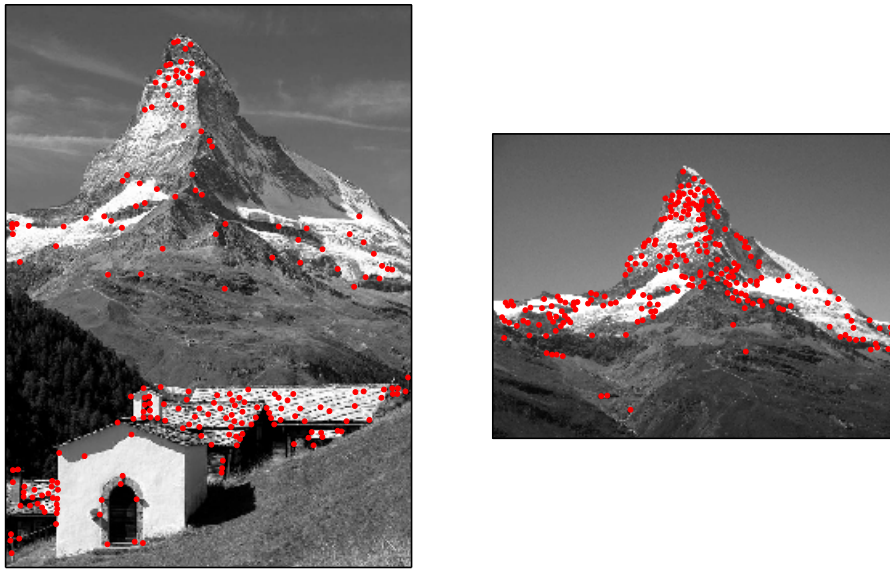


Figure 3.7 – Detection of Harris corner features.

where \mathbf{A} is:

$$\mathbf{A} = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (3.20)$$

As illustrated in Figure 3.6, the eigenvectors of \mathbf{A} point in the direction of the highest variation of E_{AC} with a magnitude λ_0 and λ_1 . Hence, to detect key-points, which are in this case corners, one can simply find maximums of the smallest eigenvalues (Shi and Tomasi, 1994).

The photogrammetrists Förstner (1986) and Harris and Stephens (1988) were the first to use rotationally invariant scalar measures derived from the auto-correlation matrix to locate key-points. The invariance to rotation means that in the case of the initial example (Figure 3.4), similar key-points are detected even if the second image is rotated. The detection of Harris key-points is illustrated in Figure 3.7. However, as illustrated in the example of Figure 3.4, the images can also have varying scales,

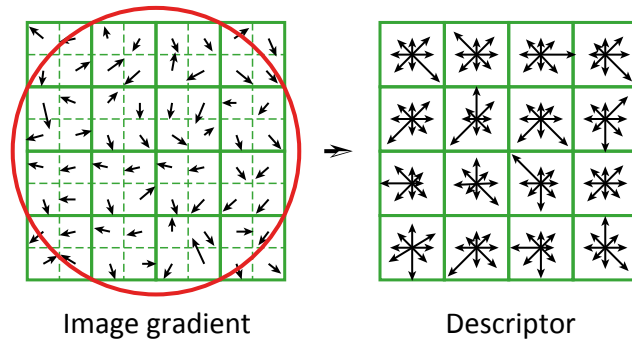


Figure 3.8 – Construction of a SIFT descriptor. (CC, Wikimedia)

in this scenario, features should not only be rotationally invariant but also scale invariant. Lowe (2004) looks for 3D extrema in spatial and scale space.

Finally, for instance for wide-baseline stereo matching (i.e. there is a large distance between the viewpoints of the two images and thus there are large visual deformations), affine invariance of the key-points is also required. In this case, one can fit an ellipse to the autocorrelation matrix and use its axes as the affine coordinate frame. A review of the key-point detection can be found in Tuytelaars and Mikolajczyk (2007).

3.4.2 Feature description

The similarity of two key-points is generally measured by comparing the descriptions of patches surrounding the key-points. In the previous section, we presented the Sum of Squared Difference (SSD) to measure directly the similarity of the intensities of two patches. SSD or correlation (Normalized Cross-Correlation, NCC) are appropriate if no rotation or scaling are expected. However, for more complex scenarios, these hypotheses are not respected and more evolved descriptors are required. Some successful descriptors such as SIFT (Lowe, 2004), GLOH (Mikolajczyk et al., 2005) are based on a histogram of gradient orientation computed for sub-regions of the image patch. In SIFT, illustrated in Figure 3.8, the image patch has 16 pixels and in each 4 sub-patches an 8 bin histogram of orientation gradient is computed, resulting in a descriptor of size 128. GLOH is quite similar, but a log-polar structure rather than quadrants is used to compute the histograms. A review of key-point detectors and descriptors in the context of photogrammetry can be found in Remondino (2006).

3.4.3 Feature matching

Once each key-point is described, the distance between two descriptors can be used to measure their similarity. Again, the matching strategy needs to be adjusted to the problematic. Typically, in the case of Figure 3.7, all the key-points on the bottom of the left-image do not have a correspondence in the second image. Thus, if for each key-point, the nearest neighbor in the second set is accepted, many false correspondences are matched. Two other strategies are then applied usually. The first is to set a maximum distance. The second solution, which often provides good result for disparate images is the Nearest Neighbor Distance Ratio (NNDR, Lowe (2004)). For this strategy, the ratio of the distances of the two nearest neighbour is observed. If this ratio is close to one, then the two nearest neighbours are very similar and can generate a false correspondence. In contrast, if this ratio is big, the first neighbour

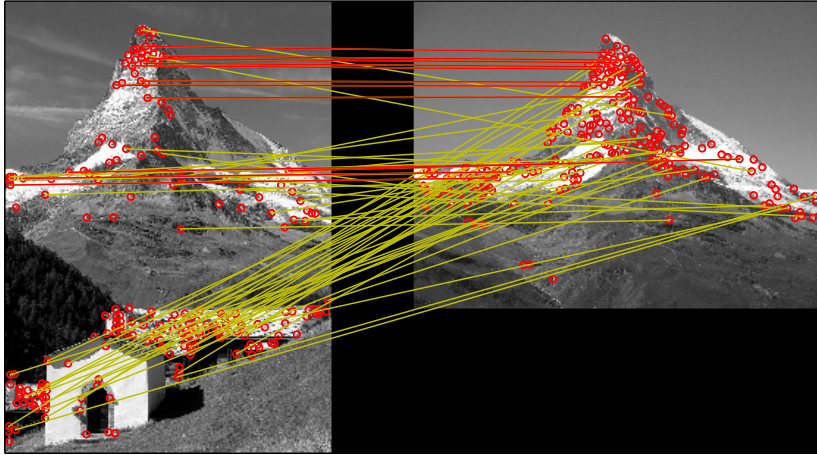


Figure 3.9 – Matching of key-points. True correspondences are linked with a red line.

is much more similar than the second neighbour. It is then likely to be a true correspondence.

In other scenarios, for which the images are expected to be very similar, one can first detect a feature in the query image. The patch surrounding the feature is compared with each patch of the target image using a sliding window (that is to say, measure the similarity in every location rather than key locations). Usually, this dense matching is used in association with NCC or SSD to measure the similarity. In this way, the feature detection in the target image is skipped. For instance, it can be justified if the feature detectors are not able to select similar key-points across the images.

3.5 Simultaneous pose estimation and matching

Once key-points have been matched according to their description and a matching strategy, a set of pairs of key-points is obtained. As illustrated in Figure 3.9, this set typically contains a certain amount of false positive correspondences (wrongly matched pairs). They have to be excluded to provide an accurate registration. A simple method to sort out the good correspondences is the insertion of a geometric constraint. For instance, in Figure 3.9, one can see that true correspondences are parallel (red lines). Similarly, in the initial example (Figure 3.4, p. 30), we expect a similarity transformation (translation, rotation and scaling) to overlap the images. We will show two solutions which, by adding the expected transformation model, exclude outliers and also compute the parameters of the transformation.

3.5.1 Noisy correspondences: RANSAC

RANSAC (Fischler and Bolles, 1981), for Random Sample Consensus, is a very simple and efficient technique to add the model information to the matching. An example is presented in Figure 3.10. Some points are expected to be on a line. If all the points are used to estimate the parameters of the line, they result in the dashed line, highly perturbed by the outliers. If the two red points are selected randomly to estimate the line, this line corresponds to one other point (within the blurred region). In contrast, if the two green points are selected, the estimated line corresponds to a maximum of other points.

In the context of key-point matching, firstly, a set of potential correspondences is computed, as in

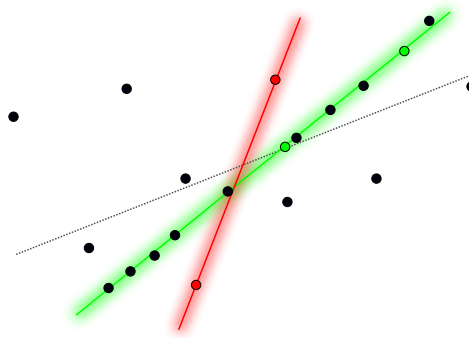


Figure 3.10 – Illustration of the RANSAC algorithm, a line has to be fitted to the points.

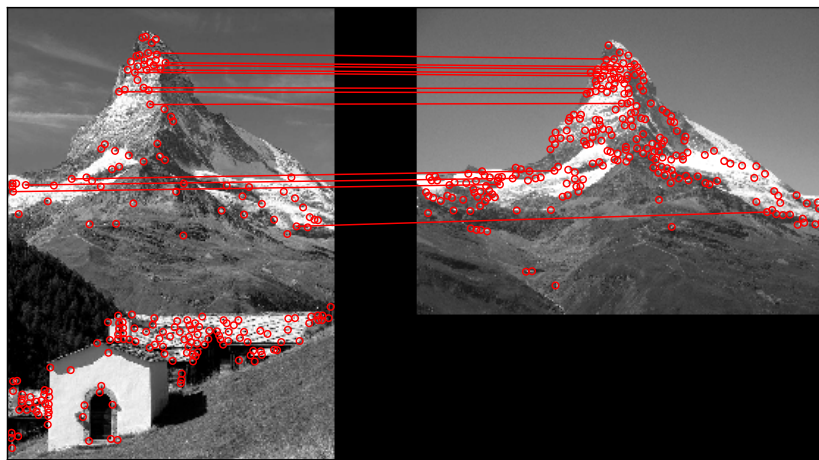


Figure 3.11 – Correspondences resulting from an affine transformation are extracted from the correspondences presented in Figure 3.9.

the Figure 3.9. The RANSAC algorithm is then the following:

- First, some correspondences are selected at random and used to estimate the parameters of the transformation.
- Second, all the other pairs corresponding to this model are selected.
- Third, an error is associated with this model. The error typically takes into account the number of matches and their fitting with the model.
- These steps are repeated iteratively. At the end of the iterations, the parameters which minimize the error, and the corresponding set of correspondences, are returned.

The result of RANSAC for the matching of the Harris key-points is presented in Figure 3.11. RANSAC can be used with several models, and works perfectly fine with a perspective model.

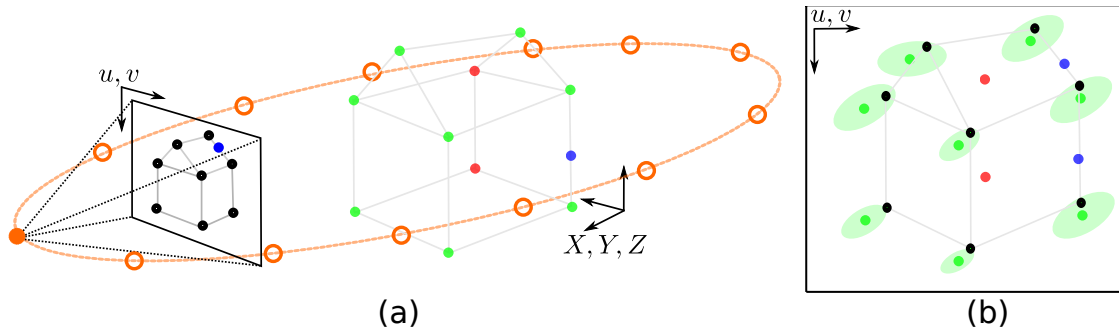


Figure 3.12 – (a) A set of 2D key-points extracted from an image (black dots) has to be matched with the corresponding set of 3D key-points extracted from the 3D model (green dots). The pose prior is represented by potential locations in orange. (b) The 3D points are projected with their error ellipses in the image limit potential matches.

3.5.2 No correspondences, but pose prior: Blind-PnP

A more complex scenario is when the key-points are available, but they are not matched with a descriptor. Namely, each key-point is potentially matched to any other key-point. This situation is illustrated in Figure 3.12 (a), a set of 3D points is detected in a 3D model and in an image of this model. Some key-points are obstructed (red) and thus do not appear in the image. Other key-points are falsely detected (blue). The goal is to retrieve the pose of the image given these key-points. If RANSAC is applied in these conditions, a high amount of iterations have to be computed (each 2D point is the potential match of each 3D point) and it is not ensured to converge to the best solution.

Moreno-Noguer et al. (2008) propose an approach, called Blind-PnP, to solve this problem. Blind-PnP is based on the hypothesis that some priors about the camera location and orientation are available. In our example, these priors can be that the camera is located somewhere on the orange ellipse and is facing the building. For each pose tested (an orange point) and associated with a standard deviation, only a subset of 2D-3D correspondences is then possible. Indeed, the 3D key-points and the standard deviation are projected in the image, resulting in the green confidence ellipses). Thus, only the black and green points within a similar ellipse are potential matches. The potential correspondences are then iteratively used to initiate a Kalman filter. The goal of the Kalman filter is to use the detected 2D-3D correspondences to improve the initial pose and its confidence (and thus reduce the ellipses). Similarly to RANSAC, the final pose is the one which generates many correspondences with small reprojection errors.

This scenario is close to our problem in the sense that we also have a 3D model and a 2D image. Moreover, our images are of different kinds: 2D real images on the one hand and 3D synthetic images on the other hand. We can then neither expect a feature detector to find similar locations in the images, nor a feature descriptor to be entirely adapted for the matching of patches of different kind. Finally, as discussed in the introduction, most of the images found on the web or in the archives are associated with a prior which can be used to initiate the pose estimation and limit the potential correspondences. The method described in the next section, that we developed to simultaneously estimate the pose and match correspondences is directly derived from Blind-PnP.

3.5.3 Pose Estimation with Pose and Landscape Model Priors, PEP-ALP

We introduced RANSAC, whose goal is to compute a pose from a set of 2D-3D correspondences containing false correspondences. Next, we presented Blind-PnP as a solution to compute a pose from a set of 2D key-points, a set of 3D key-points (but no correspondences between them) and a prior on the pose. In our scenarios, we typically have a prior on the pose, that we want to insert in the pose estimation (which discards the use of RANSAC testing). We also have noisy 2D-3D correspondences matched by their appearance. We propose then to modify the method of Moreno-Noguer et al. (2008) to insert 2D-3D correspondences matched with a feature descriptor. Our method becomes then sensibly similar to the one of Serradell et al. (2010) which is the extension of Blind-PnP to correspondences having appearance prior. However, we test only one prior (rather than multiple priors), estimate a camera orientation rather than a homography and we consider several correspondences at each iteration. We will call our approach PEP-ALP for Pose Estimation with Pose And Landscape model Priors.

Similarly, we will use a Kalman filter (Kalman, 1960) to compute the pose from 2D-3D correspondences. Kalman filter was developed to estimate the parameters of a model from a succession of observations. Typically, it is used with real-time observations or observations coming from several sensors (for instance, a GPS and IMU to estimate the successive locations of a vehicle). With the first measurement provided by the sensors, a location is estimated. Next, it is refined with new measurements. A Kalman filter keeps a memory of the successive states to avoid that a corrupted block of observations perturbs the location estimation.

In our problem, we want to estimate the pose of a camera from several blocks of 2D-3D correspondences. Initially, the pose is started with the prior (geotag location and a potential azimuth), p_0 . It is associated with a standard deviation Σ^{p_0} corresponding to the uncertainty of the location and orientation. Next, a first set of 2D-3D correspondences is detected. It updates the pose, but takes into account the previous state $p_1 = \Delta p + p_0$ where Δp is a "gain" and will be defined later. These correspondences also modify the uncertainty of the pose Σ^{p_1} . The knowledge of the pose and its standard deviation can be used to reduce potential 2D-3D correspondences. Indeed, a 3D key-point X_i can be projected in the picture with the collinearity equation $\mathbf{u} = \mathbf{F}(\mathbf{p}_1, \mathbf{X}_i)$ and associated with a confidence ellipse. Hence, only the 2D key-points located within this ellipse are possible correspondences of X_i (with a 95% confidence for a 2σ confidence ellipse). A new set of correspondences, less impacted by false positives is computed. We can then iterate this process.

A toy example is presented in Figure 3.13. The translation which aligns the query image Q with the reference image R has to be recovered from key-points (black dots) matching (assuming a metric to match their description). At the initial state, the center of the query image (p_0 , red dot) is expected to be within the red ellipse (Σ_0^p). Thus, the key-point \mathbf{X}_0 is expected to have its correspondence within the projected ellipse in the query image (dashed, Σ_0^u). This key-point has then to be compared with all the key-points within this ellipse. A correspondence is found (red line), it is used to refine the estimate of the translation (p_1) and thus reduce the covariance of the translation (orange ellipse, Σ_1^p). The next key-point \mathbf{X}_1 must be compared with a reduced amount of key-points in the query image. For pose estimation from 2D-3D correspondences, the context is similar, but 3D key-points are extracted from the reference image R and projected in the query image Q with a confidence ellipse.

We can now describe the steps of this process in more details. We want to minimize Equation 3.12

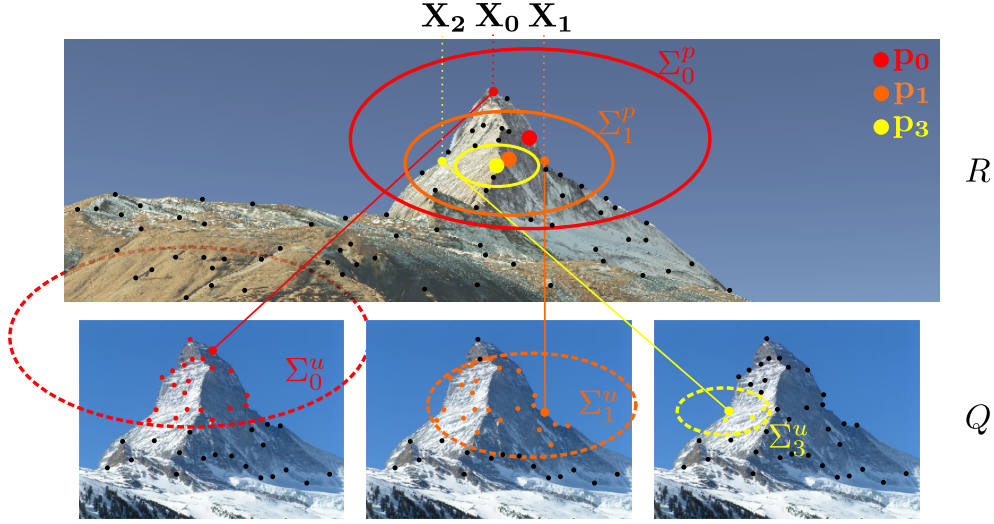


Figure 3.13 – Toy example of the pose estimation with a Kalman Filter, here a translation estimated from 2D key-points. A key-point of the reference image (top) is projected in the query image (bottom). The error ellipse delineate the region of potential matches.

(p. 28) having J successive blocks of 2D-3D observations:

$$\min_{p \in \mathbb{R}^6} \sum_{j=1}^J \|F(\hat{p}, \mathbf{X}) - \mathbf{u}\|^2. \quad (3.21)$$

In our Kalman filter, a pose p_j is associated with a covariance matrix Σ_j^p and the noise covariance of a measurement (a 2D-3D correspondence) is Σ_j^u . A block of observations is constituted of n 2D-3D correspondences. Thus, the jacobian A_j of a block of observations is:

$$A_j = \begin{bmatrix} \frac{\partial F_0(p_j, \mathbf{X}_0)}{\partial \alpha} & \frac{\partial F_0(p_j, \mathbf{X}_0)}{\partial \beta} & \frac{\partial F_0(p_j, \mathbf{X}_0)}{\partial \gamma} & \frac{\partial F_0(p_j, \mathbf{X}_0)}{\partial X_C} & \frac{\partial F_0(p_j, \mathbf{X}_0)}{\partial Y_C} & \frac{\partial F_0(p_j, \mathbf{X}_0)}{\partial Z_C} \\ \frac{\partial F_1(p_j, \mathbf{X}_0)}{\partial \alpha} & \frac{\partial F_1(p_j, \mathbf{X}_0)}{\partial \beta} & \frac{\partial F_1(p_j, \mathbf{X}_0)}{\partial \gamma} & \frac{\partial F_1(p_j, \mathbf{X}_0)}{\partial X_C} & \frac{\partial F_1(p_j, \mathbf{X}_0)}{\partial Y_C} & \frac{\partial F_1(p_j, \mathbf{X}_0)}{\partial Z_C} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial F_0(p_j, \mathbf{X}_n)}{\partial \alpha} & \frac{\partial F_0(p_j, \mathbf{X}_n)}{\partial \beta} & \frac{\partial F_0(p_j, \mathbf{X}_n)}{\partial \gamma} & \frac{\partial F_0(p_j, \mathbf{X}_n)}{\partial X_C} & \frac{\partial F_0(p_j, \mathbf{X}_n)}{\partial Y_C} & \frac{\partial F_0(p_j, \mathbf{X}_n)}{\partial Z_C} \\ \frac{\partial F_1(p_j, \mathbf{X}_n)}{\partial \alpha} & \frac{\partial F_1(p_j, \mathbf{X}_n)}{\partial \beta} & \frac{\partial F_1(p_j, \mathbf{X}_n)}{\partial \gamma} & \frac{\partial F_1(p_j, \mathbf{X}_n)}{\partial X_C} & \frac{\partial F_1(p_j, \mathbf{X}_n)}{\partial Y_C} & \frac{\partial F_1(p_j, \mathbf{X}_n)}{\partial Z_C} \end{bmatrix}_{n \times 6} \quad (3.22)$$

The gain K_j associated with a block of 2D-3D observation is:

$$K_j = \Sigma_j^p A_j^T (A_j \Sigma_j^p A_j^T + \Sigma_j^u)^{-1} \quad (3.23)$$

The gain is applied to update the pose:

$$p_{j+1} = p_j + \underbrace{K_j (\mathbf{u}_j - A_j p_j)}_{\Delta p} \quad (3.24)$$

And update its covariance:

$$\Sigma_{j+1}^p = (I - K_j A_j) \Sigma_j^p \quad (3.25)$$

The Kalman Filter is iterative, each new block updates the pose and the covariance. If the observations are trustful, the pose is refined and the covariance reduced. Based on this observation, we want to refine the correspondences detection occurring after each update. By using variance propagation, one can draw confidence ellipses of a 3D correspondence \mathbf{X}_i in the image:

$$\Sigma_i^u = A_i \Sigma^p A_i^T \quad (3.26)$$

From the covariance matrix Σ_i^u , one can compute the axes of the confidence ellipse associated with the projection of \mathbf{X}_i in the image. This ellipse encloses potential 2D locations of the \mathbf{X}_i and thus its potential correspondences. These ellipses improve the robustness of the key-point matching by limiting potential 2D-3D correspondences.

Hence the algorithm of the pose estimation and matching with a Kalman filter can be summarized with this pseudo-code.

```

Data: 2D key-points =  $u_Q$ , 3D key-points =  $X_R$ , location prior =  $\mathbf{T}_0$ , orientation prior =  $\mathbf{r}_0$ 
Result: query image pose =  $\mathbf{p}$ 
 $p \leftarrow (\mathbf{T}_0, \mathbf{r}_0);$  /* Initialisation of the pose */
 $\Sigma^p \leftarrow \text{diag}(\sigma_X, \sigma_Y, \sigma_Z, \sigma_\alpha, \sigma_\beta, \sigma_\gamma);$  /* Initialisation of the covariance matrix */

while  $n \leq nIteration$  do
     $u_R \leftarrow \text{Projection}(p, X_R);$  /* Projection of the 3D key-points in the image */

     $\Sigma_R^u \leftarrow \text{VariancePropagation}(p, X_R, \Sigma^p);$ 
     $\text{ellipses} \leftarrow \text{drawConfidenceEllipses}(u_R, \Sigma_R^u);$ 
     $M \leftarrow \text{MatchWithinEllipses}(u_Q, u_R, \text{ellipses});$  /* Matching geometrically constrained */
     $p, \Sigma^p \leftarrow \text{Kalman}(p, \Sigma^p, M);$  /* Kalman-Filter based pose estimation */
end
return  $p;$ 

```

The goal of this algorithm is to take into account the pose and landscape model priors existing for a picture. These priors have two impacts. First, they force the computed pose to be within a feasible region. Second, they limit the potential correspondences between the 2D and 3D key-points, this allows the matching to be more robust by eliminating noisy correspondences, to be less sensitive to the appearance. Thus, the pose estimation is also more accurate by getting more correspondences. PEP-ALP will be used in Section 5.1 to estimate the pose from skyline features and in Section 5.3 to estimate the pose of an image based on already georeferenced images.

At this stage, we have defined the tools that we will implement in the next chapters. We have defined the relation between a picture and 3D objects. It is required to project geographic information in the picture or to generate synthetic images from a DEM and a texture. Next, we discuss some pose estimation algorithms, their goal is to compute the orientation of an image from 2D-3D correspondences. If these correspondences are not detected by an operator, they have to be detected by image matching. This is why we discussed the key-point detection, description and matching in this section. Finally, we combined the pose estimation and the key-points matching in a single loop.

4 Monoplotter: Pic2Map, a plugin for the integration of pictures in QGIS

In this chapter, we will present a plugin created to integrate photographs in QGIS, an open-source GIS. First, the pose of the image is computed with Ground Control Points. This pose relates directly the picture pixels to GIS data. First, this software took the form of a set of Python functions presented in Produit and Tuia (2012) and second, thanks to G. Milani (Milani, 2014), it was developed and implemented as a plugin for QGIS. In the proposal of this thesis, the implementation of this software was not planned. However, for the implementation of the algorithms dedicated to the registration of an image with a landscape model, we had to develop a set of functions, including:

- The pose estimation from 2D-3D correspondences (GCP).
- The generation of 3D synthetic images from a DEM textured with an orthoimage (virtual globe).
- The projection of geographic vector data in the image (reference vector model).

Only a small additional effort was then required to create a software dedicated to the manual georeferencing of oblique images and their complete interaction with a GIS (namely a monoplotter). We took this opportunity to implement our own monoplotter, allowing us to compare manual and automatic registration. We propose some improvements compared to existing solutions and yet it is available for the GIS community.

4.1 Introduction

Single images require a particular processing to extract geographic data from their content. Obviously, regular methods applied in photogrammetry, based on the matching of key-points across a set of images, cannot be applied to single images. Their orientation, bound to a geographic coordinate system, is then usually set manually.

We define a **monoplotter** as a tool designed to compute the orientation of a single image from Ground Control Points. GCP are point correspondences detected respectively in a georeferenced layer (orthoimage, topographic map) and in the image, they correspond to the set of 2D-3D correspondences required for pose estimation. The monoplotter exploits subsequently a DEM to assign a world coordinate to each pixel of the image. The digitization of GCP is time-consuming. Hence, the manual georeferencing of oblique images is essentially used in research and applied to pictures having a high value: historical images (Bozzini et al., 2012), but also images recorded continuously to sense a land-

scape with high temporal and spatial resolutions (Corripio, 2004; Messerli and Grinsted, 2014; Crouzy et al., 2015). However, if the manual pose estimation was more user-friendly other opportunities could be created outside the academia.

Among the existing monoplotters, three are really accessible to users. First, the MonoPlotting Tool presented in Bozzini et al. (2012) is probably the most advanced (*Monoplotting Tool* 🐞, Version 1.5.1, Swiss Federal Institute for Forest, Snow and Landscape Research, Bellinzona, 2015). Its main drawbacks are that the software is developed only by one person and is proprietary. Hence, the software development responds only slowly to particular requirements coming from the community of users. Moreover, it is yet only available for one Operating System. Finally, it is not integrated with a GIS (even if it reads and writes GIS standard formats), which is an additional limit for the usability. A second monoplotter is provided within the poorly widespread GIS ILWIS (*ILWIS GIS*, 2012, 52° North ILWIS Community, ilwis.org 🐞). It was meant for aerial images and lacks some functions such as the projection of GIS data into the image. A third and recently developed monoplotter, called Imgraff 🐞, is presented in Messerli and Grinsted (2014) and consists of a set of Matlab functions. It is then mainly thought for and used by scientists. An alternative method to estimate the pose of a picture is the browsing of a virtual globe, such as Google Earth "add photo" function (*Google Earth*, Google Inc., Mountainview, 2015). This method is much more inaccurate than GCP, but it is more suited to get a quick rough estimate of the pose in a user-friendly way. However, virtual globes are not meant for the extraction of geographic data from the pictures, but only their visualization.

Hence, through Pic2Map, we would like to make the georeferencing of oblique images easily accessible to end-users. Indeed, the plugin is open source and can be developed by scientists and programmers for particular needs. Moreover, it is available for one of the most widespread open GIS and will thus reach the community of the GIS professionals and amateurs. Finally, it can render reference GIS data in 3D to enable a hybrid georeferencing (user-friendliness of virtual globes and accuracy of 2D-3D camera orientation).

4.2 Description of a monoplotter

A monoplotter has three main functions: first, the pose estimation from GCP, second, the projection of geographic data in the picture and finally, the monoplotting functions, which embrace the interaction of the image with the geographic data. These functions are illustrated in Figure 4.1: on the left-side (a) the workflow of the pose estimation is presented, the right side (b) explains the interaction of the picture with the GIS and with geographic data.

There are two mandatory datasets for a monoplotter. The first is a picture (Figure 4.1, **A**), not necessary oblique. However, since collections of aerial nadir images are easily processed by aero-triangulation, monoplotters are generally used with terrestrial images. Second, the depth of a pixel cannot be measured from a single image, a DEM (**B**) is the reference for the heights. The DEM is used both to get the altitude of GCP and to compute the 3D coordinates of a pixel. Finally, a third data set is recommended for the GCP digitization: a "map" of the area on which a user can recognize control points also visible in the oblique image (**C**). Usually, if available, orthoimages should be preferred because they have a good accuracy, high-resolution and are visually more similar to pictures than topographic maps. The orthoimage and DEM can be combined in a 3D synthetic image (**C**).

Typically, monoplotters process images which are not associated with an orientation, because they are beyond the framework of the bundle adjustment and are not georeferenced directly (with instruments). Hence, in our monoplotter, we propose to the user to browse the 3D virtual environment

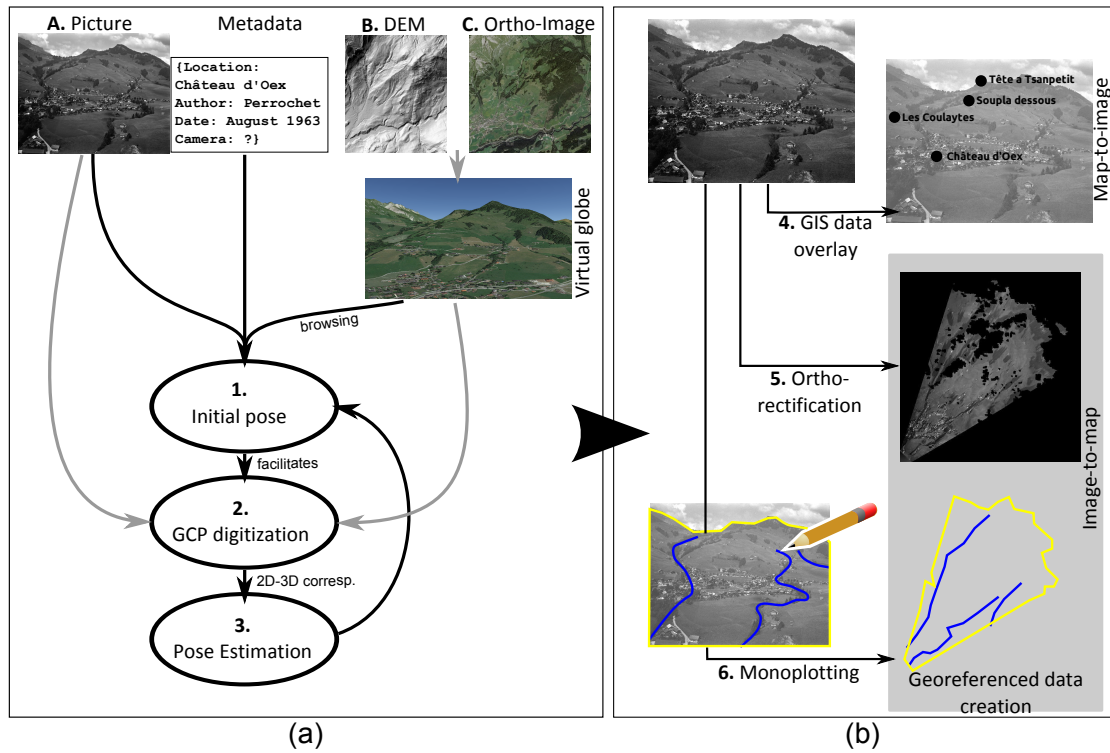


Figure 4.1 – (a) Workflow of the pose estimation. (b) Illustration of the interaction of a picture with GIS data. (*Archives de la Construction Moderne - EPFL, Swisstopo*)

to get an initial pose of the camera (Figure 4.1, **step 1**). Next, this pose is refined with GCP (**step 2**), points clicked in the image and in a georeferenced layer (an orthoimage) which are the 2D-3D correspondences required for pose estimation (**step 3**). The pose estimation is presented in Section 4.3.2. The pose bound the image pixels with the world coordinates system. Thus, there are two possible directions for the combination of a picture and georeferenced data. First, the 3D geographic data is projected in the image (**step 4**), we will call this function *map-to-image* and it will be presented in Section 4.3.4. Second, the image is projected in the GIS (**step 5 and 6**), we will call these functions *image-to-map* and we will present their implementation in Section 4.3.5.

4.3 Description of the implementation

4.3.1 Structure of the database

The goal of a monoplottter is to provide geographic attributes and geographic functions to photographs, such as illustrated in Figure 4.2. Black attributes highlight compulsory attributes of a picture, gray ones are optional, orange attributes are geographic attributes that an image may have before its orientation (but are refined after the orientation), red attributes are those which are computed after the orientation and are not existing in usual image databases. The Figure is divided into two parts. On the top, the attributes of an image are presented. The picture is related with geographic layers represented on the bottom. The gray area contains the geographic layers, included those generated from the image.

Description of the relations: An image is shot with a particular camera which has a brand, a model.

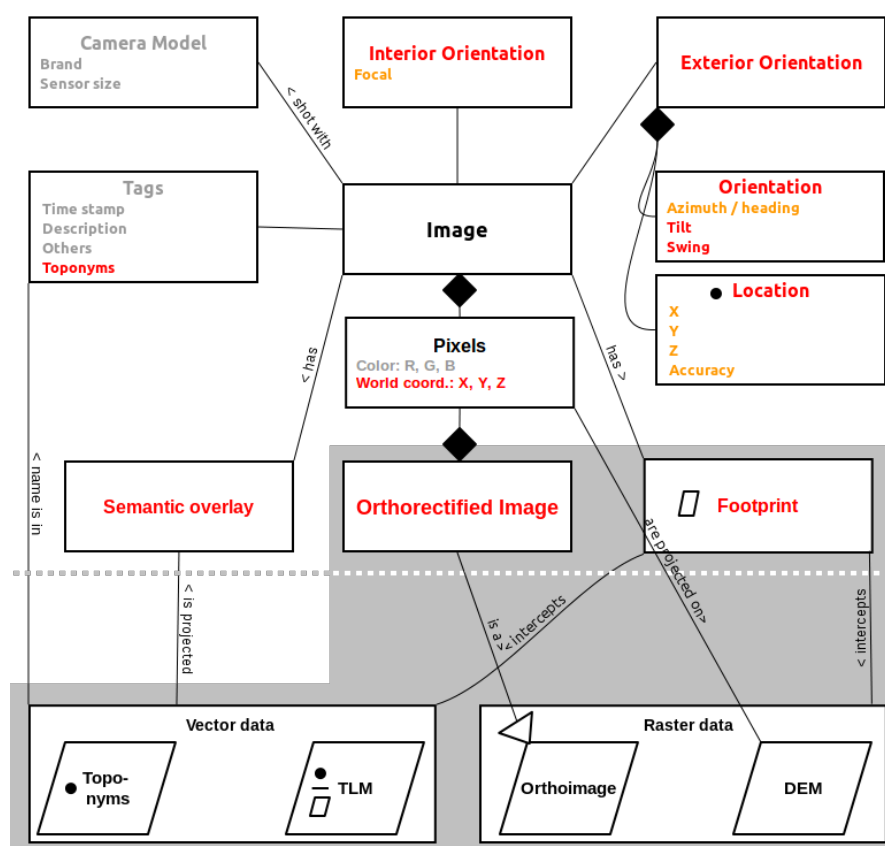


Figure 4.2 – Structure and relation between the data inserted in a GIS dedicated to the monoplotting.

The model can be associated with some physical properties (such as the sensor size). Pictures shot with recent cameras store the focal (zoom) of the shot. Sometimes, textual descriptions (or tags) are linked to the picture: a date, a location name, a description of the image content. Tags may refer to the name of a place or of a landscape object (i.e. the name of a glacier, of a summit or of a village).

The photographer shoots a picture from a particular location: world coordinates are used to describe the location of the projection center of the camera. The camera is pointed in a particular direction, which is described with camera orientation angles. The camera location and orientation compose the exterior orientation which stores the parameters used to orient a camera with respect to a coordinate system.

An image is composed of pixels. Usually, pixel colors are described by a triplet measuring the red, green and blue composition (it can also be a black and white image and have only one band). If the orientation of the picture is computed, we can also compute the world coordinate of each pixel (compulsory to measure objects in the image or to compute the orthorectification of the image). Hence, the image can be extended with three more bands storing the X, Y and Z coordinates. An image has a footprint (or viewshed): the polygon enclosing areas visible in the photography.

If the picture is georeferenced, relations with geographic layers are possible:

- **Toponyms:** Location names associated with world coordinates. They can have a correspondence

in the description of an image or they can be overlaid on an image as an Augmented Reality layer (semantic overlay). This function can also be used to complete the image tags, for instance by adding the name of visible locations.

- **Topographic Landscape Model:** Vector-based topographic objects with associated attributes. Similarly to toponyms, their names can be found in the description of an image and they can be overlaid on the image (semantic overlay).
- **DEM:** Raster-based 3D model of the landscape. It is used to compute the world coordinates of the pixels, as well as the footprint of the image. The projection of each image pixel on the DEM generates an ortho-rectified image.
- **Orthoimage:** They are generally computed from the ortho-rectification of aerial images.

In this section, we described the data inserted and created by the monoplottor. In the next section, we will explain how the camera orientation is computed.

4.3.2 Pose estimation

The most tedious task in monoplottor is the detection of GCP. To improve the user-friendliness, the GCP detection has to be simplified as much as possible. There are two levers to ease the GCP digitization. The first aims at helping the user to digitize the GCP. The recognition of similar points in an orthoimage and in an oblique image is difficult due to the difference of viewpoint (top orthogonal vs side perspective). Hence, we propose a 3D rendering of the orthoimage. These synthetic 3D images are visually more similar to real images and greatly help the user. The second lever is the computation of a pose with few GCP. As suggested in Figure 4.1 (a), both levers are interlaced. Indeed, a pose is required to compute a 3D view, and the 3D view facilitates the digitization of additional GCP.

The Pic2Map plugin offers a 3D navigation interface to detect a prior rough pose. The navigation in a 3D virtual globe is particularly useful if the search area is wide, either because the user does not know the area or because the image only has a rough georeference (as a toponym). Based on this initial pose, the user can digitize GCP in the 3D view or in the GIS (both interact). Typically, the 3D view should be preferred for a rough digitization of the GCP while the orthoimage allows the user to improve their accuracy. These 2D-3D correspondences are then inserted in a pose estimation algorithm (presented in Section 3.3). With 4 GCP and assuming a normal focal (or iterating over some potential values of the focal length) and the principal point at the image center, we can estimate the pose using EPnP (Lepetit et al., 2009), (not yet implemented in Pic2Map, but presented in Produit and Tuia (2012)). By adding more GCP and using the DLT-based pose estimation (Abdel-Aziz and Karara, 1971), the focal and the principal point location are estimated as well. Often, the user has knowledge of some of the exterior and interior parameters (a GPS-measured location, the focal recorded by a digital camera etc.). It is then required that the operator can fix these parameters during the pose estimation. In these situations, a first estimation of the pose is obtained with EPnP or DLT and is subsequently refined with a least-square method (Levenberg-Marquardt, Hartley and Zisserman (2003)) in which these parameters are fixed by the user.

4.3.3 3D-rendering on the GPU

Currently, our computers have efficient graphic cards, which are meant to be fast for the rendering of 3D virtual environments (Computer games, Computer Assisted Drawing etc.). The graphic card or

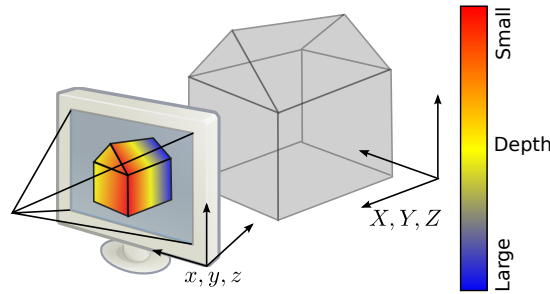



Figure 4.3 – The primitives of the model are projected on the monitor and rasterized at the size of the monitor pixels. Only the pixels the closest to the projection center are displayed.

Graphical Processing Unit (GPU) is accessible to programmers with libraries such as OpenGL . In our monoplotter, we will take advantage of this technology for the efficiency of its 3D rendering functions, which are in fact perspective functions used to project 3D models on the monitor.

Typically, a 3D model (consisting of primitive geometries) and, if needed, a texture are stored in the GPU. In our monoplotter, the geometries are created from the triangulated DEM whereas the texture is the orthoimage. They are subsequently displayed for a virtual camera pose whose image plane is the monitor. The GPU implements the perspective projection (in a linear form) to project the 3D model and its texture on the 2D computer monitor. As illustrated in Figure 4.3, multiple 3D objects can be projected onto the same monitor location, the GPU ensures then that only the object the closest to the virtual camera (having the smallest depth) appears on the monitor.

The GPU proves to be useful in the monoplotter for the fast rendering of 3D images created from the DEM and the orthoimage. The 3D rendering is used, for instance, to allow the user to browse the virtual environment and approximate the orientation of a picture. The user can also digitize GCP directly in the virtual environment. However, we will also use the GPU implementation of the perspective projection indirectly for the *map-to-image* and *image-to-map* functions discussed below.

4.3.4 Map-to-image functions

For a fixed camera pose, the *map-to-image* transformation aims at projecting 3D data in the image plane. Hence, each 3D point feature is projected in the image plan according to the camera parameters computed during the pose estimation. In our implementation, the GPU renders a synthetic image which has the same exterior and interior orientation as the original image. The *map-to-image* functions are used in two ways:

- For a given pose, a synthetic image is generated from the DEM and the orthoimage. The synthetic image is generated on the GPU as detailed earlier. Each pixel is then associated with a 3D world coordinate.
- GIS vectors are projected onto the image to add a semantic layer as in augmented reality.

The management of the occlusion is inherent to the rendering on the GPU. In the projection of vector objects, the difficulty is to avoid the display of the objects (or part of objects) hidden behind the relief. As illustrated in the Figure 4.4, P'_0 and P'_1 are projected on a same image location but P'_1 is

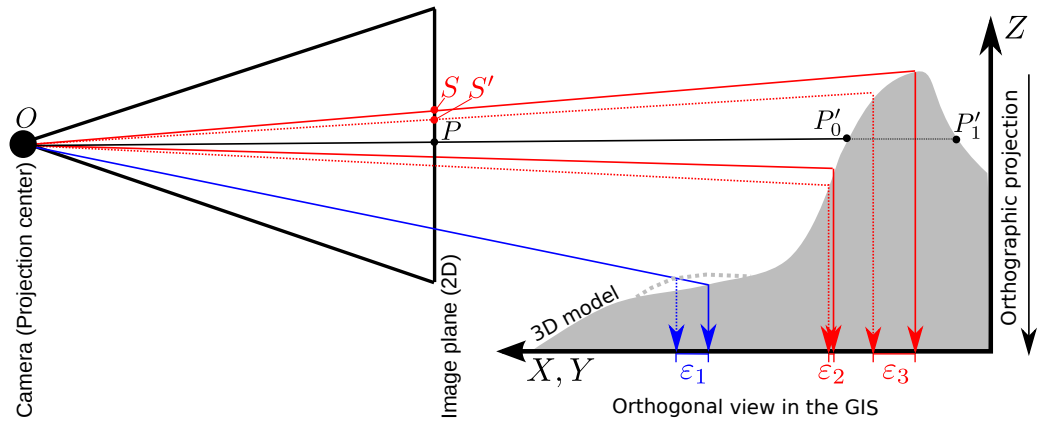


Figure 4.4 – Geometry of the monoplottling as defined by Kraus (2007). O is the center of projection and S is a pixel located on the image plane. P'_0 and P'_1 are possible intersection of the line OP with the DEM, only P'_0 , closer to the camera, is visible. In blue, the error ε_1 engendered by a DEM error. In red, the error engendered by an orientation inaccuracy or an inaccurate localization of S . The impact of the error is increased in regions where the relief makes an acute angle with OS : $\varepsilon_2 < \varepsilon_3$ even if they are generated by a same angular inaccuracy.

occluded. Assuming that P'_0 and P'_1 are the world coordinates of a vector object to be projected in the image, if the point is located on the 3D surface, it is located at the same distance as the 3D surface (or closer) and it is visible. If it is further, it is occluded.

4.3.5 Image-to-map functions

The *image-to-map* functions are going in the opposite direction, they project the image pixels on the DEM to get their 3D coordinates. These functions are the ones which are called monoplottling in the literature (Kraus, 2007).

"Single image analytical analysis of curved object surfaces is known as monoplottling. Monoplottling is used for the analysis of both aerial and terrestrial photographs. The accuracy of the analysis primarily depends on the intersection angle of the ray OS (editor's note: the line defined by the projection center and a pixel) with the surface profile in the vertical plane. When the intersecting rays are very oblique, large XY coordinate errors are generated where there are:

- *Minor height errors in the DSM*
- *Small errors in the interior and exterior orientation*
- *Small measurement errors in the image coordinates of point S*

In general, therefore, monoplottling cannot replace stereoanalysis." (Kraus, 2007). This statement is illustrated in Figure 4.4.

In accordance with this definition, monoplottling is used to create geographic data from a single image or to measure objects directly from the image. Typically, an operator delineates an object in

the picture and this object is simultaneously transformed into geographic vector layer. With the same process, an image is ortho-rectified by projecting every pixel on the DEM and keeping only the planar coordinates. As it can be seen in Figure 4.4, the perspective transform is not invertible: an image pixel P can have two correspondences P'_0 and P'_1 in the 3D space. In this case, P'_1 is hidden, and only P'_0 appears in the image. To cope with this problem, authors in Fluehler et al. (2005) use single-ray backprojection (Mikhail et al., 2001) to compute the coordinate of the closest location where OS intersects the DEM (ray tracing in computer graphics). In short, the vector OS is incrementally extended until it crosses the 3D model. This intersection defines the 3D coordinate of the pixel. This function can be implemented efficiently, but still if the 3D coordinates of a high-resolution image associated with a high-resolution DEM need to be computed, the task is time-consuming. Moreover, the projection of the 3D surface in the image and the computation of the corresponding world coordinate is already implemented in the GPU. Following the definition of the projection of world coordinate in an image, discussed in Equation 3.4, the computation of the depth is based on the coordinate \mathbf{x} in the camera system, which can be transformed back to the world coordinates \mathbf{X} by inverting the rigid body transformation. Thus, our implementation is directly derived from the GPU functions.

The world coordinates computed in this way have several implication in the monoplotter software:

- A top orthogonal view of the oblique image textured on the DEM generates an orthorectification of the oblique image (Figure 4.1, step 4).
- Vectors drawn on the image are simultaneously measured and mapped in the GIS (Figure 4.1, step 6, blue line).
- X, Y coordinates of the pixels on the image border can be merged to draw the footprint polygon of the image (Figure 4.1, step 6, yellow line).

4.3.6 Integration with an open source GIS

The integration of Pic2Map in an open source GIS and especially in QGIS has several advantages. First, QGIS provides the Graphic User Interface (GUI) and the interaction with the GIS libraries (i.e. coordinate system management, GIS format reading and writing etc.). Namely, most of the libraries required for the development of a GIS-based software are already pre-existent in the QGIS core and only the functions particular to pic2map have to be implemented. Thus, the time spent for the implementation and maintenance is greatly reduced. Second, the script is written in Python, a high-level programming language, which is more and more used in the academia for teaching and research and also widely used in the GIS community. The code of Pic2Map is then re-usable by these communities to develop new functionalities. Moreover, Python scripts need only to be copied to be installed and thus the plugin is OS-independent. The integration within QGIS is also convenient for the users since they are already familiar with the interface. Snapshots of the application are presented in Appendix A.

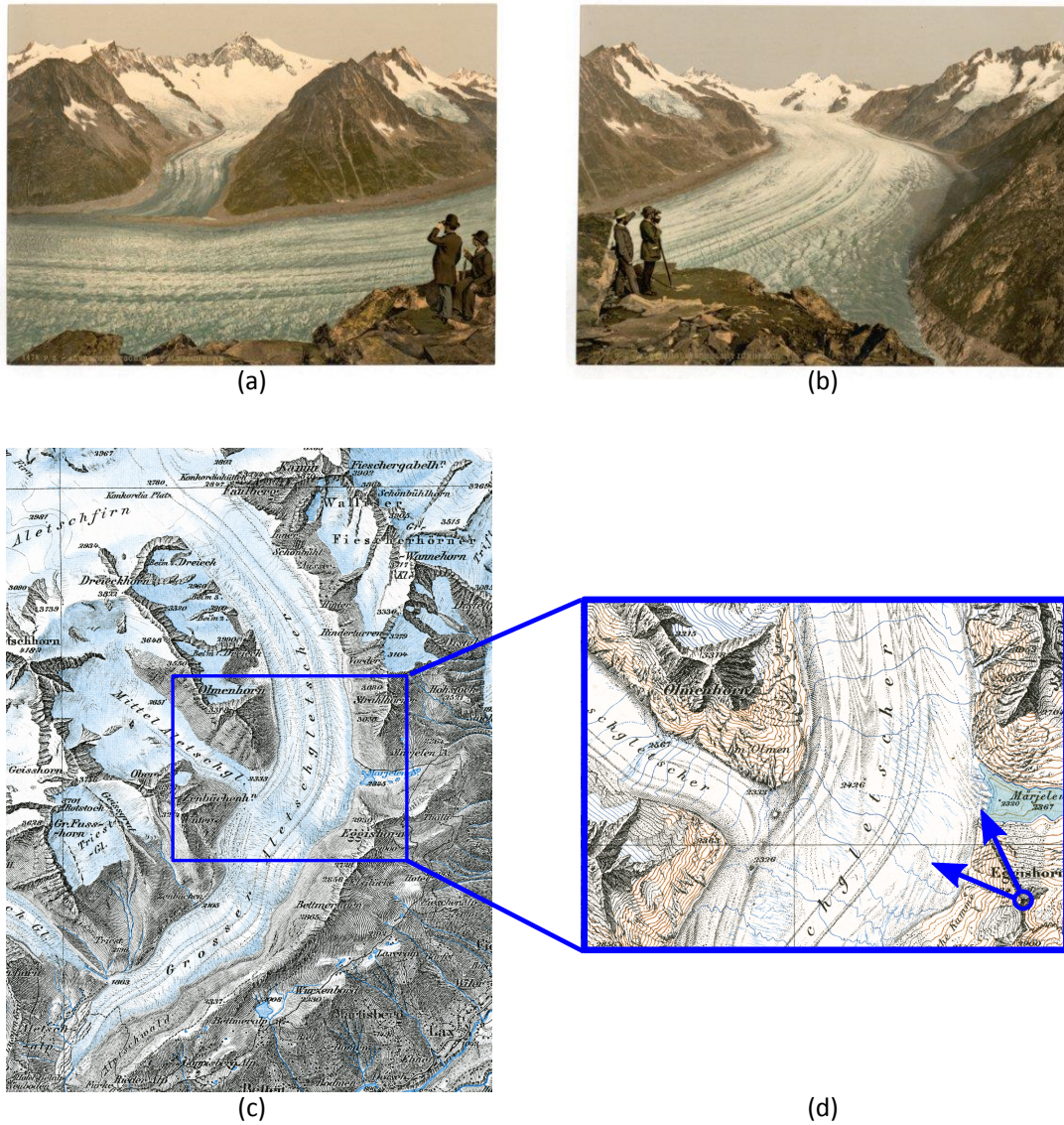


Figure 4.5 – (a) Left and (b) Right image of the Aletsch glacier. The area is represented in the Dufour map (c) and Siegfried map (d). (*Library of Congress - Swisstopo*)

4.4 Illustration of the functions and case study (Aletsch)

In this section, we will present one typical scenario of monoplotter application. The case study illustrates the georeferencing of historical photographs for land change evaluation. The goal of this case study is to illustrate the monoplotter functions rather than measuring precisely environmental parameters. It is illustrated with screenshots of the plugin and QGIS interfaces.

We study two images, dated from a period between 1890 and 1900, representing the Aletsch glacier (Figure 4.5 (a-b)). These images are thus a vestige of a period when the glacier level was higher, but also when aerial images were not yet produced. They are complementary to other sources, such as the ancient maps available in this area: Dufour (published between 1845 and 1865) and Siegfried (published over the period 1870 to 1926), see Figure 4.5 (c-d). Specifically, these images have a higher spatial resolution and are a direct record of the landscape state (almost free of interpretation) whereas the maps are drawn from measurement and then classified in topographic classes (glacier, cliffs etc.) having a fuzzy delineation.

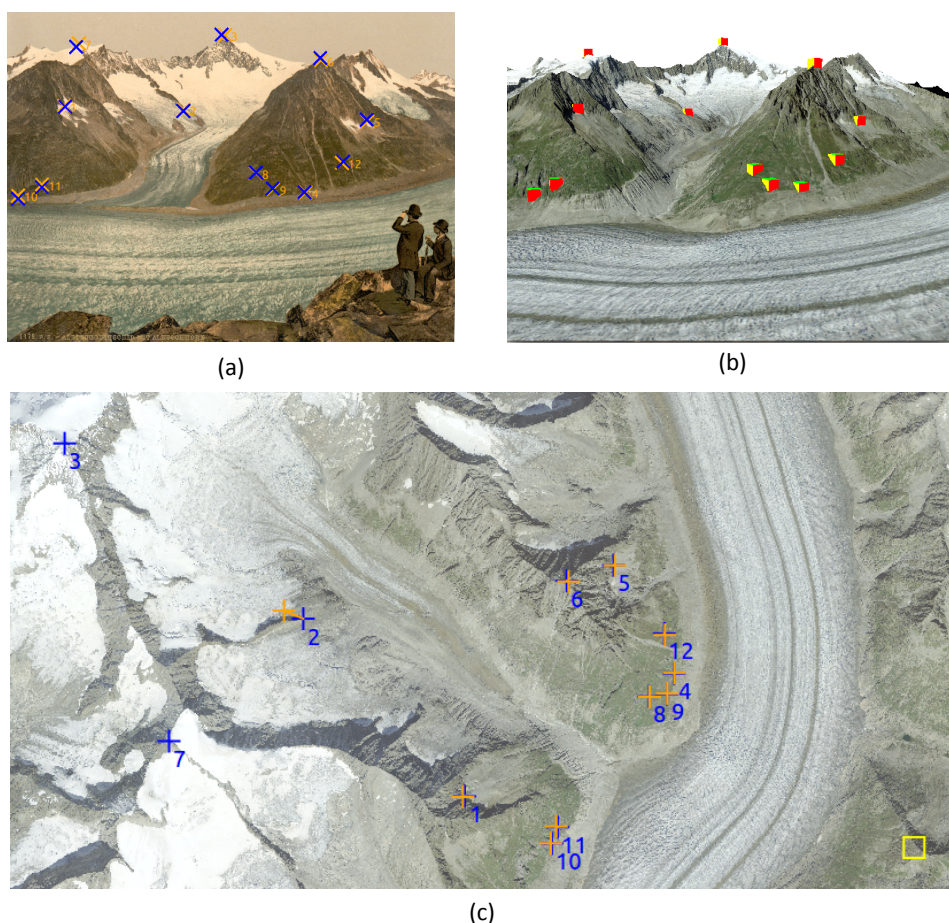


Figure 4.6 – The left picture is georeferenced with 12 GCP represented as blue crosses in the original image (a), cubes in the 3D view (b) and blues crosses in the plan (QGIS) (c). The orange crosses shows the reprojection error and the yellow frame is the computed camera location.

4.4. Illustration of the functions and case study (Aletsch)

GCP ID	Left image Reproj. error [m]	Right image Reproj. error [m]
1	18	12
2	188	33
3	Nan	57
4	6	52
5	16	62
6	28	8
7	Nan	6
8	6	2
9	7	1102
10	22	1181
11	20	37
12	15	19

Table 4.1 – GCP errors measured in meter for both pictures. Errors of GCP close to the glacier are highlighted.

These two black-and-white images were shot from the top of the Eggishorn Mountain, encircled in Figure 4.5. They were then manually colored and are thus called photochroms (and this is why they are only almost free of interpretation). They are downloaded from the "*Library of Congress*" and belong to the collection "*View of Switzerland*". Both images are registered with 12 GCP (see Figures 4.6(a) and 4.7(a)). The digitization of GCP is particularly difficult in these images for which the manual colorization perturbs the detection of similar objects. They are thus not a good example of the best georeferencing accuracy which can be expected with a monoplottter. More GCP than required for pose estimation are digitized and we can thus estimate an error on each GCP. The plugin provides two error measures for each GCP. First, the projection error is the distance between a 2D control point and the projection of the corresponding 3D control point in the picture. This error is the distance minimized during the pose estimation. However, it is measured in pixel and is thus difficult to evaluate. The second error (presented in Table 4.1) is measured in meters between the projection of a 2D control point on the DEM and the corresponding 3D control point. *Nan* values are associated to 2D control points which are projected in the sky and thus do not cross the DEM. The largest errors are generally due to 2D control points chosen on a peak or a ridge and projected in their background (they have a small shift in the image, but a large shift on the map). Generally, GCP close to the glacier limit (in bold) have small errors. An exception is GCP no 5 in the right image, which is a good illustration of a large error due to a bad configuration of the relief and the picture. These errors represent the accuracy which can be expected for geographic data extracted from these images.

On this basis, we want to evaluate the glacier variation between 1900 and today. In this mountainous area, the DEM is not supposed to vary drastically except on the glacier itself, which has melted. Hence, the DEM, recording the current glacier level, is appropriate for this task. For the comparison of the 1900 and current state, we use the ortho-rectified historic images and the current glacier contours provided by the Swiss topographic agency (state 2013). Several distances are represented in Figure 4.8: red ones are digitized from the right image and green ones from the left image. The values labelled on the distances are respectively the horizontal and the vertical differences. The glacier limits extracted from both images are coherent. However, red and green lines diverge in the border of the left image. Indeed, the non-modelled distortions have more impact on the exterior of the images. Moreover, the geometric configuration of the right image with respect to the relief is prone to errors in this area.

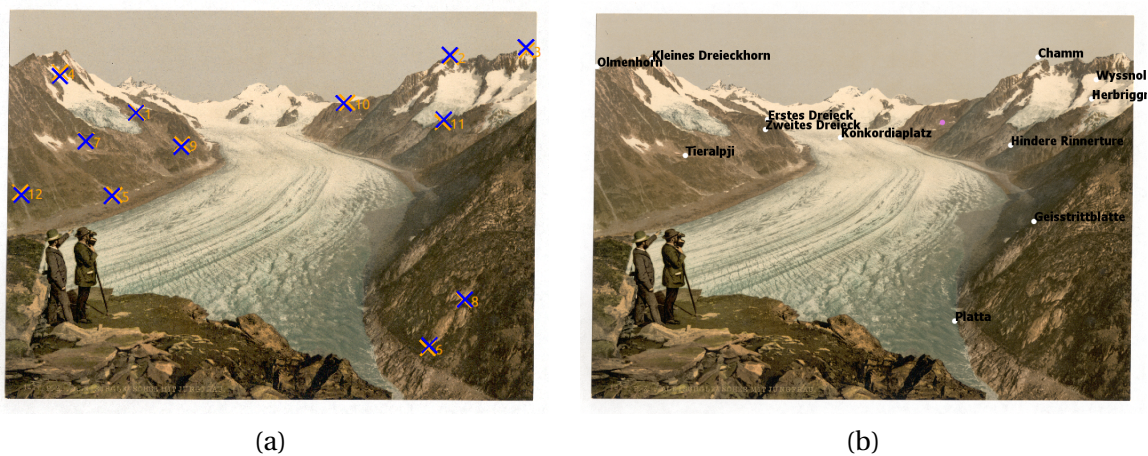


Figure 4.7 – a) GCP of the picture looking in the right. (b) After the pose estimation an image is augmented with GIS data, here the toponyms.

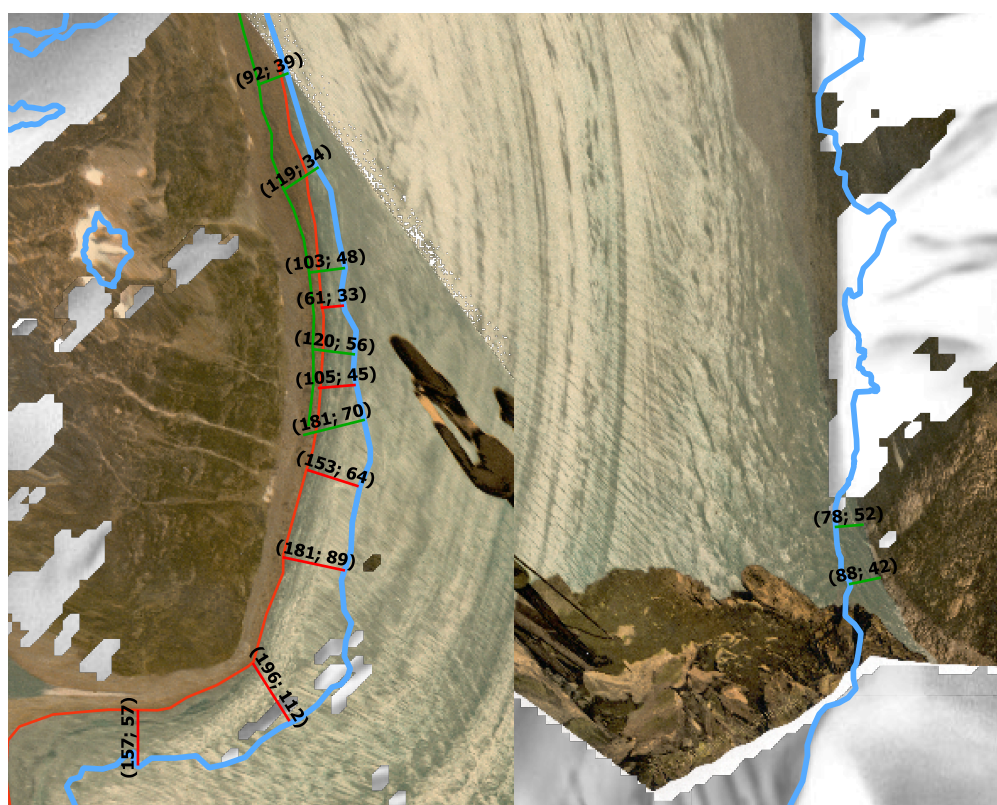


Figure 4.8 – Distances between the glacier limit determined with the ortho-rectified historical images (red and green lines) and the current state represented by the blue lines. The first number is the horizontal difference, the second is the vertical difference. Red lines are drawn from the left image, green ones with the right image.

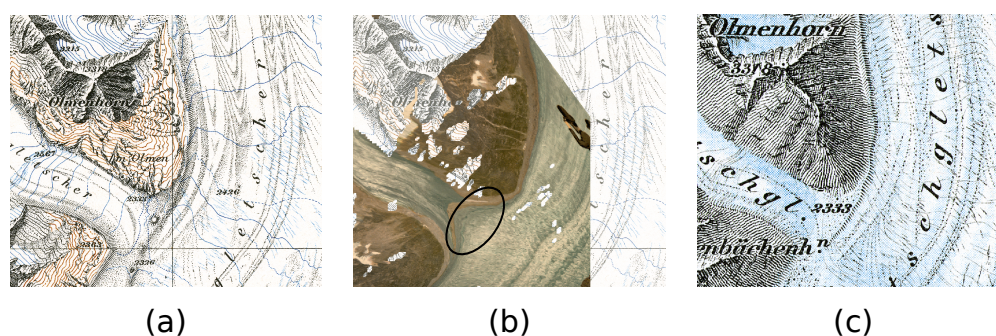


Figure 4.9 – Comparison of historical maps and the ortho-rectified picture. (a) Siegfried, (b) Ortho-rectified image, (c) Dufour. (*Swisstopo*)

The analysis of the glacier evolution is beyond the goal of this case study, but in the front of the left image (where the geometric configuration is the best adapted to monoplotting) the measured lateral retreat is between 200m and 150m whereas the height diminution is 110m and 57m. Beyond the analysis of the glacier evolution, this case study confirms the pertinence of using historical image in conjunction with historical maps. Figure 4.9 shows the resolution and representation difference between the maps and the picture. In the maps, it would be difficult to delineate precisely the glacier limit whereas it is easy to find in the picture. However, it should be recalled that the pictures were manually colored and thus already subject to an interpretation. Finally, in some regions, the errors resulting from a bad geometric configuration, generate huge distortions. These errors can be seen at the junction of the glacier (ellipse). In such regions, the maps are more trustful than the information extracted from the pictures.

4.5 Discussion

High oblique (terrestrial) images are complementary to more traditional data (orthoimages and maps) to assess historical state of a landscape, but also to evaluate changes at finer temporal and spatial resolution for instance with fixed cameras. They are thus valuable to study landscape evolution, but are mainly studied in the academia. One of the reasons why they are not used more widely is that there are no easily accessible monoplotters. In this chapter, we presented a software meant to the wider public. To this end, we mainly act on two levers: its insertion in an open source GIS (free and reusable) and 3D rendering to facilitate the pose estimation from GCP. This plugin was already used by bachelor students during a GIS introduction course. Only based on the documentation of the software and limited help of the assistants, the students were able to measure the evolution of environmental parameters from pictures. Hence, by providing QGIS with monoplotting possibilities, we can provide solutions for:

- **Rapid mapping:** under emergency constraints, for instance, avalanche or landslide, a photogrammetric campaign may be impossible: the weather could prevent flying with manned or unmanned vehicles and the processing time can be too long. Under these conditions, a monoplottter can be useful to rapidly produce maps.
- **Tourist images augmentation:** Every walker is yet equipped with GPS and camera. Websites such as CampToCamp 🏕️ dedicated to the sharing of mountaineering itineraries show us that people like to share pictures and tracks. The projection of geographic vector layer in their pictures can

be used to enrich pictures with a GPS track or other geographic information (topographic names, walking routes etc.).

- Civil and rural engineering project illustration: A 3D model of a work of engineering can be added to real images.
- UAV or kite-based images georeferencing: UAV are yet in the consumer market, but the processing via SfM can be over-sized for general public needs.

In these applications, the involvement of the operator is not a main limitation, especially if the interface is user-friendly and allows non-specialists to detect GCP rapidly. The main limit of the monoplotting is its accuracy. Even if the user spent ages for the digitization of the GCP, errors can be expected on each of the three points aforementioned by Kraus (2007):

- An inaccurate exterior orientation is computed from GCP (the operators provides correspondences which are not exact),
- The interior orientation is unknown or inaccurately evaluated,
- Often only coarse resolution DEM are available and the landscape may have evolved between the acquisition of the picture and the acquisition of the DEM. The height accuracy perturbs both the accuracy of the GCP (and thus the pose estimation) and the georeferencing of the pixels.

Hence, with monophotogrammetry, it is difficult to reach the accuracy of the aero-triangulation. However, for historical pictures, such as those presented in this chapter, it is often the only way to extract geographic information and thus quantifiable data. In other scenarios, (rapid mapping, augmented images) the accuracy is less critical. Nevertheless, it is important to be aware of these limits and assess the accuracy of the results, for instance with the error measured on the GCP. Thus, an additional functionality to be added in the monoplotter is the detection of the regions prone to errors due to a bad geometric configuration.

The pose estimation of a picture with GCP is generally time-consuming. With the implementation of a 3D browser, we already have reduced the time spent by the operator for the digitization. However, there is still room for the improvement of the user-friendliness, for instance by adding some automated functions. First, for small collections of images, image matching (with a feature descriptor like SIFT) can be used to infer a rough pose from a georeferenced picture (to be discussed in Section 5.3). For instance, in our example the images have a small overlap. The first georeferenced image could be then used to extract automatically some GCP for the second image. Second, it is often easier to detect similar linear features (horizon, rivers, roads etc.) than point-wise features. We will present a method to match and use linear features for the pose estimation in Section 5.1. Finally, the automatic detection of GCP could replace the user at least for the initial rough pose estimation: we will present a method for this task in Section 5.2.

From our experiences with the monoplotter, we can already discuss the expected accuracy of these automatic methods to estimate the pose of an image. With user-detected GCP, the best accuracy is reached with accurate GCP detected on stable and remarkable objects (such as the corner of a building) and well distributed in the image. It seems difficult to detect automatically such correspondences: automatic methods are generally noisy (false as well as inaccurate correspondences are detected), orthoimages are very distorted and the interpretation capacity of the operator is required to identify similar objects (see Figure 4.10 for an example of GCP on a building). Hence, apparently it will be

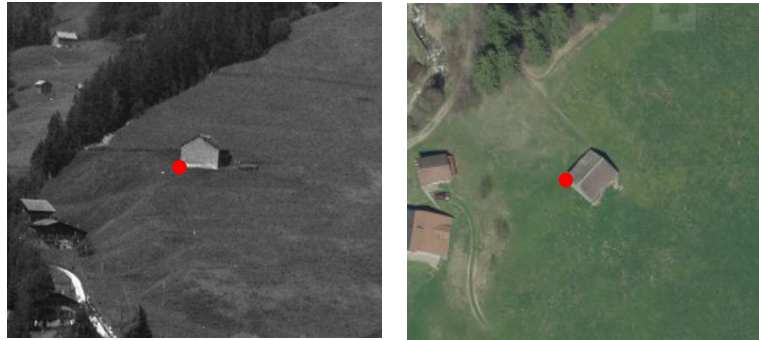


Figure 4.10 – Illustration of an accurate 2D-3D correspondence. Left: query image. Right: orthoimage. (*Archives de la construction Moderne - ACM - EPFL, Swisstopo*)

difficult to replace the user for an accurate pose estimation. Thus, if automatic methods could align the query image with a 3D synthetic view, we could provide a good estimate of the exterior orientation. This estimate can be further refined by a user if accurate data have to be generated from the image.

The Pic2Map-plugin is available in the repository of the QGIS plugins. It still needs some efforts to incorporate every function of the software presented in Produit and Tuia (2012) (i.e. EPnP to compute a pose from four correspondences) and to improve its stability.

4.6 Conclusion

We think that this software will foster the acquisition of georeferenced raster data by the crowd. Indeed, in comparison, everyone is able to produce geographic vector layers thanks to consumers GPS and facilities offered by web-GIS to digitize vector data. For example, the crowd has created the world-wide database OpenStreetMap. Concerning raster data, it is still difficult (or reserved to professional) to convert images to geographic rasters. However, everyone possesses sensors (smartphones and camera) and UAV have reached the consumer market. Complementary to SfM applications, monoplotters could be used to sense and record our environment. Hence, a final goal would be the insertion of the monoplotters in a web-based virtual globe. For instance, it could be very valuable for archives manager to crowdsource the georeferencing of pictures just like they already have crowdsourced the georeferencing of maps (Kowal and Pridal, 2012).

5 Registration of a landscape image with a landscape model

As illustrated in Figure 5.1, the georeferencing of an oblique image with our monoplottter can be divided into three steps. First, a rough orientation of the image is found within a virtual globe (this step is skipped if the image has a geotag or is acquired with a GPS-enabled camera). Second, 2D-3D correspondences are found in the image and in the synthetic image created from the rendering of a DEM textured with an orthoimage. Finally, precise GCP are detected in the image and in the orthoimage.

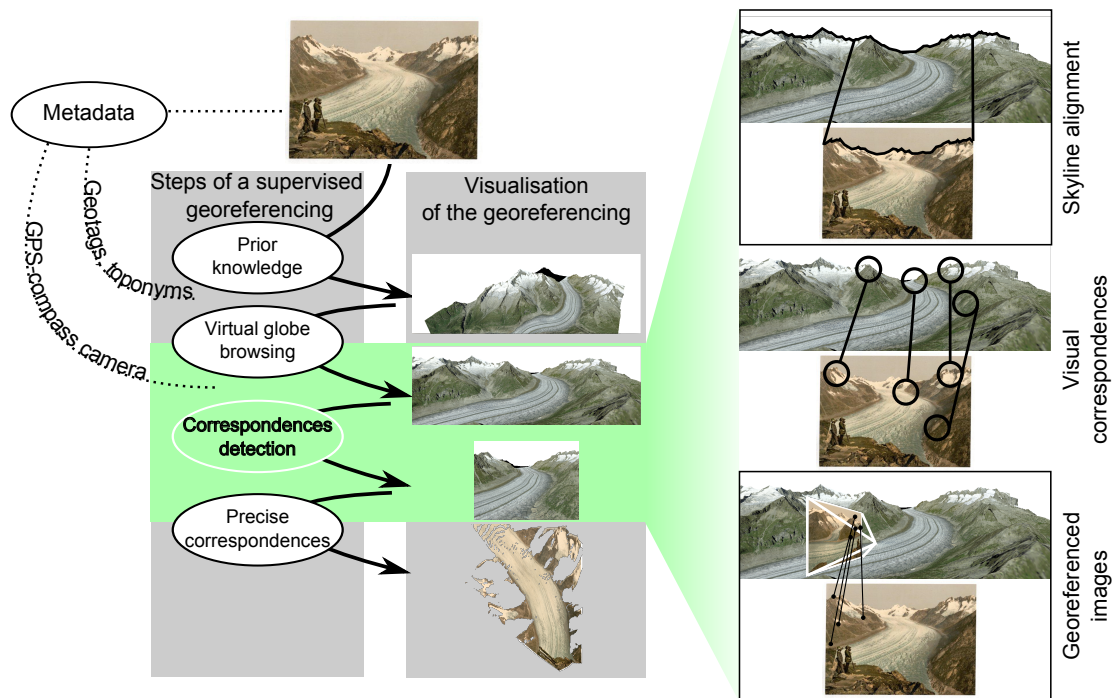


Figure 5.1 – Left: illustration of the registration steps. The contributions discussed in this Chapter are highlighted in green. First, we will provide a method to align the skyline. Second we will determine corresponding regions of the images. Finally, we will use images already georeferenced or images shared on the web as the reference.

In this Chapter, we want to ease the second step (and thus the third indirectly) by detecting automatically some correspondences between the image and the 3D landscape model. We will discuss three contributions. First, in Section 5.1, we will consider the skyline as a source of 2D-3D correspondences. Here, we propose a semi-automatic detection of the skyline which is then matched with the synthetic skyline using a novel algorithm based on Dynamic Time Warping (DTW). Second, in Section 5.2, we propose a method based on Histogram of Oriented Gradient (HOG) to match patches of the query image with a realistic synthetic image. These methods are validated with a set of images acquired by the author and oriented with a GPS and an IMU. Specifically, we will assess how these methods can retrieve the 3DOF orientation of the camera (the location is known) or the 6DOF orientation of the camera (a prior on the camera location and camera azimuth are known).

Finally, in the task of georeferencing a database of images, we can also consider the detection of tie-points between overlapping images. This solution is valuable to use as reference images shared on the web and located with a GPS or images already georeferenced. The landscape model is used here to avoid drifts and determine the world coordinates of key-points detected in the reference images (georeferenced manually or with one of the methods discussed above). In this method, each image is georeferenced one after one. This strategy is chosen to be compatible with evolving databases (new images are inserted in the collection) and with supervised georeferencing (which has to be considered for the most challenging collections). In Section 5.3, we will present a case study constituted of a real collection of images, and illustrating how the methods presented in this Chapter can interact.

5.1 Camera orientation with DTW-based Skyline alignment

5.1.1 Introduction

In the previous chapter, we presented a monoplottter which computes the orientation of a camera with GCP provided by the user. One of the limits of the monoplottter is that the digitization of the GCP is time-demanding. The expertise of an operator is compulsory for a precise georeferencing. Our experiences of manual georeferencing taught us that the 3D rendering is helpful for the operator. For instance, the skyline is the first landmark used by the user to align the picture with the virtual landscape. This remarkable feature can be detected and matched automatically and thus reduce the time spent by the operator for the GCP digitization. In this chapter, we want to assess how the skyline matching can be used to compute the orientation and location of a picture. After a review of the literature related to the georeferencing of cameras using the skyline, we propose a new method for the precise matching and alignment of the skyline. Namely, we apply Dynamic Time Warping, an algorithm originally used to match time sequences. The DTW-based horizon matching was presented in Produit et al. (2014b) as a part of a larger processing chain. In this chapter, we explain it in more detail and also analyze its performance on a set of ground truth data.

5.1.2 Review

During the 1990s, the robotic community was very interested in the localization of vehicles in unknown terrains based on the availability of maps or 3D models. Indeed, at this time, the GPS was not open to civil applications (we had to wait the year 2000). The only reference data covering the entire earth, and even other planets, was the DEM. Thus, the use of DEM for the georeferencing of robots, and more generally of cameras mounted on any vehicle, could have opened large opportunities for the navigation on earth and other planets.

Some solutions proposed during this period are based on specific scenarios. For instance Talluri and Aggarwal (1992) used a rotating camera and a compass to take pictures in the four cardinal directions. Next, synthetic skylines generated from possible locations were compared with the skylines extracted from the pictures. The best location was detected by measuring the similarity between the reference horizon and the observed horizon using a sum of squared errors. Quite similarly, Stein and Medioni (1992) generate a 360° panorama with a camera having a compass. The reference silhouettes were encoded with polygonal approximations, subdivided into fragments and stored in a database. The fragments were then matched to the observed horizon with an error measure taking into account the direction, the shape of the angles forming peaks and the location and number of vertex constituting a fragment. Some years later, Cozman and Krotkov (1996) applied horizon matching with the goal of localizing a lunar rover. They implemented a mountain detector, which approximates the distance of a peak with the roughness of the silhouette. The localization was retrieved by the relative angular relations between the peaks detected in the pictures and the ones found in the DEM. Later, Naval Jr et al. (1997) also detected peaks in the skyline and in the DEM. They used three peaks as 2D-3D correspondences to estimate the camera location. To avoid the test of every triplet, they added several constraints (proximity, visibility) and thus reduced the number of combination to be tested. Finally, the projected horizon that best matched the image horizon was selected as the best pose.

Afterward, with the availability of GPS, the focus slightly moved from a totally unknown pose to a known location. The goal is no more to locate a vehicle in 2D, but to get the viewing direction of a camera if its location is measured with a GPS. These approaches are related to augmented reality

for which one wants to compute an orientation sufficiently accurate to overlap geographic data with an image. Behringer (1999) follows this setup. He extracts the skyline extrema in the image and in a panorama. Hypothetical matches of the extrema are used to compute the camera orientation and the best pose is selected as the one minimizing the distance between the horizon silhouettes. In this setup, the roll is always set to zero, forcing the camera to be horizontal. Baboud et al. (2011) share a similar problematic but focus on the overlay of mountain names in tourist pictures shot from a given location. They do not use only the horizon, but every morphological edge occurring between successive overlays of relief. Their matching is done in a spherical domain and is twofold. First, a cross-correlation detects possible orientations. Second, an error measure was developed to evaluate how well the edge features of the image match the morphological edges. This method has the advantage to not require skyline detection, but only edge detection.

Lately, the motivations behind the use of the horizon as the reference changed again with the apparition of UAV. Woo et al. (2007) use horizon to determine the position of a UAV if GPS and INS are disabled. Their system uses a coarse pose to limit the possible correspondences between skyline peaks and DEM peaks. Similar peaks detected in successive images shot by the UAV are used to reconstruct their 3D. Possible correspondences are then used in RANSAC to compute an affine transformation between the real peaks and the reconstructed ones. For the fine pose estimation, the image horizon and the synthetic horizon are matched with Monte Carlo Markov Chain using as the scoring function the number of overlapping horizon pixels. Finally, in Baatz et al. (2012b), the authors build offline a database of quantized contourlets of skyline (rather than storing every fragment of skyline, they are classified in a subset of typical contourlets). The database covers the entire surface of Switzerland in each location separated by 100m. The contourlets are then inserted in a bag-of-feature (for each image the distribution of the contourlets is stored). The matching is then done by retrieving the database skylines which have a similar distribution and relative position of the contourlets. Next, the best solutions are aligned with ICP (Iterative Closest Points: the closest points are used as correspondences to estimate the transformation, the process is iterated) to estimate the silhouettes similarity. Since ICP is known to converge only if the initial alignment is good, they test several azimuths. With this system, 88% of the query images were located within 1km of their real location. With this promising performance, they provide the most suitable method for large-scale search available currently.

All these methods (except the one proposed by Baboud et al. (2011)) suppose that the horizon silhouette is extracted directly from the picture. This extraction has to be robust against the variation of weather and illumination, obstruction (clouds), color and texture of the ground (snow). It is thus not trivial. There are solutions proposed in the literature based on classification of pixels features, on the detection of a high luminosity band occurring at the interface between the ground and the sky, or based on edge detection. A recent review can be found in Boroujeni et al. (2012). However, in the methods listed in our review, the user is often asked to interact with the algorithm to add more robustness to the detection.

The Figure 5.2 summarizes the existing methods for horizon matching. First, the skyline is detected in the picture and in a rendered view of the DEM. These 3D views are either generated online (if few synthetic views have to be created, for instance if priors of the location are available) or offline and stored in a database. Typically, 360° panoramas assuming a horizontal camera are generated for a grid of locations (1). As other authors, we provide a solution for the geometric comparison. They are three competitive strategies at this level:

- 2A** The first is proposed in Baboud et al. (2011): every possible orientation of the camera is compared with the synthetic model. This computationally intensive method is only adapted

5.1. Camera orientation with DTW-based Skyline alignment

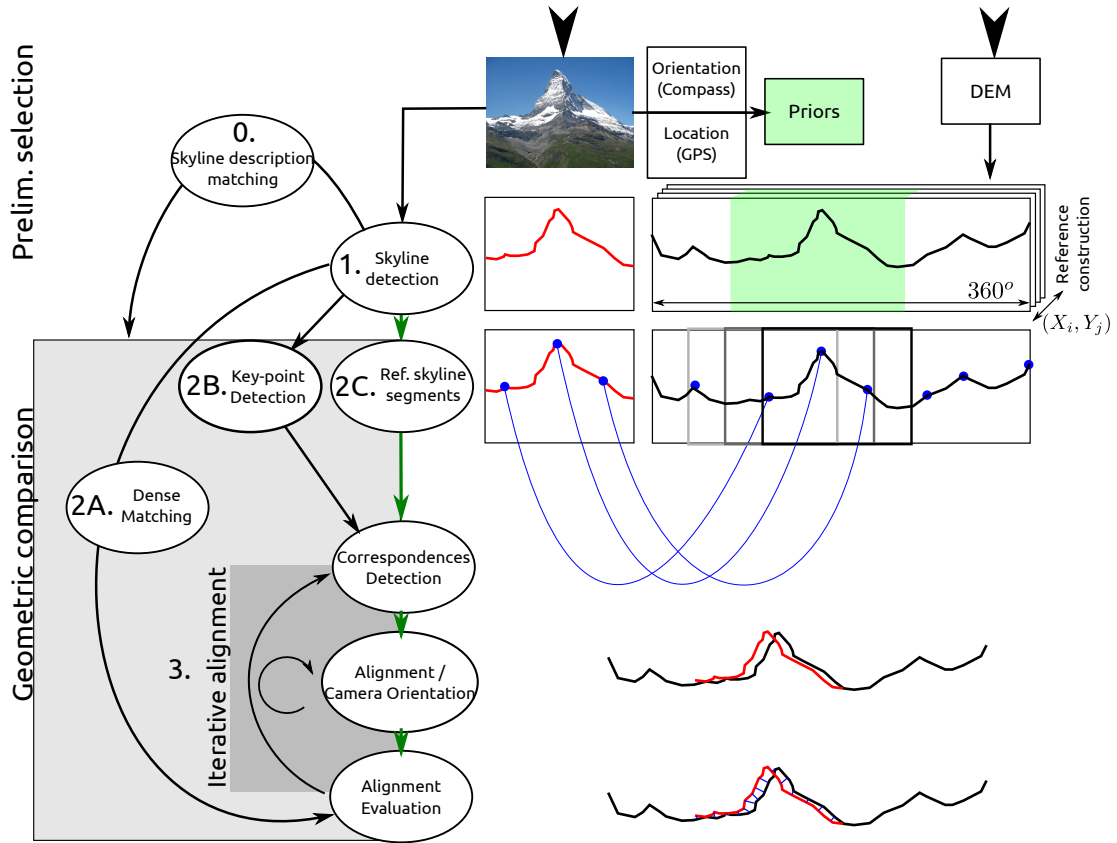


Figure 5.2 – Summary of the methods found in the literature to retrieve the pose of an image with skyline. Typically, synthetic skylines are generated online or offline (black skyline). If the image has a pose prior, the query skyline (red) can be matched with a subset of the database of reference skylines. Our strategy is highlighted with the green arrows.

to pictures associated with an accurate location. It is not adapted to deformations (imperfect location or imperfect focal estimation).

2B A second strategy is the extraction of matchable and remarkable key-points. Methods proposed in the literature include peaks, extrema.

2C The last method is the division of the entire 360° skyline in overlapping segments.

For the strategies **2B** and **2C**, the next step is the extraction of correspondences to estimates the parameters of the transformation which aligns the skylines (**3**). If key-points were detected (**2B**), RANSAC integrates the correspondences detection, the camera orientation and the evaluation of the alignment. If reference segments must be compared with the query skyline (**2C**), ICP can perform the iterative alignment using the nearest neighbor as correspondences. In every strategy, the parameters associated with the best alignment provide the camera pose.

Baatz et al. (2012a) search for the image location at large scale. Hence, they add a preliminary selection to estimate a set of potential locations and thus avoid the geometric comparison of every 360° panorama within their database of Swiss skylines. In this sense, this selection is like a prior which reduces the location possibilities. At this level, they perform the iterative alignment of skyline segment

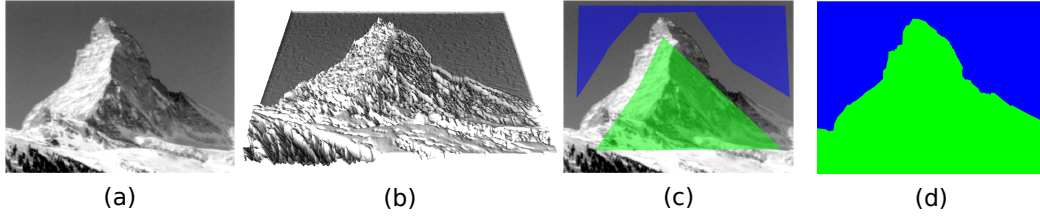


Figure 5.3 – Illustration of the watershed segmentation: (a) initial image (smoothed for the purpose of the example), (b) 3D visualization of the pixel heights, (c) initial regions, (d) resulting segmentation.

with ICP, a step which could be improved. Indeed, with their approach only 55% of the images are located in a 200m radius of the real location meaning that there is room for the improvement of the local alignment. Specifically, ICP requires a good initial alignment of the silhouettes to converge and cannot recover deformations. In these conditions, the goal is to develop a method, which compared to ICP, i) converges from worse initial alignment (and thus reduces the number of segment to be tested, ii) is less sensitive to silhouette deformations (such as induced by rendering, by a shifted location or by a wrong estimate of the focal) and iii) remains simple and fast. To avoid the detection of extrema, which is adapted only for skyline having multiple marked peaks and skylines perfectly delineated (difficult to obtain even with supervised detection), we also propose to match skyline segments, but with a more evolved algorithm than ICP.

5.1.3 Proposed method

The method that we present is a dense matching and illustrated with the green arrows in Figure 5.2. First, the skyline is delineated in the image with a watershed segmentation. Second, the reference skyline is generated from a 3D rendering of the DEM. Third, the skylines are matched with DTW. DTW provides correspondences and a measure of the similarity of the skyline. Fourth, the 2D-3D correspondences are used to align the query skyline with the reference skyline. The third and fourth steps are iterated. Fifth, the camera orientation which minimizes the alignment error is selected. If there is no prior about the camera azimuth, the 360° panorama is discretized and each subset is tested.

5.1.3.1 Skyline detection

To extract the horizon from the query images, we simply propose an interactive watershed segmentation of the sky and the ground. The name of this algorithm is familiar to GIS specialists. Indeed, the pixel values are treated as the height of a landscape and the goal of a watershed segmentation is to delineate the regions which flow into a same river. In image processing, initial regions (which can be seen as lakes) are provided to the algorithm, each pixel of the image is then attributed to one of the initial regions. The process is illustrated in Figure 5.3.

As stated in the review, sky detection is not trivial. Hence, we did not investigate this research area and focus only on the horizon matching. Moreover, the sky detection method proposed does not require a large investment of the operator who can achieve the task quickly (circa 20 seconds).

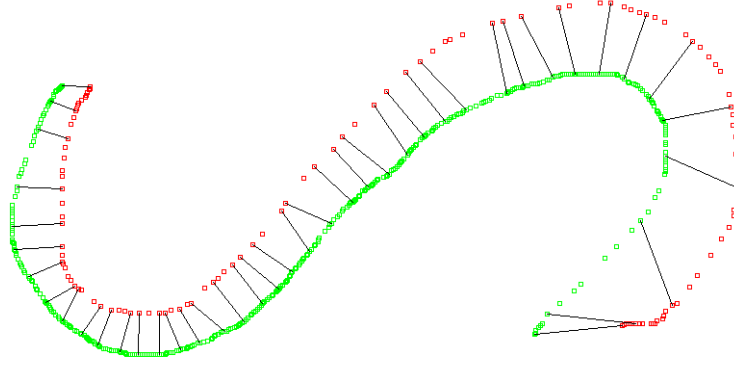


Figure 5.4 – DTW is used to estimate corresponding points between two GPS tracks. (CC, Wikimedia, author: Jsoler)

5.1.3.2 Dynamic Time Warping

Horizon matching is similar to the problem of sequences matching. For the matching of sequences, two sets of temporal measurement have to be compared and matched. We will define $x^Q = \{x_0^Q \dots x_i^Q \dots x_n^Q\} = f^Q(t_i)$ as the query sequence and $x^R = \{x_0^R \dots x_j^R \dots x_m^R\} = f^R(t_j)$ as the reference sequence. Typically, the sequences are deformed due to acceleration and deceleration, such as the illustration of GPS tracks matching in Figure 5.4. There, the difference of speed explain the difference of point density, in speeches the difference is induced by the speech rate. The DTW algorithm is used i) to find the corresponding regions in the sequences and ii) to provide a measure of their similarity.

A dynamic programming-based method is applied to this problem. Dynamic programming solves complex problems by assuming that they can be divided into smaller and simpler sub-problems. With this strategy, the smaller problems are solved only once and thus the resolution of the complex problem is more efficient. Dynamic Time Warping (Berndt and Clifford, 1994) was developed to solve time-sequence matching as a shortest-path problem. Each potential pair x_i^Q and x_j^R is associated with a distance (or cost) $c(x_i^Q, x_j^R)$, which in the example of Figure 5.4 could be the Euclidean distance between two points. The goal is to find the matching, which minimizes the sum of the costs of each pair. This matching is then a succession of pairs (i, j) forming a path with two constraints. First, each feature must appear once, that is each feature has a correspondence in the other set. Second, the order of the sequence (the time) has to be preserved. Hence, a pair (i, j) is compulsorily followed by one of these pairs $(i + 1, j + 1)$, $(i + 1, j)$, $(i, j + 1)$. Dynamic programming takes advantage of these constraints to fill efficiently a matrix D of size $n \times m$ which stores in each cell (i, j) the total cost of the most efficient path to reach this pair. Hence, for any cell $D_{i,j}$ the problem has already been partially solved for the cell $D_{i-1,j-1}$, $D_{i-1,j}$ and $D_{i,j-1}$ and the lowest cost to reach (i, j) is:

$$D_{i,j} = \min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1}) + c(x_i^Q, x_j^R) \quad (5.1)$$

Once the whole matrix filled, the shortest path can be reconstructed by going from $D_{0,0}$ to $D_{n,m}$ and following the smaller costs. The sequences and the matrix D are illustrated in Figure 5.5 (a). Finally, the cost associated with a pair (i, j) has to be defined. For the matching of time sequences, a natural cost is the difference of magnitude (height) of the signal: $c(x_i^Q, x_j^R) = |x_i^Q - x_j^R|$. For other applications, the cost can be adapted.

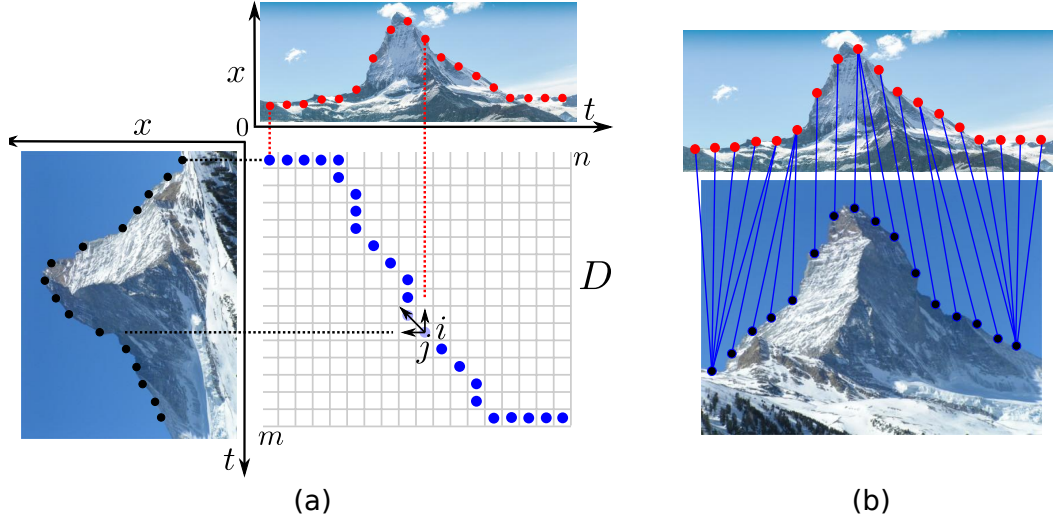


Figure 5.5 – (a) Illustration of DTW: The sequences are ordered. The shortest path ensures that the time constraint is not violated and that each feature has a correspondence. The shortest path to the cell (i,j) has already been partially solved for the previous pairs (note that the image on the left is mirrored for the purpose of the example). (b) Illustration of the set of correspondences which minimizes the sum of vertical distances between the features: each feature is matched, the ordering constraint is respected.

5.1.3.3 Dynamic Time warping for horizon-matching

In our problem, we want to adapt DTW for skyline matching and pose estimation, as illustrated in Figure 5.5. Typically, the horizon silhouette is extracted from a rendered view or a synthetic panorama and assuming the camera roll is equal to zero. In this specific setup, DTW shows some desirable properties:

- First, the warping is used to be robust to skyline distortions induced by image or panorama distortions, inaccurate focal estimation, inaccurate location and rendering artefacts.
- Moreover, in contrast to other matching strategies, it is able to take into account the order of the sequence, which for the horizon corresponds to an ordering according to the horizontal coordinates. Indeed, it is reasonable to formulate the hypothesis that there are no overhangs in the skylines.

In our problem of horizon matching, we have some skyline features located in $\mathbf{x} = [x, y]$. The time is replaced by the image coordinates on the horizontal axis x . The first series is the set of n query horizon features $h^Q = \{\mathbf{x}_0; \dots; \mathbf{x}_n\}$ and the second is composed of m reference horizon features $h^R = \{\mathbf{x}_0; \dots; \mathbf{x}_m\}$, both ordered by increasing horizontal coordinate: $x_0 < x_1 < \dots < x_n$. DTW computes a set of pairs $P\{(i \in [0; n]; j \in [0; m])\} = \{\dots; (i, j); \dots\}$, which minimizes the global performance measure between the matched h^Q and h^R . The global performance is the sum of the costs measured between corresponding features in P :

$$C_{tot} = \sum c(\mathbf{x}_i^Q, \mathbf{x}_j^R) \text{ with } (i, j) \in P. \quad (5.2)$$

and correspond to the value of the cell $D_{n,m}$. This value is a measure of the similarity of the skylines. Moreover, each horizon feature in the reference image is linked to a 3D coordinate $H^R = \{\mathbf{X}_0; \dots; \mathbf{X}_M\}$.

5.1. Camera orientation with DTW-based Skyline alignment

Correspondences computed with DTW are iteratively inserted in the least squares problem of Equation 3.12 (p. 28) where the translation is fixed to the GPS location and only the camera angles are optimized. Hence, a more accurate reference horizon is computed. Iterations are stopped once the orientation is stable or after a predefined number of iterations. In this way, the global performance measures the similarity after the alignment and the 2D-3D correspondences are used iteratively to retrieve the azimuth, roll and tilt.

We found that for horizon matching two cost functions can be considered. During the iterations of the DTW, we use Euclidean distances:

$$c_E(\mathbf{x}_i^Q, \mathbf{x}_j^R) = \sqrt{(x_i^Q - x_j^R)^2 + (y_i^Q - y_j^R)^2} \quad (5.3)$$

and vertical distances:

$$c_V(\mathbf{x}_i^Q, \mathbf{x}_j^R) = |y_i^Q - y_j^R| \quad (5.4)$$

as costs.

5.1.3.4 Skyline alignment with DTW

The general workflow for the skyline alignment is presented in Figure 5.6. The horizon sequences are firstly centered on their gravity center (b) and then rotated to make their principal orientation pointing in the same direction (c). Secondly, correspondences are extracted with DTW and used to estimate the camera rotation in several iterations (five in our experiment). The first and last iterations use the Euclidean distance as the cost function (d). Respectively, the first is used to align the sequence vertically and the last for fine pose estimation. During the other iterations, the vertical distance is applied to privilege lateral moves and thus improve the convergence rate (e). At each iteration, the camera orientation computed with the 2D-3D correspondences is used to project the 3D reference horizon features into the query pictures. Finally, the quality of the matching between the synthetic horizon and the query horizon (f) is simply measured by the sum of the squared distances between the query horizon features and their nearest neighbor in the reference feature set.

The different behaviours induced by the vertical and Euclidean costs are compared with ICP and presented in Figure 5.7. With DTW matching, the skylines are aligned in few iterations even if the initial horizontal alignment is bad. In comparison and such as illustrated in the second column of Figure 5.7, ICP is not able to converge if the initial alignment is too rough. It neither ensures that the features of each sequence are matched, nor that the horizontal order is respected. Finally, as the sequence order is not part of the algorithm, some matches can pull in one direction and some other in the opposite way. Hence, in the worst case, ICP aligns vertically the sequences but does not allow lateral moves.

5.1.3.5 Azimuth retrieval

If the two sequences are not aligned at the initial state (i.e. there is no prior about the camera azimuth but the location is known), a potential orientation has to be found. In this scenario, a 360° reference horizon (such as illustrated on top of the Figure 5.7 and in the schematic Figure 5.8(a)) is scanned with a sliding windows which is slightly ($\times 1.3$ in our experiments) larger than the query image horizon. Each sample of the reference horizon is then matched with the query horizon.

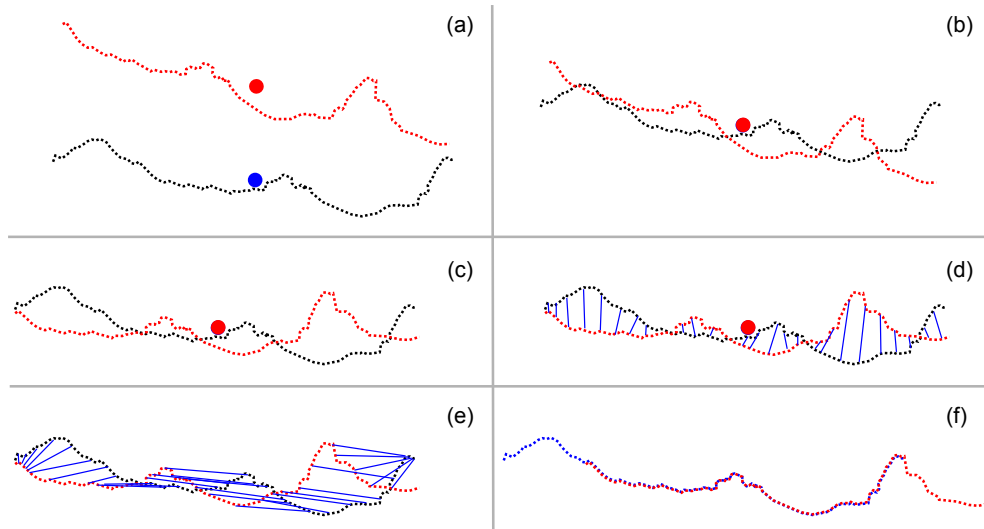


Figure 5.6 – Alignment of the skylines. (a) Skylines extracted from the picture (red) and from the DEM (black). (b) The skylines are centered on their gravity center. (c) Their principal orientations are aligned. Correspondences (blue lines) are extracted with DTW and Euclidean cost (d), with DTW and a vertical cost (e). The parameters of the alignment (e) are computed from the correspondences.

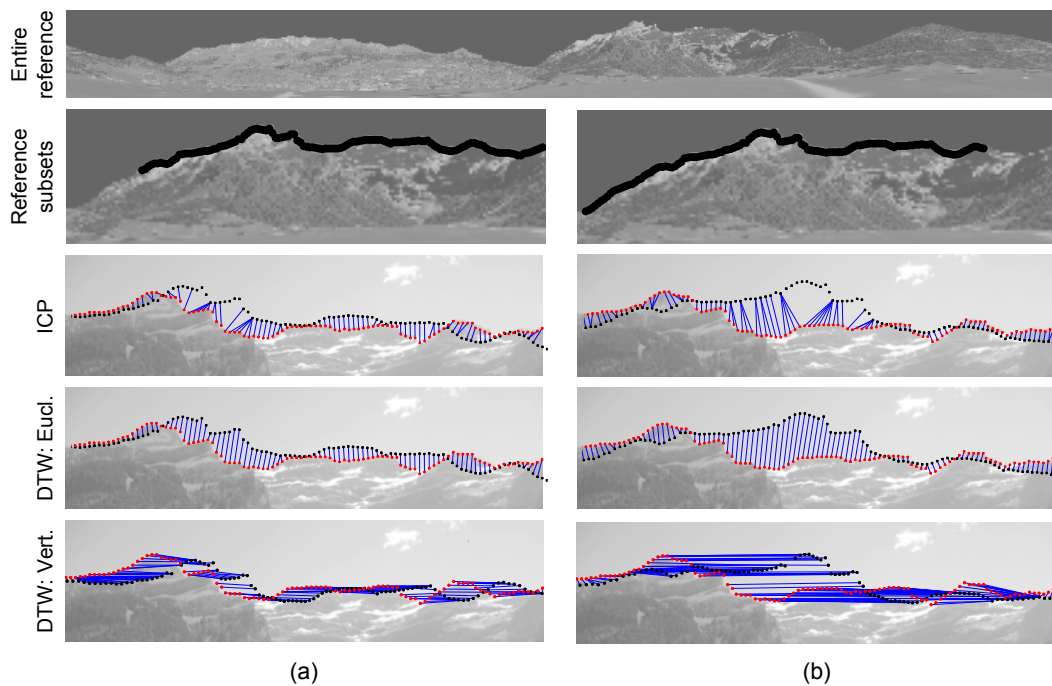


Figure 5.7 – Comparison of ICP and DTW (Euclidean and Vertical costs). The first line shows the entire reference panorama, the second two successive subset of the reference. The first column illustrates a small horizontal shift between the sequences, the second a large shift. The three last lines show the detected correspondences with the three different approaches.

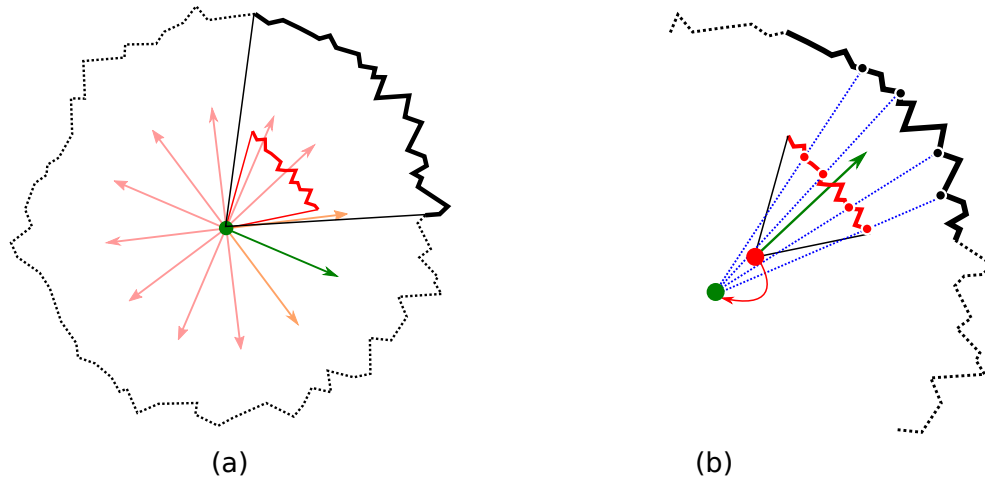


Figure 5.8 – Pose estimation with skyline matching. The image skyline is represented with the red line, the synthetic skyline is black. (a) A 360° reference skyline is computed. Subsets of the 360° skylines are matched and aligned with the image skyline. The direction corresponding to the best alignment is used to compute the image orientation. (b) The exact location of the camera is unknown. The same process as in (a) is applied to detect the best azimuth. Next, the detected 2D-3D correspondences are inserted in a pose estimation algorithm to compute an improved location.

Our algorithm has the advantage to compute not only the azimuth but also the tilt and roll. However, the drawback is that by allowing the camera to roll, the horizon becomes less discriminative, and multiple minima are found and sometimes the lowest cost does not indicate the exact orientation. This can be corrected by simply applying a threshold on the roll values computed and thus ensure that they remain in an acceptable range ($\pm 10^\circ$). A pseudo-algorithm is provided in Annex C.1.

5.1.3.6 Location retrieval

Finally, this method can be used to retrieve the 6DOF orientation of the camera. Indeed, a shift of the camera location induces a deformation and scaling of the skyline. The warping behavior of DTW can recover these deformations. Thus, the 2D-3D correspondences extracted can be inserted in pose estimation.

The algorithm proposed for this task is sensibly similar to the one presented to retrieve the camera orientation. It uses the same workflow to retrieve the best azimuth. However, in this case-study, the center of the panorama is not the GPS location (unknown, green location in Figure 5.8(b)) but a shifted location (the red dot). The 2D-3D correspondences extracted with the previous method constitute a block of observation for a Kalman filter (detailed in Section 3.5.3, p. 37). In the Kalman filter, the location is relaxed and a refined location is computed. Typically, this location is slightly above or under the ground level, the camera height is then shifted two meters above the ground. A new block of correspondences is extracted with the DTW matching. Finally, after a specified amount of iterations of the Kalman filter a new reference 3D image is generated in order to generate a new skyline sequence corresponding to the new location. The pseudo-algorithm is provided in Appendix C.2.

5.1.4 Experiments

5.1.4.1 Proposed scenarios

To validate these methods (see the objectives in Section 1.4, p. 7). We will detect correspondences between the query and reference skylines and use these 2D-3D correspondences to align the skylines and thus estimate the 3DOF and 6DOF camera orientation. We will use our own dataset which is presented in the next section.

In the first experiment, we apply the horizon matching to retrieve the camera orientation if its location is known and there is an azimuth direction prior. It corresponds to several useful scenarios:

- Annotation of pictures shot with a GPS-INS-compass enabled camera. Indeed, sensors mounted in our cameras, are currently not precise enough to provide a good direct georeferencing, that is an accurate overlay of geographic data with the picture (see Figure 5.9);
- Get (semi-)automatic additional GCP for the monoplottor if the user provides an initial orientation.

In the second experiment, we apply skyline matching more like Behringer (1999); Baboud et al. (2011), that is, to estimate the camera orientation if the camera location is known. This method can be applied to pictures shot with GPS camera but no compass or manually geotagged pictures.

Finally, in the last experiment, we would like our method to converge under imperfect conditions. Example of these conditions includes an approximation of the focal or an approximative location, we will then shift the initial location. Hence, the skyline matching has to be slightly elastic to take into account these deformations. Specifically, after the matching, we will extract 2D-3D correspondences and see if our method can retrieve the exact location.

5.1.4.2 Description of the dataset

The set of images was acquired in summer 2013, around "Les Diablerets", a Swiss resort close to Lausanne. This region was chosen for its proximity but also for its shape: a mountainous cirque. This morphology is ideal to test the skyline matching but also for the detection of visual correspondences between the picture and a synthetic model (see Section 5.2) which require hilly relief to detect matches in the background. Finally, it is also convenient for the 3D rendering because the cirque delineates the potentially visible regions and thus limits the size of the DEM and orthoimage to be processed.

The camera was attached to GPS/INS/compass sensors, in order to measure the camera orientation. However, it appeared that the values measured by the INS and the compass cannot be assumed to be the ground truth. For example, in Figure 5.9, the skyline measured with these values is overlaid on the pictures. One can easily notice that the alignment is bad. The orientation parameters were measured with sensors which are at least as good as the ones inserted in cameras, smartphones and tablets. The angular values measured are neither accurate enough for augmented reality nor to consider them as the ground truth for our experiments. Consequently, the ground truth azimuth was provided by the author by aligning visually the picture with the 360° panorama.

The images, detailed in Appendix B, are generally shot to avoid too much occlusion engendered by buildings or vegetation. In the database of 105 images, 73 images do not have foreground objects

5.1. Camera orientation with DTW-based Skyline alignment



Figure 5.9 – The reference skyline (red line) is projected in the picture according to the orientation parameters measured by GPS, compass and IMU. The alignment is bad.

(trees, buildings) that perturb the horizon line. Generally, a large part of the skyline is constituted from relief located far from the camera. These silhouettes are more stable against the variation of the camera location than closer skylines.

The generation of the synthetic images and panorama is made on the GPU with OpenGL. In our implementation, we take care to store not only the synthetic image but also the 3D world coordinates of each pixel. The DEM that we used has a 25m resolution. We choose to not involve a DSM for several reasons. First, a DSM has to be used with a finer resolution (to ensure that small objects appear). This may slow the rendering and induce memory errors. Second, since the skyline is far from the camera and the skyline usually not planted with trees, the distortions remain low and do not change drastically the skyline shape.

5.1.4.3 3DOF orientation with orientation prior

For the first experiment, we reproduced the case study of an image localized with a GPS and oriented with an IMU and a compass. Hence, we took the ground truth azimuth and generated random azimuths with a normal distribution and respectively a standard deviation equal to 2, 5, 10 and 15 degrees. The tilt and roll are also randomly fixed, with a standard deviation equal to 10 degrees (tilt) and 5 degrees (roll). These parameters are used to generate reference images 1.5 time larger than the query image. For each query image which does not have an occluded skyline, we produced 10 images for each azimuth distribution. In the worst cases, the initialization of the roll and the azimuth of the reference images with a normal distribution, some reference images have unnatural rolls and limited overlap with the query image and the alignment is extremely challenging. Then, the algorithm presented previously in Section 5.1.3.4 is applied to align the skylines and estimate the 3DOF camera orientation.

In the Figure 5.10, we show the boxplot of the computed angles, centred on the median of the poses computed with an azimuth distribution having a two degrees standard deviation and which can be considered as the ground truth. The boxes surround the data within the second and third quartile and the red line is the median. The whiskers are set to the 5 and 95 percentiles, and the outliers are illustrated with red dots.

The easiest parameter to retrieve is the tilt. Indeed, it is set with a vertical alignment of the skylines which is trivial since, in general, the skylines are horizontal. The heading and roll are more correlated. If a wrong heading is computed, then the roll compensates for it. It appears that the outliers respect the initial normal distribution used to generate them. Hence, for the outliers, the DTW matching does not change the initial state. However, for most reference images, the pose estimation improves the initial

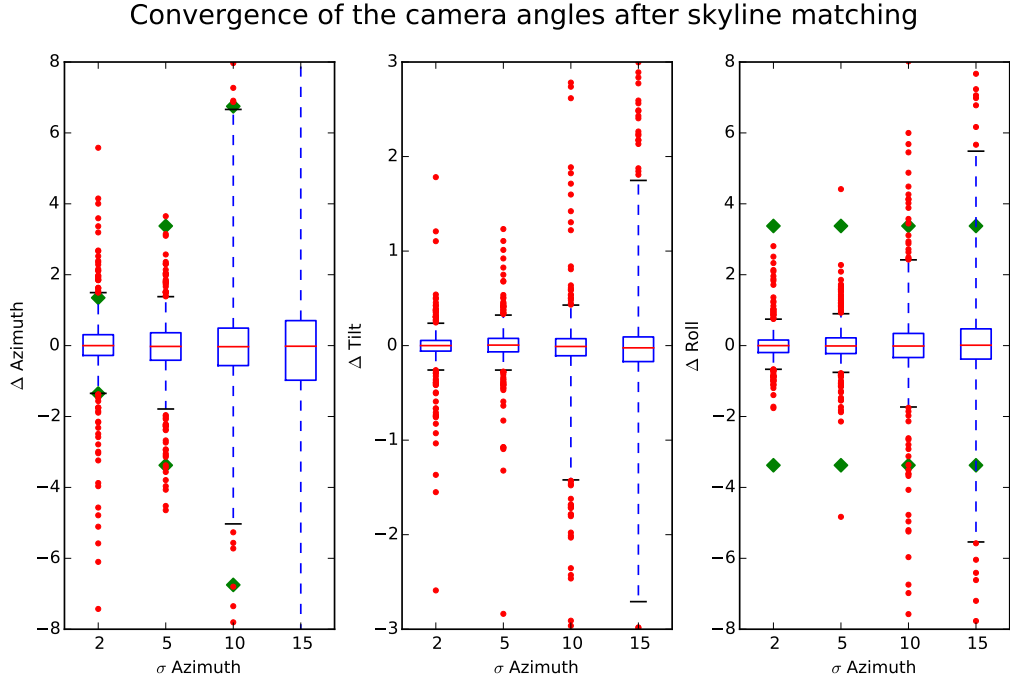


Figure 5.10 – Analysis of the convergence of the pose. Initial reference skylines are generated from a normal distribution of the azimuth with sigma equal to 2, 5, 10 and 15° (horizontal axis) and varying tilt and roll. The boxplots summarize the distribution of the computed camera angles centred on the median (red line). The green diamonds show the 5 and 95 percentiles before the alignment (do not appear for the tilt because they are outside the limits of the plot).

state which can be seen with the size and extent of the boxplot.

This experiment is proposed mainly to analyze the performances of our matching strategy. In real conditions, we would rather use the settings of the next experiment. That is, we would generate a horizontal panorama cropped according to the azimuth prior. However, the current experiment tells us about the number of steps to be used to initiate the alignment (the number of arrows in Figure 5.8(a)). Hence, with an azimuth step of 5 degrees we can expect an azimuth error below two degrees, a tilt below 0.5 degree and a roll below one degree in 90% of the cases.

With this information about the convergence, we can now lead the second experiment, which is the computation of the 3DOF orientation without azimuth prior.

5.1.4.4 3DOF orientation

In this experiment, we will use DTW to recover the camera orientation if its location is fixed and no prior on the azimuth is provided. To assess if the orientation converges to the proper orientation, we compared it with the ground-truth azimuth generated by the author. The reference is a synthetic 360° panorama generated by virtually swiveling the camera around the GPS position. The panorama dimensions are computed according to the focal and dimensions of the query image. Hence, our query images have a dimension (380 × 500) while the panoramas have a dimension (650 × 3000)

Using the algorithm presented in Section 5.1.3.5 and 50 subsets of the panorama (each 7.2°), 81

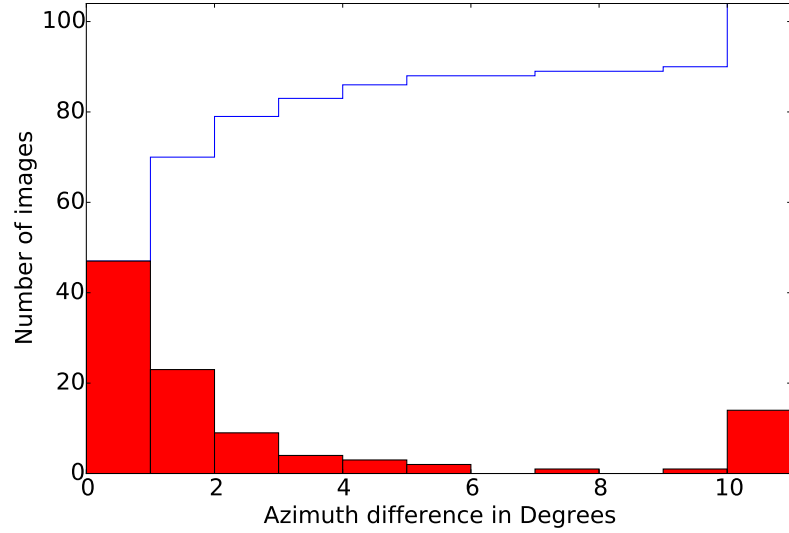


Figure 5.11 – This histogram show the number of images which are in a given angular distance of the Ground Truth. The last bin contains every difference above 10° .

images (78%) converge to an azimuth which is within 5° of the Ground Truth (see Figure 5.11). We use this limit of 5° to evaluate this result since the ground truth is detected visually and cannot be assumed to be perfect. Among the failure cases, we can find horizon lines perturbed by foreground objects, images representing mountains not contained in the reference panorama (because situated outside of the selected DEM) and images which have a poorly discriminative horizon.

To improve slightly this result, we fixed a threshold (10 degrees) on the roll angles and discard poses, which are associated with these unlikely values. By adding this prior, five more images are well oriented (86 images: 82%).

Finally, can also compare ICP with DTW. We kept the same settings but replaced the correspondence detection with DTW by the nearest neighbor. Only 15 images (14%) converge to the proper azimuth. As expected, ICP is not able to converge if the initial alignment is too rough. Hence, to improve the ICP-based matching and try to reach the performance of the DTW, we should increase both the azimuth discretization and the number of iterations of the correspondences detection.

With this experiment, we proved that we can successfully retrieve the camera orientation of a camera which has a known location (measured with GPS). We can indeed use the same scheme using an azimuth prior. This prior will decrease the search range and make the orientation faster and more robust. Moreover, the generation of the 360° panorama induces a deformation of the reference skyline (mainly in the horizontal direction) which is recovered by DTW. The main source of failure is the presence of foreground objects, which occlude the skyline.

5.1.4.5 6DOF orientation

In this experiment, we use the 2D-3D correspondences detected by DTW to recover the full pose (6DOF). For this case-study, the initial pose is generated by shifting the ground truth location and providing an azimuth prior. With this experiment, we want to assess i) whether DTW detects correspondences if the shape of the skyline is deformed and ii) whether these correspondences can improve the estimate of

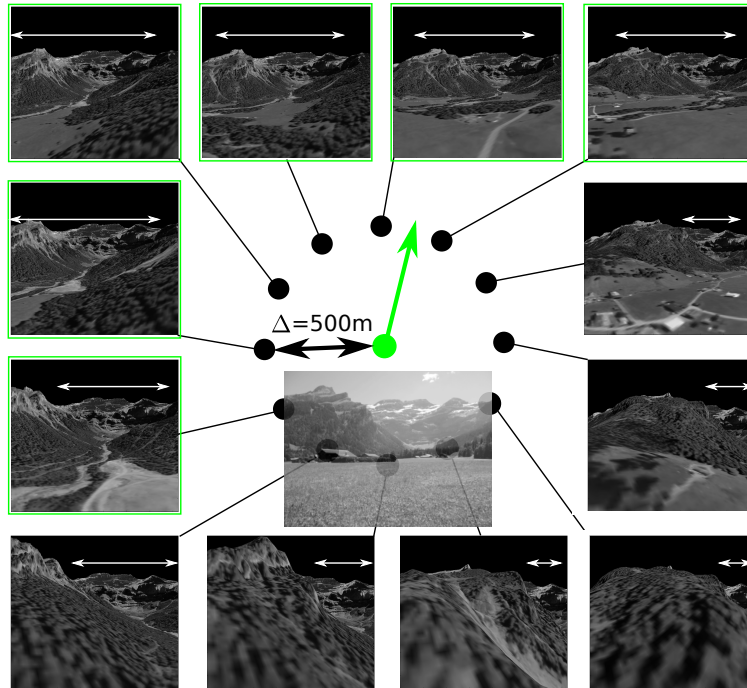


Figure 5.12 – 12 reference images (black dots) are generated at locations around the ground truth location (green dot) according to a fixed shift Δ . The azimuth is illustrated with a green arrow. Matchable skylines are framed in green. The with arrows show the part of the reference skyline which is similar to the query skyline.

the camera location. For this case study, we keep only images without occluded skyline and without DEM mismatches.

Figure 5.12 illustrates the experiment. Twelve reference images are generated in every direction around a query image location (black dots). In this example, the locations are shifted 500m away from the real location (green dot). Only the skylines of the images framed with a green box can be considered as a deformation of the observed skyline and thus possibly aligned with DTW. The other skylines have new elements appearing and causing occlusion which make them challenging (or even impossible) to match. The white arrows delineate the part of the reference skylines which overlap the query skyline.

Figure 5.13 summarizes the results for four images. The green dot represents the ground truth location and the green line the viewing direction. The black dots are the starting locations shifted by 100m, 250m, 500m and 1000m. They are linked to red dots which are three successive locations computed during the pose estimation iterations. Each red dot represents also a new reference image computed.

These images represent ideal skylines. In these ideal conditions, our algorithm is able to improve the initial pose. For a shift of 100m, the matching improve the location of the image A and C. For the two other images, the matching can even worsen the initial location. Thus, if the location prior is expected to be within a distance of 100m of the real location, it would be more appropriate to fix this location. This result suggests also that skyline matching (at least in the conditions of our experiment) can not be expected to estimate a location within a 100m radius of the real location. With a shift of 250m, our

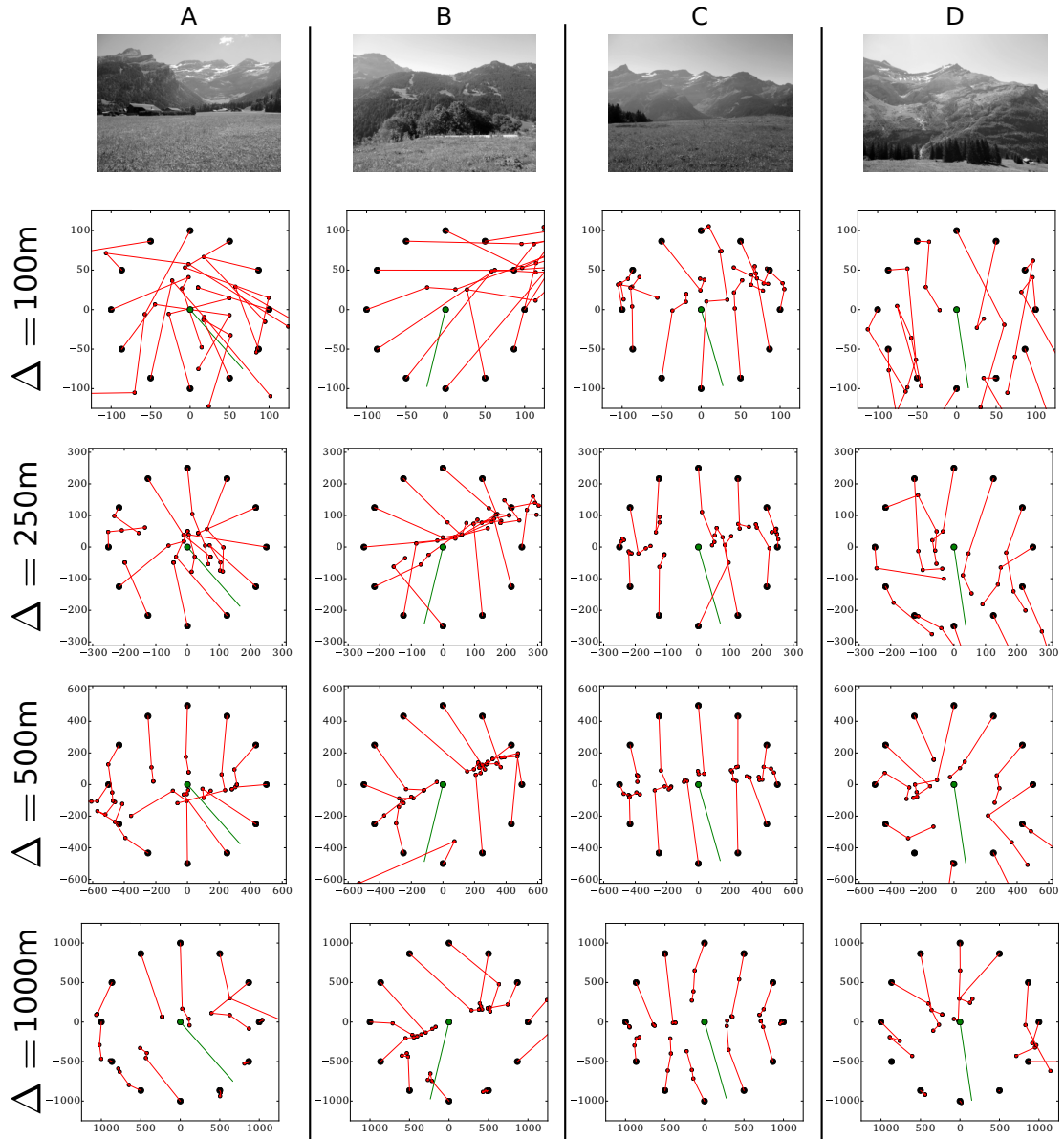


Figure 5.13 – For ideal skyline, our algorithm can improve an a priori pose. Black points are the initial locations shifted by 100, 250, 500 and 1000m. Red points and lines represent three successive iterations of the pose estimation.

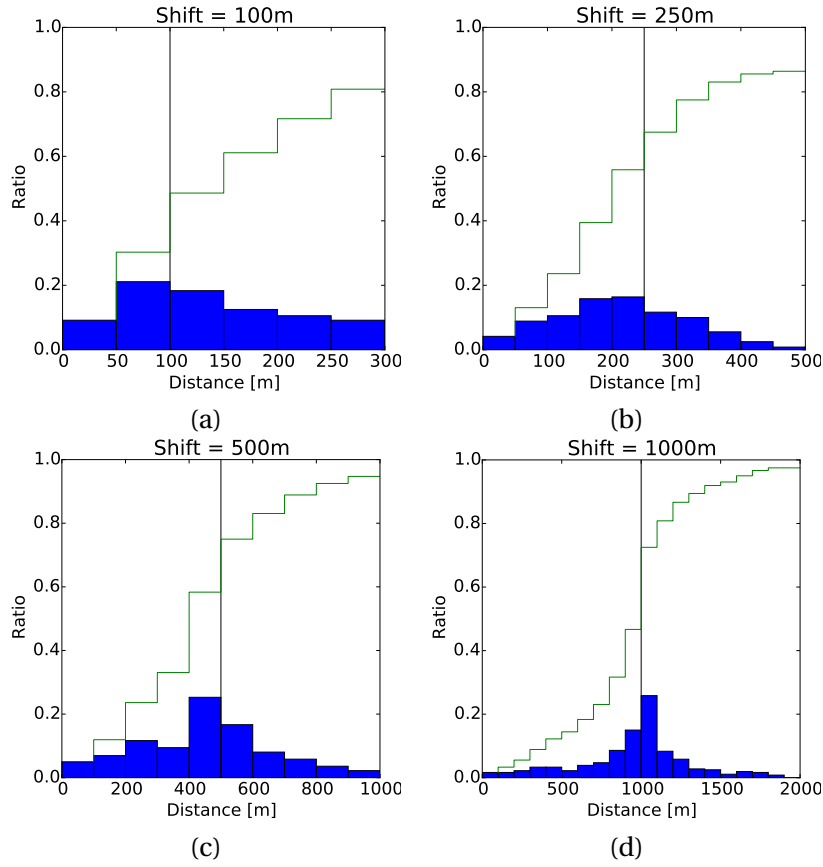


Figure 5.14 – Summary of the distances measured after three iterations for 30 images for which the skyline is not occluded. For each image, 12 shifted locations in every direction are generated to initialize the skyline matching. The black line represents the shift, on its left the initial states are improved.

algorithm generally improves the initial location, some exceptions can be found for the images B and D. A shift of 500m does not change drastically the result of the 250m shift. However, more computations diverge for the two first images (A, B). Finally, even with a 1000m shift, under ideal conditions, some pose estimation iteration converge to the real location. Often, the computed locations are aligned along a line perpendicular to the viewing direction. We can then deduce that the skyline is good at detecting the scale (depth) but has more difficulty to retrieve the proper unique combination of azimuth and location and allows lateral moves of the camera. However, since our algorithm can still improve an initial shifted location even if this location is 1000m away, we can say that the characteristics of DTW are appropriate to match skylines with distortions.

Figure 5.14 synthesizes the results for the 30 images which do not have an occluded skyline. For each image, twelve shifted reference were generated and used to start the georeferencing. After three iterations, the distance between the ground truth location and the computed location is measured (the 30 images and the pose iterations started at 500m are presented in Appendix D). Figure 5.14 shows the resulting histograms. The vertical line is centered on the initial shift. Thus, on its left, we can find the estimated locations which improve the initial location. Generally, the vertical line is very close to the median and we can thus not significantly argue that our method improve the initial state. The reasons for this mitigated result is evident in Figure 5.12 (corresponding to the third line of the first

column in Figure 5.13): the skylines can be drastically modified rather than simply distorted within a short distance. These modifications are beyond the warping capacity of DTW. A second reason is the presence of skylines not marked enough which are hard to align (and subsequently the 6DOF orientation fails).

5.1.5 Discussion

We can draw three conclusions from these experiments.

- First, for the 3DOF orientation, the main sources of failure are an occluded skyline and unmarked skylines.
- Next, for the 6DOF orientation, as expected, DTW can recover skyline distortions engendered by a shift of the location.
- However, in real conditions and hilly relief, it appears that the skyline is drastically modified with small shifts of the camera location. Only a small part of the reference and query skyline are similar if at all. The matching of part of sequences is beyond the capacity of DTW which supposes that the entire sequence is similar, but distorted.

Hence, for future works, we could first consider higher resolution images. In this way, we can get more marked skylines (in which the peaks are real variations of the relief and not due to noise of the watershed segmentation or noise of the 3D rendering). Higher resolution would certainly improve the skyline alignment with DTW and thus the accuracy of 6DOF orientation. Next, in this work, we consider the entire skyline, but we could probably reduce the processing time and improve the accuracy by considering specific features such as local maxima and minima of the skyline sequence. Finally, the elasticity of DTW is promising to recover also the focal of the camera if its location is known. It is especially desirable for collections of images recorded with various cameras such as considered in this work. Regarding our results, we can also say that it is not DTW which is inappropriate for 6DOF orientation (since it works in ideal conditions), rather the proposed workflow. Since the skylines vary drastically with short-range displacement, one should consider another method to determine the camera location. For instance, we could measure the similarity of the observed skyline with skylines generated for a grid of location (e.g each 200m) and deduce the location having the skyline the most similar to the image. In this scenario, DTW keeps its validity to align the skyline and evaluate the similarity.

5.1.6 Conclusion

In this Section, we illustrated the use of skyline matching for fine pose estimation of a camera. In particular, we exploited the DTW algorithm which was already successfully applied by several authors for alignment of speech sequences or walking tracks. Compared to ICP, which is often applied in similar conditions, DTW considers the horizon as an ordered sequence. Thus, the detected correspondences respect the natural horizontal ordering of a skyline. Moreover, it can be used jointly to a distance measured only in the vertical direction, (the horizontal direction being the "time" of the sequence). These two characteristics make DTW-based skyline matching converging even if the initial alignment is bad and also capable to recover skyline distortions due to scaling or location shift. DTW is computationally more intensive than ICP, but its dynamic implementation associated with a sampling of the horizon sequences and a better convergence keeps it fast and efficient.

First, we tested it to recover the orientation of a camera associated with a fixed location. In these conditions, it converges for most of the images in our database. A typical use of this method is the 3DOF orientation of a picture shot with a GPS camera for instance for augmented reality. With ideal skyline shapes, DTW matching can be used to retrieve not only the camera orientation but also its location. These ideal conditions are marked skylines not perturbed by occlusion or relief close to the camera. It appears that, in practice, these conditions are rarely met. However, in supervised georeferencing, the 6DOF orientation method could be easily and beneficially inserted in our monoplotters. In particular, after the detection of an initial orientation by the operator, skyline matching with DTW would be appropriate to provide GCP. Moreover, it can be applied not only to skylines but also to any linear features (rivers, roads etc.) digitized by the operator (Ground Control Lines rather than GCP) and thus facilitate considerably the georeferencing. In unsupervised conditions and at a local scale, we could consider the comparison and matching of the query horizon with skylines generated for a regular grid of potential locations. At a larger scale, Baatz et al. (2012a) already proposed an efficient solution. The method that we propose could be complementary to their results to improve the final alignment.

A clear limit of skyline matching is the provision of a solution only in mountainous areas where the shape of the skyline is unique. Hilly regions can already be beyond these limits. Skylines provide also correspondences only in the top part of the images, a configuration which is not optimal for pose estimation. Another drawback of techniques based on the skyline or morphological break-lines is that they require an initial processing of the query image to detect the sky or edges. This process is usually not entirely mastered for challenging images (historical, black and white, having specific weather conditions).

5.2 Detection of correspondences between real and synthetic images with HOG

In the previous section, we showed how a DEM, and especially the skylines, can be used for the fine pose estimation of a landscape image. To complete these methods, in particular to detect correspondences not only in the skyline but in the entire image, we want to mimic the task of an operator providing GCP in our monoplottter. Thus, we propose to generate realistic 3D synthetic images and detect automatically corresponding regions of the query image. In a preliminary study (Produit et al., 2012), we showed that we can match a synthetic image with a real picture with Normalized Cross Correlation and a geometric constraint. This technique was applied to the alignment of web-shared images in a 3D model. In this chapter, we propose a more compact descriptor which was successfully applied for cross-domain matching (different type of images), namely the Histogram of Oriented Gradients (HOG). Moreover, we will assess the proposed method on a dataset with ground truth orientation.

5.2.1 Introduction

Real and synthetic images, similarly to those presented in Figure 5.15, show similarities. Nevertheless, when subjected to closer scrutiny, they show significant differences. First, orthoimages do not have high temporal resolution: for instance, in Switzerland, they are updated every six years and campaigns are done during the summer only. Hence, there is very little chance that an orthoimage captures the same illumination and land cover state than a query image. Note that with new satellites, dedicated to the acquisition of aerial views at higher spatial and temporal resolutions, the renewal time should decrease and seasonal views will become available. Nevertheless, as long as aerial views are not shot exactly at the same time as the query image, illumination, shadows and land cover differences will occur. Second, the differences between the viewpoint (top and orthogonal for the orthoimages, high-oblique perspective for the query images) generate distortions visible mainly in the foreground and on steep slopes. Once again, new sensors record not only nadir views, but also oblique images. However, in most landscape areas, only top views are available as references. Finally, the DEM also has an impact on the rendered synthetic views. Digital Surface Models (DSM) created from stereophotography or LIDAR acquisitions can be useful to generate more realistic images. Their main drawback is technical: they have a higher resolution (i.e. $<2\text{m}$), which may generate some rendering problems due to memory. Moreover, for terrestrial pictures, the high vegetation and buildings in the foreground can also have undesired properties such as occlusions of the landscape (where the correspondences are detected). Furthermore, high-resolution DSM are not yet covering the entire earth.

In summary, real and synthetic images are not as similar as they seem at a first glance. As illustrated in Chapter 4, an operator can overlook these dissimilarities, whereas these differences limit the success of standard key-points detectors and descriptors. Indeed, the similarities are essentially visible at large scale, but they disappear by zooming in the images. Hence, in the proposed methods, we will match large patches of images rather than local features. In this chapter, as in the previous one, we will break the problem into two parts. Firstly, methods to retrieve the 3DOF camera orientation will be presented. In a second time, on the basis of our preliminary work (Produit et al., 2012), we will discuss how the relaxation of the location could be implemented. For the 3DOF camera orientation, we will compare two methods: firstly, the matching of the entire query image with a 360° panorama and secondly, the matching of local patches of the panorama with the query images. Smaller patches are expected to be more accurate, more capable to handle occlusion but also more impacted by the rendering distortions.

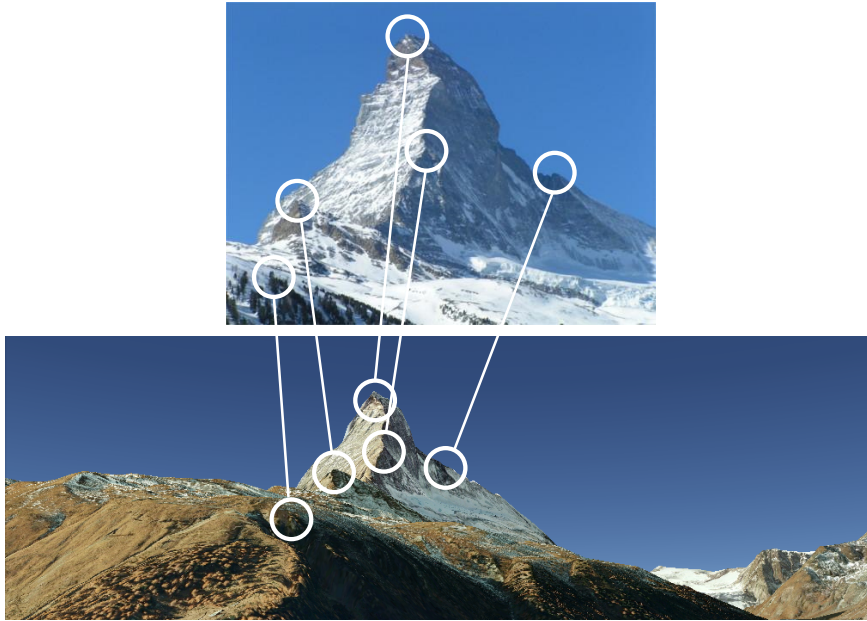


Figure 5.15 – Top: Example of a query image. Winter shot of the Matterhorn. Bottom: Example of a synthetic reference image (Google Earth snapshot). The aerial views used for the rendering have been shot during the autumn. Distortions of the trees caused by the DEM and the orthoimage are visible. The details of the face are smoothed by the steep slope. (Wikimedia, Google Earth)

5.2.2 Review

The existing methods to register and georeference landscape images were discussed in the previous chapters. To our best knowledge, none of them considers synthetic images as the reference. Hence, to tackle our problem, we decided to test approaches used for cross-domain matching, i.e. the matching of pictures from different seasons, lighting conditions or even paintings and sketches (Shrivastava et al., 2011). In cross-domain matching, *"pixel-wise matching fares quite poorly, because small perceptual differences can result in arbitrarily large pixel-wise differences. What is needed is a visual metric that can capture the important structures that make two images appear similar; yet show robustness to small, unimportant visual details"* (Shrivastava et al., 2011). Some successful approaches use Histogram of Oriented Gradient to describe and match images under these conditions. For instance, Aubry et al. (2014) detect, describe and match "discriminative visual elements" of a 3D model of an architectural site with HOG. In that work, these patches are then matched with paintings to retrieve their pose.

The Histogram of Oriented Gradient (HOG) was developed firstly for pedestrian detection (Dalal and Triggs, 2005). The descriptor is quite similar to other feature descriptors (such as SIFT): an image is first divided into cells, in which the histogram of orientation of the gradient is computed. Next, to make the descriptor more robust to illumination and shadow variations, the histogram is normalized locally. In the example of Figure 5.16, cells with a side of 40 pixels divide this 200×200 image. Hence, 25 histograms are computed, each of them has six orientation bins whose magnitude are presented in Figure 5.16 (b). Results of HOG matching are usually improved if a block normalization is applied, which means that the histograms values are normalized within a block of contiguous cells. Here, a 2×2 block normalization is illustrated with the red box.

What make HOG different to its predecessors is essentially the way it is matched and the size of the

5.2. Detection of correspondences between real and synthetic images with HOG

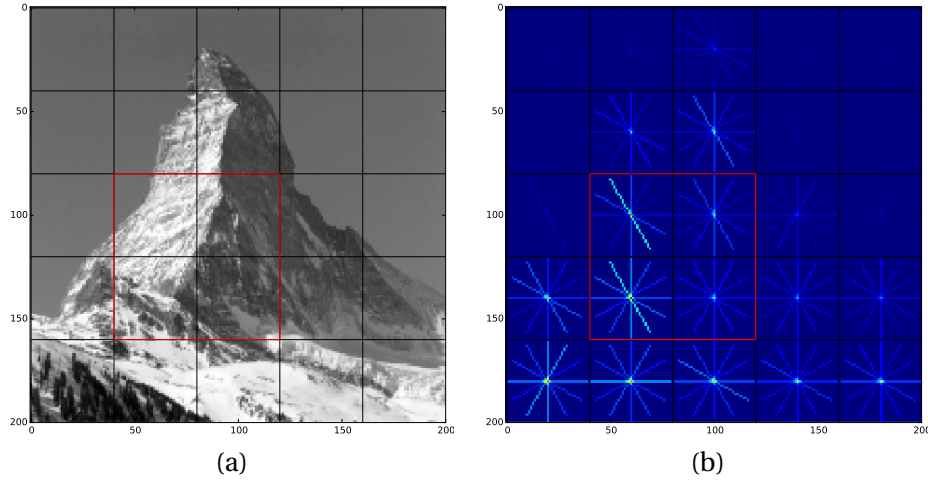


Figure 5.16 – Patch of a query image and visualisation of the HOG descriptor. (a) Cells of 40 pixels have been used in this example for a 200 image resulting in 25 histograms. Orientations are split into six slices resulting in a HOG feature of 150 elements. A 2 block applied for the normalization is illustrated in the red box. (b) The histogram of each cells are represented with the magnitude of the orientations.

patch being described. While SIFT is usually used in association with a key-point detector and matched with a Euclidean distance, HOG is regularly and densely computed on the image. In the original paper, HOG is matched with a SVM classifier trained to detect pedestrian versus non-pedestrian classes. For the matching of a 3D model with pictures, exemplar-SVM is implemented instead (Aubry et al., 2014). In that method, a classifier is trained to separate each patch of the model described with HOG (the positive example) from a set of negative example (patches that do not contain the positive example). The main limitation is then the SVM-matcher, which involves an intensive offline training with positive and negative data and intensive matching with the classifier at test time. This specific processing is then not well adapted to our problematic.

5.2.3 Method

This section presents the proposed approach. To give a general overview, on the one hand, we produce synthetic 3D images according to the prior of the image pose. On the other hand, we have the query image which has to be aligned with the synthetic image. To do so, a patch of the query image is described with HOG and slid on the panorama (or inversely). The HOG descriptor similarity is measured at each location. With this method, our intention is to avoid a key-point detector, which is expected to be hardly capable of extracting similar key locations under these cross-domain conditions (except perhaps on the skyline). Finally, for the matching we do not implement a SVM-classifier to keep our method easily applicable and processable in a limited time. Indeed, with a pose prior, the possible correspondences are limited and we expect that a simpler matching scheme can still detect correspondences. Especially, we can measure the similarity of the descriptors \mathbf{x} and \mathbf{y} with a correlation-based distance:

$$d_c = \frac{1}{n} \frac{\sum_{i=0}^n (x_i y_i - \mu_x \mu_y)}{\sigma_x \sigma_y} \quad (5.5)$$

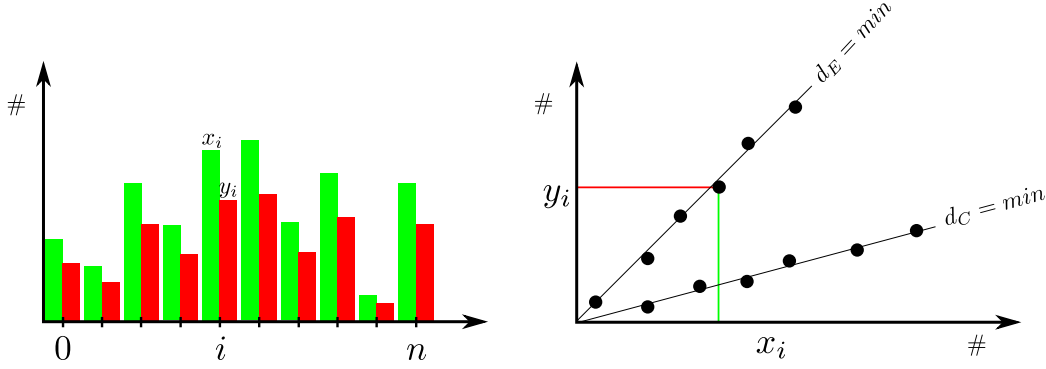


Figure 5.17 – x and y are two descriptors having a dimension n , they are a juxtaposition of histograms (one histogram for each cell) to be compared. The Euclidean distance is minimal if the values are aligned on the diagonal, whereas the correlation-based distance is minimum if the values are aligned on any line. For instance, for the histograms presented on the left, the Euclidean distance is non-zero, but the correlation-based distance is minimal.

where n is the number of dimensions of the descriptor, μ its mean and σ the standard deviation. The correlation-based distance proves to be more appropriate than an Euclidean distance:

$$d_e = \sqrt{\sum_{i=0}^n (x_i - y_i)^2} = \sqrt{\sum_{i=0}^n x_i^2 + \sum_{i=0}^n y_i^2 - 2 \sum_{i=0}^n x_i y_i} \quad (5.6)$$

Indeed, the correlation distance has the advantage to respond positively also under changes of scale (see Figure 5.17). Note that if x and y are reduces to standard scores, d_E and d_C are equivalent.

Next, we will compare two methods to retrieve the 3DOF orientation of a picture. A global matching considering the entire query image and the matching of local patches. They are illustrated in Figure 5.18.

5.2.3.1 Global matching

For the global matching (see Figure 5.18(a)), the entire image is described with a HOG descriptor. The similarity with the HOG descriptors of the patches of the panorama is measured densely. Local minima of the correlation distance highlight possible locations of the query image. This algorithm is presented in the Appendix E.1.

This approach is fast and easy to implement. However, it takes into account every region of the query image, including the sky perturb the descriptor, foreground where the most extensive distortions are expected and where occlusion of the landscape by vegetation and buildings appears. These regions of the query image are visually the most dissimilar and thus perturb the matching.

5.2.3.2 Local matching

Thus, in the second method, for the purpose of limiting the negative influence of these regions, we will consider local patches and add three constraints i) **a geographic constraint** to discard the potentially disturbing patches, ii) **a multiscale constraint** to promote inliers correspondences which are supposed to match for varying size of patches, iii) **a geometric constraint** to reject outliers which do not follow

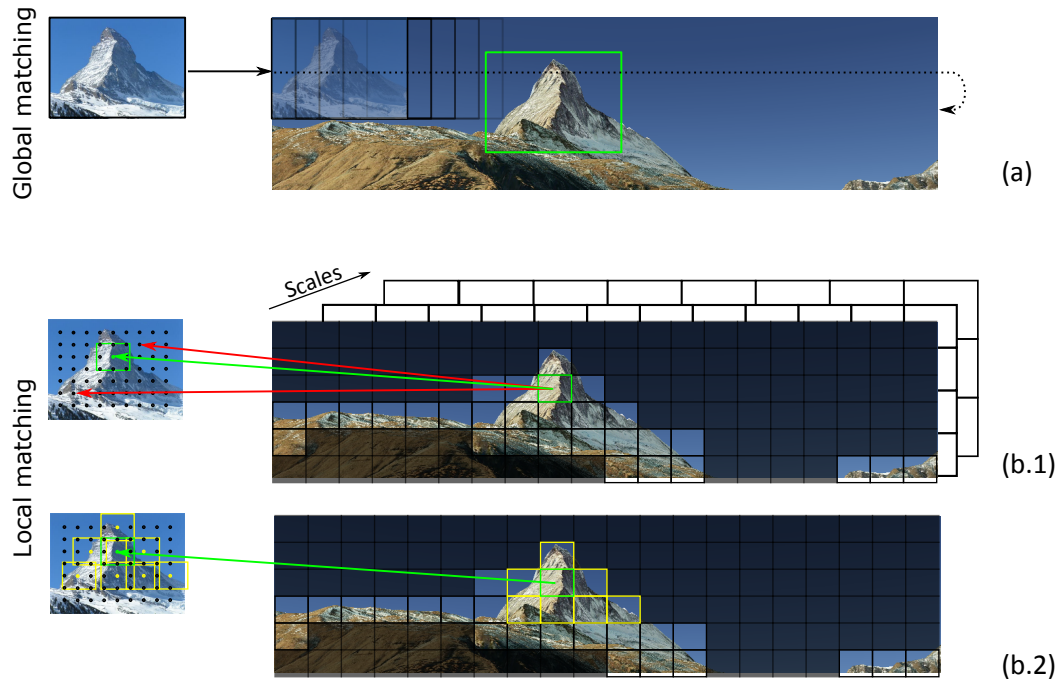


Figure 5.18 – (a) Global matching, the query image, described with HOG, is slid on the panorama. (b-c) Multi-scale matching, patches of panorama are matched with the query image. The patches are described at several scale and a geometric constraint is applied. Disturbing cells (black) are discarded: patches within the sky or close to the camera.

the alignment model.

Geographic constraint The panoramas, unlike query images, are associated with geographic information. Especially, for each panorama, we record not only the image but also the 3D world coordinate of each pixel. We create a regular grid of locations on the panorama defining the centers of a set of patches. Since we suspect the regions close to the camera to have a bad influence, we suppress local patches near to the camera. Similarly, we also discard patches dominated by sky.

Multiscale constraint Typically, many false correspondences are detected at this level because the patches are small and prone to mismatch. We will then add a multiscale constraint to improve the detection. Firstly, unlike false matches, corresponding patches are expected to match at several scales. Thus, for each center applied on the panorama, several descriptors computed for several size of patches are computed.

Geometric constraint and outlier rejection In parallel, the query image is processed in the same way: HOG descriptors are computed for a regular, but denser, grid of location. For each scale, a patch of the synthetic image is slid on the query image and the three first local minima are selected as potential correspondences.

The goal of the geometric constraint is to force patches which are close in the panorama to remain

close in the image. If we assume that the roll of the query images is small, each detected correspondence defines a translation from the image to the panorama (green arrow in the Figure 5.18(b.1)). We can then try each of these translations and find the one that matches the largest amount of correspondences. This step is very similar to RANSAC. However, since a simple translation is expected from the panorama to the image (but no scaling and no rotation), a single correspondence defines the translation and each correspondence can be tried rather than a random combination of correspondences. In this way, the method is not subjected to the random behavior of RANSAC.

3DOF orientation Finally, the correspondences detected with this method are used as 2D-3D correspondences to estimate the 3DOF orientation of the image. This time a real RANSAC is computed to eliminate remaining disturbing correspondences. The entire algorithm is presented in the Appendix E.2.

5.2.4 Experiments

To validate the objectives and hypotheses presented in Section 1.4 (p. 7). We will estimate the 3DOF orientation of the dataset of pictures presented in the previous section (see Section 5.1.4.2). We implemented both methods discussed earlier to retrieve an image location within a panorama generated from the picture location and hence retrieve the 3DOF orientation.

5.2.4.1 HOG to describe the entire image

In this experiment, the same synthetic panoramas as those presented for the skyline matching are the reference for the orientation. The goal is to retrieve the best location of the query image within the panorama. Ground truth locations were generated by the author with a click on the panorama. In this first experiment, we want to use the entire information contained in the query image. Hence, the image is down-scaled and described with a unique HOG description (Figure 5.19(a-b)). In parallel, the panorama is scaled accordingly. Finally, the query image is slid on the panorama and a distance is measured in every location of the panorama. Local minima are detected as potential locations (and thus orientation) of the query image (c). For the plots presented in Figure 5.20, we will measure for the entire dataset of images if the ground truth location is in the proximity of a local minima and the rank of this minima (sorted by the measure of similarity). The best configuration is that the ground truth location matches the first local minima (positive detection).

A HOG descriptor has many parameters: the number of cells within the patch, the number of pixels in one cell and the number of orientation bins. By varying these parameters, those which bring the best rate of good detections were chosen. Figure 5.20 shows the amount of positive detection and the cumulated amount of ground truth matches within the four first local minima (sorted). Hence, the crossing of a line with the vertical axis indicates the ratio of positive detection. We propose as baseline a HOG descriptor with 10 cells of 8 pixels (which means that the images are down-scaled to the size 80×80), 7 bins to quantize the orientations and a normalization with 3 blocks. The matching of the HOG description is done with a correlation-based distance.

- In Figure 5.20 (a) the baseline is compared with the result of a Euclidean distance. The correlation distance improves greatly the results: from 35% to 55% of positive detection in the dataset. Hence, apparently, HOG descriptions of synthetic images and real images are similar but have different

5.2. Detection of correspondences between real and synthetic images with HOG

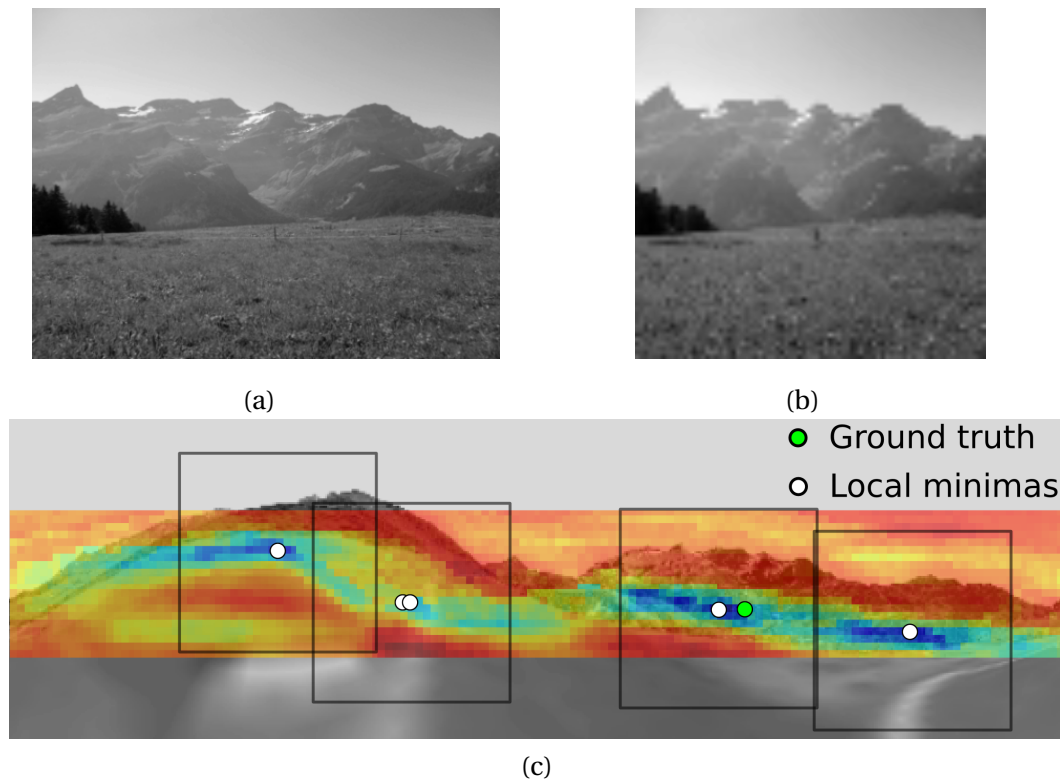


Figure 5.19 – Original (a) and scaled (b) query images. (c) Scaled panorama overlaid by the measured correlation distance. Five local minima are selected (white) and can be compared with the ground truth location (green)

scales.

- In the Figure 5.20 (b), we vary the number of pixels per cell, by increasing it from 8 to 14 pixels, the ratio of positive detection slightly improve to reach 60%. However, it also makes the computation heavier (with 14 pixels an image of size 140 have to be processed).
- We tried then to change the number of orientation bins, it is illustrated in Figure 5.20 (c). It is difficult to extract a general trend from this graph. The variation of the number of bins does not change drastically the accuracy within the first minima.
- Then, the influence of the number of cells is presented in Figure 5.20 (d). Apparently, 12 cells can improve the accuracy whether the other number of cell tested do not change the results.
- Finally, in the Figure 5.20 (e), we propose to change number of blocks for the normalization: a normalization with 2 and 3 blocks produces the same result which is improved compared to no normalization.

For the best combination of parameters, we obtain 62% of positive detection and around 80% of the ground truth location within the four first local minima. Regarding our dataset composed of images with occlusion, backlight, slight camera rotation and large ratio of foreground, it is a good result. Indeed, the failure cases can be explained by occlusion made by foreground objects (trees, buildings) or by large parts of the image composed of first plane objects which are too much deformed to be

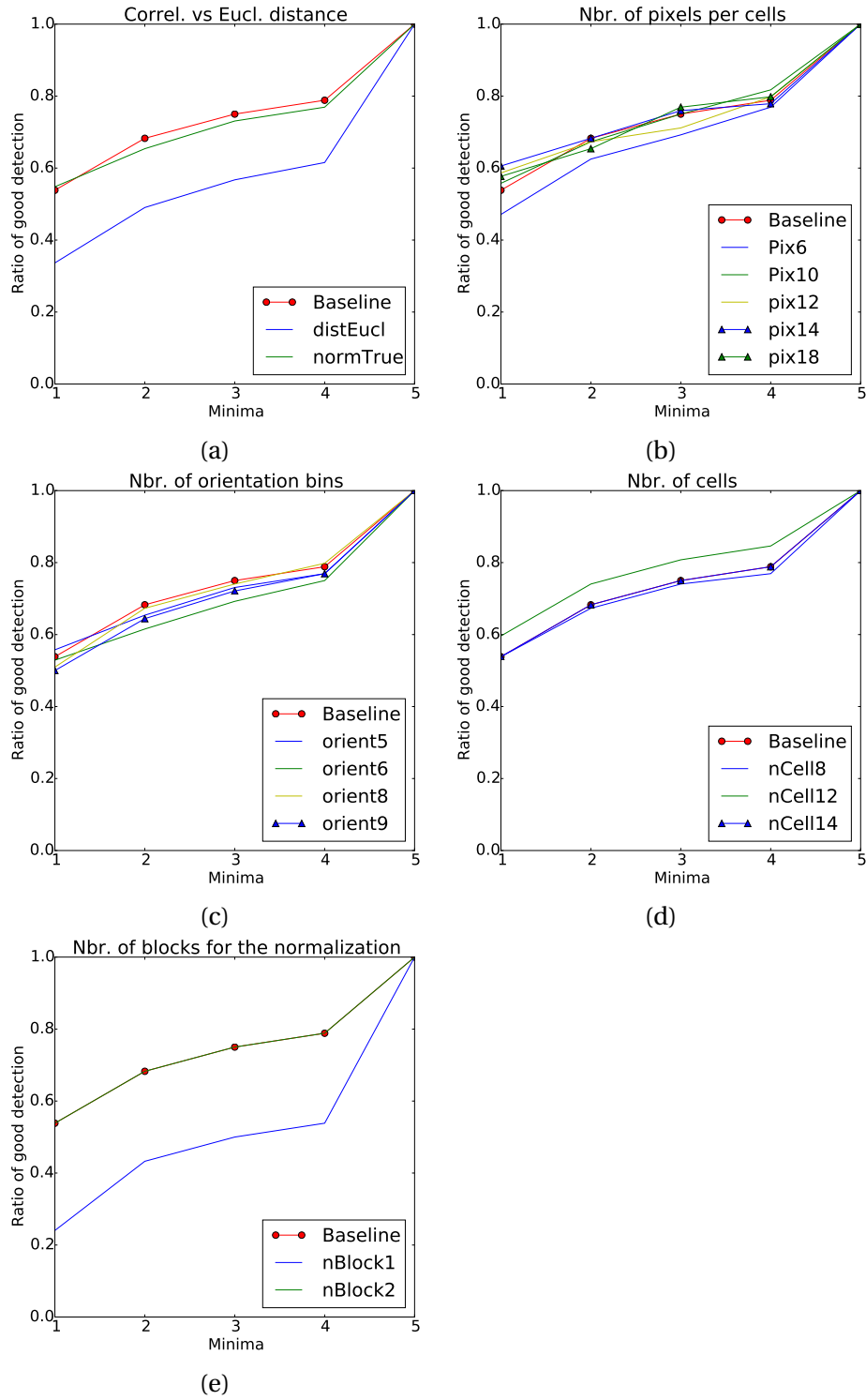


Figure 5.20 – Influence of the HOG parameters on the result of the global matching. (a) Euclidean vs Correlation-based distance (the baseline), (b) Number of pixels per cell, (c) Number of orientation bins, (d) Number of Cells, (e) Number of blocks for the normalization.

5.2. Detection of correspondences between real and synthetic images with HOG

matched (buildings, roads). Some successful images and mismatches are presented in Figure 5.21. The mismatched images have generally occlusion or more than 1/3 of foreground area.

From these studies, it is difficult to extract guidelines regarding the HOG parameters. It is clear that the correlation-based distance and the local normalization improve greatly the results. The impact of the number of pixels per cell and the number of bins are less clear. Specifically, the number of pixels per cell is a trade-off between the recall and the processing-time. Finally, our dataset is relatively small and small variations of the performances are not significant. Ideally, these parameters should be determined from an independent set of ground truth image. In our case, with around 100 images shot in the same area and representing only variation of light and shadows occurring during a summer day, we cannot decently generalize the choice of the parameters such as the number of bins or number of pixels to other datasets, other types of landscape or seasonal variations. This study is left for future extensions.

5.2.4.2 Local matching

In this second experiment, we test the performance of the local patches and the three constraints. The goal is to improve the first experiment by trying to be more robust to the influence of occlusion and foreground. The steps of the method are illustrated in Figure 5.18(b-c). First, we use the 3D coordinates of the panorama to generate three classes (sky, foreground and stable regions). Rather than detecting key-points, we create a regular and dense grid of points. The content of the patch surrounding a point is observed at each scale (300 to 150 with step of 25 pixels), if the ratio of sky is above a given threshold (70%) or if the point is in the foreground, it is discarded. Each patch is scaled according to the HOG parameters and is slid on the query image. Based on the result of the previous experiment, we chose to keep the three first local minima as potential matches. Finally, these correspondences are processed as explained in the previous section to take into account the geometric constraint.

By applying this strategy and keeping the same parameters for the HOG description (i.e. 10 cells of 8 pixels, histogram with 7 bins, normalization with 2 blocks), we improve the accuracy to 78% (versus 60% for the global matching). Among the successful images, one can find occluded landscapes (see the second column of the Figure 5.21. It proves that, as expected, this method is useful to orient images under occlusion. Moreover, by applying a 3DOF orientation at the end of the detection, this second method is also more accurate.

By comparing the results of both methods, it appears that some images (14%) are never matched properly. These images are characterized by backlight, large area of foreground or large occlusion. They are presented in the two last columns of Figure 5.21. Oppositely, it means that 86% of the images are recovered by at least one of these methods which is an encouraging result.

5.2.5 Discussion

In this section, we show that descriptors of image patches can match synthetic images with real images. In our experiments, we describe the orientations distribution within a patch of images with HOG. Hence, it implicitly takes into account morphological edges, but also discontinuities generated by the land cover. Our approach takes advantage of geographic, multiscale and geometric constraints to estimate correctly the 3DOF orientation of 78% of the images in our dataset. This dataset is relatively small and comprises around 100 images shot in a single region during two summer days. Since the orthoimage used to texture the DEM is also stemming from a summer campaign, our dataset and results

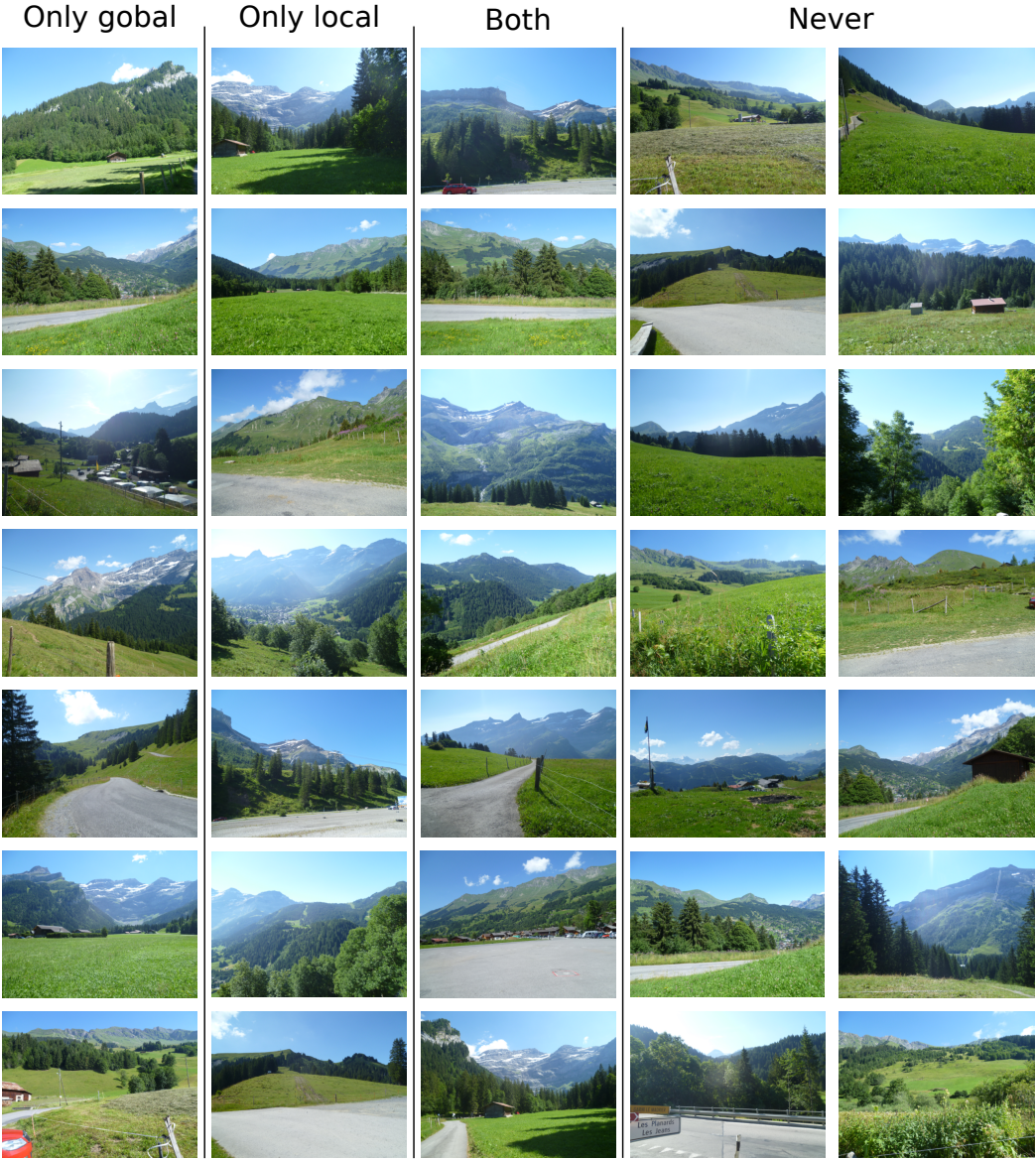


Figure 5.21 – Images of our dataset whose 3DOF orientation is recovered or not by our methods.

5.2. Detection of correspondences between real and synthetic images with HOG

are not representative of seasonal variations. However, it contains images with occlusion, foreground, slight roll and represent thus challenging conditions. The evaluation of our performance is limited by the lack of available benchmark dataset (discussed in more detail in Section 5.4.3). A benchmark would have allowed us to evaluate more rigorously our performance in comparison with other methods sharing a similar goal. Nevertheless, considering our dataset containing many challenging images, we consider the measured rate of positive detection as a promising result.

The next step would be the development of a similar approach in the context of unknown focal. Thus, the matching would have to be performed at several scales and should recover not only the best azimuth direction but also the best scale. This function is required to estimate the orientation of a picture if the focal or the location are approximate.

By a lack of time, we studied only the 3DOF orientation and we did not reproduce the experiment for a 6DOF orientation presented for the camera orientation with the skyline. Hence, the objectives are not entirely reached. However, in Produit et al. (2012), we described a method to retrieve a pose aligning a query image with the landscape model, it is summarized here:

1. A prior orientation and location is provided (rough azimuth given by the author, the location is a user-click on a map), it gives the parameters to generate a synthetic reference image.
2. Key-points are detected in the query and the synthetic image.
3. The key-points are matched with Normalized Cross Correlation and the pose estimated with RANSAC.
4. A new reference image is generated. The detection of the correspondences and the pose estimation are iterated.
5. Throughout the iteration, a geometric constraint is limiting the 2D key-points potentially matchable with a 3D key-point.

This method has several important weaknesses, that is why, we choose to not present it in this thesis without consolidations. First, in our experiment with the dataset presented in this Chapter, it appears that the detection of key-points with usual detectors can hardly highlights similar locations in real and synthetic images (other than peaks in the skylines). Second, the successive pose estimates with RANSAC are very impacted by outliers and thus the convergence is not guaranteed. Finally, the geometric constraint was not defined rigorously and neither took into account the covariance of the pose nor the 3D coordinate of the reference key-points. Hence, both the pose estimation and the geometric constraint would greatly benefit from PEP-ALP (discussed in Section 3.5.3 and implemented in the next section). For the correspondence detections, we would rather detect key-points in the 3D synthetic images, project them in the query image with their confidence ellipse and considering every location in the ellipse detect the best match. This solution can work only if the prior pose indicates a landscape region which overlaps the query image (for instance, in the context of the supervised georeferencing, a prior pose generated from the browsing of the virtual globe or a geotag plus an approximate azimuth.)

For an image with a looser prior, two approaches based on HOG features could be considered. The first would be a dense matching using the approach of the second experiment in which a panorama is generated for each location of a regular grid covering the area of interest rather than a single location. The second would be to reproduce the work of Aubry et al. (2014) in the conditions of landscape images (detection and matching of discriminative patches with an exemplar-SVM). At this stage, neither

method is compatible with an online or a large area processing unless a more robust descriptor to match real images with synthetic images is found.

5.2.6 Conclusion

For the georeferencing of single images of landscapes, methods independent from the reconstruction of camera poses from a set of images should be developed. This strategy is required to provide a solution in regions and collections where the overlap is weak and the detection of tie-points difficult. The registration with a landscape model is then desirable to georeference directly single images. Moreover, SfM-based 3D reconstructions also need to be georeferenced and could thus benefit from the alignment with a landscape model. Methods proposed until now specifically dedicated to landscape images are based on the skyline, but they face problems to deal with occlusion of the skyline (weather conditions, vegetation etc.) and provide 2D-3D correspondences aligned on the top of the picture which is a bad configuration for pose estimation.

In this section, we proposed to match real images with 3D synthetic images made from a DEM textured with an orthoimage and thus benefit not only from the morphology but also from land cover similarities. We described real and synthetic image patches with a HOG descriptor and they were matched with a correlation-based distance. Local patches matched according to geographic, multiscale and geometric constraints proved to limit the influence of occlusion and to be more compatible with 2D-3D pose estimation methods. For example, the local patches matching was successfully applied to recover the 3DOF orientation of images having the background occluded by building or vegetation. It would also be desirable to ensure the validity of the methods for pictures of other seasons or even for historical photographs. In this section, we only presented results for 3DOF orientation, in a future work we would implement the 6DOF orientation for images having a prior on the location and the orientation. In order to achieve this, we already have described PEP-ALP, which would perfectly fit our problematic and is implemented in the next section.

5.3 Georeferencing an image collection

This Section is a description of a platform dedicated to the registration of collections of landscape images. It will act as an integration of the methods presented in the previous sections. Firstly, some images are directly georeferenced by finding correspondences with the 3D landscape model. In this case study, we will use the skyline matching (presented in section 5.1) to orient GPS-located images, but we could have considered HOG matching (section 5.2) or even the monoplottter, if the two other methods are not exploitable. Secondly, we will automatically extend the georeferencing to other images, overlapping the previously georeferenced ones. We apply the PEP-ALP method presented in Section 3.5.3. Its goal is to use the pose and landscape model priors to constrain the key-points matching and thus be more robust and less sensitive to the various appearance of the landscape. With this approach, we propose a solution for a large variety of image collections and we limit the user supervision. The workflow that is presented here is, excluding small modifications, the one that was presented in Produit et al. (2014b). With respect to this previous work, we provide a more general discussion centered on the integrated georeferencing platform.

5.3.1 Introduction

In the previous chapters, we discussed the georeferencing of landscape images with DEM and textured DEM. The matching of a synthetic image with a real image is challenged by the distortions generated by the difference in the viewing angles between orthoimages and oblique images. Even by draping the orthoimages on the DEM, distortions are still visible on steep slopes and in regions close to the camera. A straightforward solution is then to use as reference oblique images rather than orthoimages. Unfortunately, if one easily finds orthoimages covering the entire world, accurately georeferenced oblique images in rural areas are still rare. For this reason, we propose to explore the images shared by the general public. Hopefully, in this huge database some partially georeferenced pictures can be found and used as the reference after a refinement of their orientation. Three ingredients are needed to use landscape images as reference:

1. First, the reference images have to be accurately georeferenced. An increasing number of images are accurately localized with a GPS and can even have a prior of their azimuth provided by the compass (currently, most of the pictures shared on the web are shot with smartphones according to Flickr statistics, flickr.com 📷). Moreover, we have already described skyline-based and synthetic images-based methods to compute the 3DOF orientation of such pictures in Section 5.1 and 5.2, respectively.
2. Second, we need overlapping images. An analysis of landscape image databases shows that most of the pictures shared have at least a small overlap with other images. Indeed, famous areas are recorded by several photographers, and a photographer usually takes several pictures of a same landscape from slightly different locations and angles. We will study the spatial distribution of pictures in more details in Chapter 6, as other authors, we can notice that few locations are very attractive and the large part of the remaining locations is sparsely covered by photographs (see Figure 6.1, p. 112). Hence, the workflow that we propose will not be applicable in neglected regions, where no reference images are found or where images do not overlap.
3. Finally, we need a key-point detector and descriptor to match the reference images with the query image. During the last decade, computer vision and photogrammetry have greatly benefited from the emergence of key-point detectors and descriptors such as SIFT, which are extremely

good at detecting similar key-points in multiple views provided that the viewpoint and the landscape appearance are sufficiently similar. Hence, we developed PEP-ALP, a method which uses a pose and a landscape model prior to improve the matching.

Exploiting these three facts, we have all the material to use shared terrestrial images as reference.

The usual workflow applied to georeference such collections of overlapping images is the following: during key-point detection and matching, SIFT-like features are detected and matched creating a web of pictures connected with key-points; then a bundle adjustment recovers the camera orientation and the 3D coordinates of the key-points in a relative coordinates system; finally, the absolute orientation and the camera exterior orientation relative to a geographic coordinate system are recovered by inserting GCP in the process. This workflow, called Structure-from-Motion in the computer vision literature, was successfully applied to reconstruct 3D models of monuments (Agarwal et al., 2011; Snavely et al., 2006) and is now a standard for the generation of 3D models in photogrammetry and computer vision.

However, if one wants to apply directly this workflow to a collection of landscape images, some issues will be encountered.

First, SfM will not connect all the images together, it will generate several clusters of images corresponding to various seasons and illuminations, and to various viewpoints (Strecha et al., 2010). Indeed, unlike a collection of pictures issued from a photogrammetry campaign or web-shared pictures of famous monuments, landscape pictures typically have less spatial and temporal overlap. In this condition, each cluster is separately reconstructed and the clusters are not linked together. Moreover, pictures badly connected or too different from the others are left out.

Second, bundle adjustment requires some Ground Control Points (GCP) to match the SfM relative coordinate system with the world coordinates. Usually, GCP are particular objects easily recognizable in the pictures and measured by GPS in the field. For the georeferencing of a collection of landscape images, the most likely GCP workflow would be an operator clicking on similar features in the pictures and in a georeferenced layer (map or orthoimage) as presented in Chapter 4. Alternatively, the geotags, if there are any, can provide the initial georeference.


Finally, the SfM-based workflow does not suit well the problematic of evolving image databases. Indeed, for a new image inserted in a cluster, it would require the processing of the entire cluster.

5.3.2 Review

The review of the existing methods was already discussed in Chapter 2. These methods can be separated into two families of processing. On one hand, there are methods based on SfM, which require multiple overlapping views of an object (Agarwal et al., 2011; Martinec and Pajdla, 2007; Snavely et al., 2006) and on the other hand, there are methods based on the matching of a single picture with a georeferenced landscape model (Baboud et al., 2011; Baatz et al., 2012a; Chippendale et al., 2008). In addition, some methods can be placed between these two groups. The advantages of these combined methods are that they avoid the SfM processing of the entire collection for the registration of a new image. For instance, Li et al. (2012); Sattler et al. (2011, 2012) propose to find matches between a query image and a database of 3D key-points issued from a SfM. Next, 2D-3D pose estimation can be applied to compute the pose of the new image. In this scheme, the number of 3D features is typically large and robust methods for the correspondences detection are required (Svärm et al., 2014).

Considering a database of images with geotags, Strecha et al. (2010) propose a different method which has several advantages. The SfM is applied to small clusters of similar images. Each 3D re-

construction created from a cluster is registered to a GIS model and thus ensures that every 3D reconstruction make up a single, consistent model. To do so, the initial georeference is extracted from the geotags and the finer registration is based on an alignment of the Z direction (which is predominant in cities because of the façades) and the alignment of the 3D model with the buildings footprint.

Finally, outside academia, Google (*Google Maps*, *Google Inc*, *Mountainview*, 2014) with the photo tours , was providing a solution close to the one presented in this case study (but seem to have been suppressed from Google Map lately). By the appearance of this product, SfM has been applied on a dataset similar to ours. An animation illustrates the navigation within a rough 3D reconstruction of the pictures. The user can also navigate through the images. The main drawback of this approach is that the images are not inserted in a georeferenced system and no interaction with geographic layers is possible.

The model of Strecha et al. (2010) is the closest to the ideal georeferencing process for a database of landscape images. However, the constraints used to align the 3D reconstruction with the geographic coordinate system are not applicable in rural areas: firstly, vertical planes are not predominant and secondly, fewer images are geotagged and they have more seasonal and illumination variations. They would result in tiny clusters. However, we can be inspired by the use of GIS data as reference. For instance, in rural areas, 3D models of the landscape are available (DEM, DSM). They are accurate and, except under some particular circumstances (glacier, landslide), do not evolve much.

Our goal is to develop a method independent from SfM for several reasons. First, we want to be easily compatible with operator supervision (which can be compulsory for the most challenging images) and thus the processing needs to be fast. Second, we want that a single georeferenced picture can be a reference for other images even if they have only a small overlapping region. Third, we want to compute the image pose directly in a geographic coordinate system and thus:

- enable landscape model matching,
- facilitate the connection of all the images,
- incorporate the knowledge of the 3D model to relax the measurement of the similarity of key-points.

5.3.3 Method

To georeference images, we need first a reference, which is assumed to be reliable. We will resort to two types of reference data: first, a DEM of the area and, second, images localized with a GPS sensor. Hence, as illustrated in Figure 5.22, the set of images to be georeferenced is divided into two subsets: GPS images, which would subsequently require only a 3DOF orientation, and all the remaining images, which have a less accurate geotag. The GPS reference images are oriented by skyline-matching as presented in Section 5.1. The 3D geographic coordinates of each of their pixels are computed by projecting them onto the DEM.

Second, all the remaining query images are matched with key-points of the reference images one at a time. We apply PEP-ALP (presented in Section 3.5.3). With this technique, the 3D surface and the pose prior reduce the possible correspondences and thus the matching becomes more robust and less impacted by the various quality of the images.

In the next section, we will discuss mainly the pose estimation of the query images, since the 3DOF

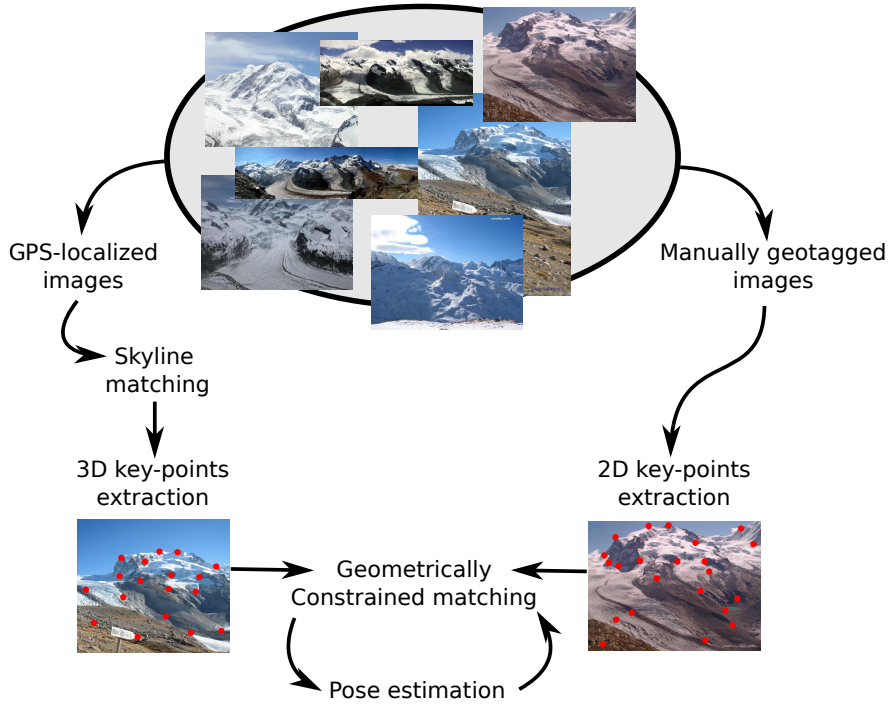


Figure 5.22 – The set of images is divided into two subsets. On the left, the GPS images are oriented with the skyline and serve as reference. On the right, the 2D key-points of the manually geotagged images are matched with the 3D key-points of the reference images.

orientation with skyline was already presented in Section 5.1.

5.3.3.1 Pose estimation of GPS-localized images

As a reminder of the skyline matching discussed in section 5.1, we present its general workflow in Figure 5.23. These images have a precise location $\mathbf{t} = [t_X, t_Y, t_Z]$, but no orientation $\mathbf{r} = [\alpha, \beta, \gamma]$. The orientation \mathbf{r} is recovered by skyline matching. Finally, the 3D coordinate of each pixel is computed by projecting the pixels on the DEM (the distance image is presented as an illustration of the coordinates in Figure 5.24 (b)).

5.3.3.2 Pose estimation of a query image with uncertain geotags

The basic ingredient is PEP-ALP, which was already detailed in Section 3.5.3. PEP-ALP supposes that a prior orientation and location can be provided by the user. The prior has two impacts. First, it is the initialisation of the pose estimation with the Kalman filter and hence improve the robustness of the pose estimation. Second, it limits the potential region of the query image which can be matched with a 3D key-point. This geometric constraint takes the form of an error ellipse and allows the similarity threshold to be relaxed and thus gain 2D-3D correspondences.

The workflow is summarized in Figure 5.25. In the first part, the geometric constraint requires an estimation of the camera location \mathbf{t}_0 (which is obtained with the geotag) and an estimation of the orientation \mathbf{r}_0 . To compute the latter, SIFT key-points are extracted from the collection of reference

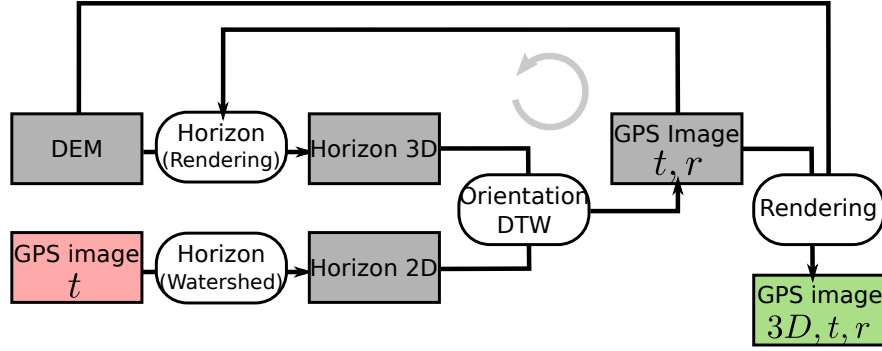


Figure 5.23 – Processing of the GPS images, t refers to the camera location measured with the GPS, r is the orientation of the camera. At the end of this step, each GPS-image pixel owns a 3D coordinate.

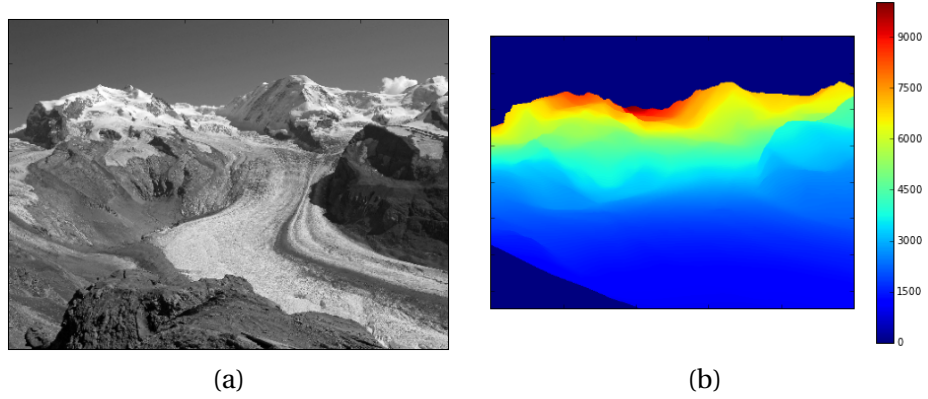


Figure 5.24 – (a) One of the reference GPS images; (b) Corresponding distance matrix: each pixel is colored as a function of its distance to the camera (in meters).

GPS images. The key-points from every reference image are merged into one set of n features R_i with corresponding 3D coordinates X_{R_i} and SIFT descriptions d_{R_i} .

Computation of \mathbf{r}_0 : Given a query image, a set of m features Q_i are computed and each of them has an image coordinate u_{Q_i} and a description d_{Q_i} . SIFT features are firstly matched according to the ratio of the two closest features descriptions found in the other set (Nearest Neighbour Distance Ratio, NNDR). Using the NNDR, a set of 2D-3D correspondences is extracted. Those first matches are used to compute the initial camera orientation with RANSAC (see Section 3.5.1) in conjunction with the 3DOF camera orientation model (the camera is initially fixed at the geotag location t_0). These steps are illustrated in Figure 5.26. Typically, few and poorly distributed matches are found, and this is why we do not use RANSAC in association with the 6DOF camera orientation model.

PEP-ALP: The pose p_0 is associated with a high standard deviation Σ_{p_0} and the matches are associated with noise Σ_{u_0} . During the second part, corresponding to the geometric matching, the reference features X_{R_i} are projected in the query image, following the Equation 3.5. The variance propagation presented in Equation 3.26 is used to compute the corresponding covariance matrix Σ_{R_u} . Based on the covariance, ellipses are drawn in the query images and used to constrain the SIFT feature matching, as illustrated in Figure 5.26(b-c). The Kalman filter prerequisites are met and with each new block of 2D-3D correspondences, a new pose is computed. During the iterations, the SIFT distance threshold

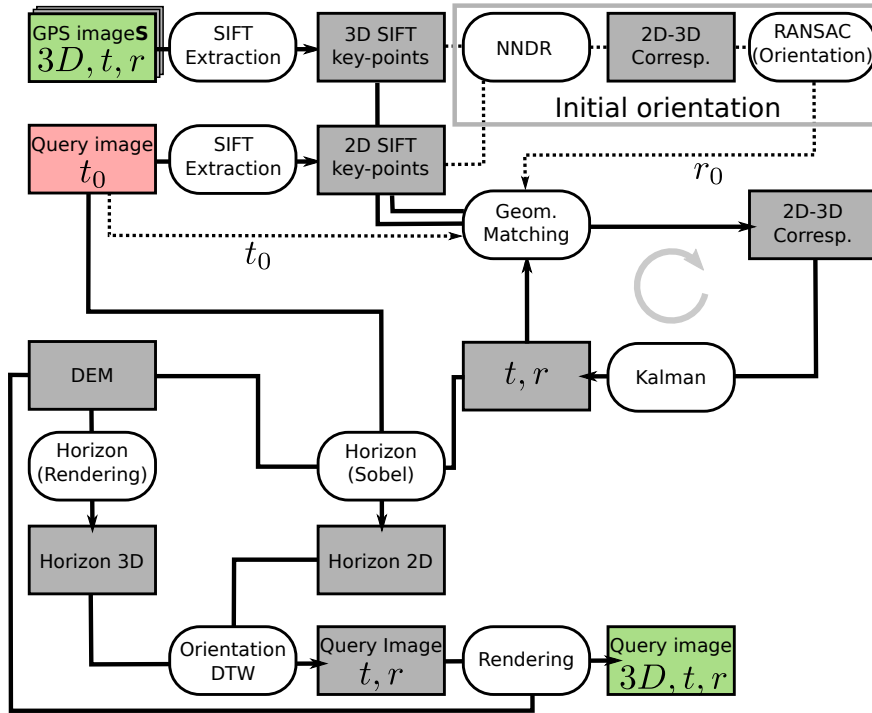


Figure 5.25 – Processing of the query images, t^0 refers to the geotag, r is the orientation of the camera.

between two key-point descriptions is relaxed to take into account texture and illumination variations.

Skyline refinement : At this stage, the estimated pose is quite accurate, but the alignment with the 3D model is still not exact because of the pose inaccuracy of some reference images and of accumulated errors. Optionally, we propose to update the pose with an additional horizon alignment. This time, the current pose and its covariance are used to delineate the region in the query image where the skyline should be located. Then, a Sobel edge extractor is applied in this region and the DTW algorithm, presented in Section 5.1 is used to refine the orientation, as shown in Figure 5.26(d).

At the end of the process, the query image is draped on the DEM to generate an orthorectified image that can be used in a GIS (Figure 5.26(f)). The pseudo-algorithm is presented in Appendix F.

5.3.4 Experiment

With this experiment, we want to validate the objectives and hypotheses presented in Section 1.4 (p. 7). Specifically, we want to integrate some methods presented in this thesis (skyline matching and PEP-ALP) for the georeferencing of a collection of various (in terms of landscape appearance and pose prior accuracy) pictures. We will also estimate the accuracy of the georeferencing, which is an indicator of the applications that can be derived from the proposed methodology.

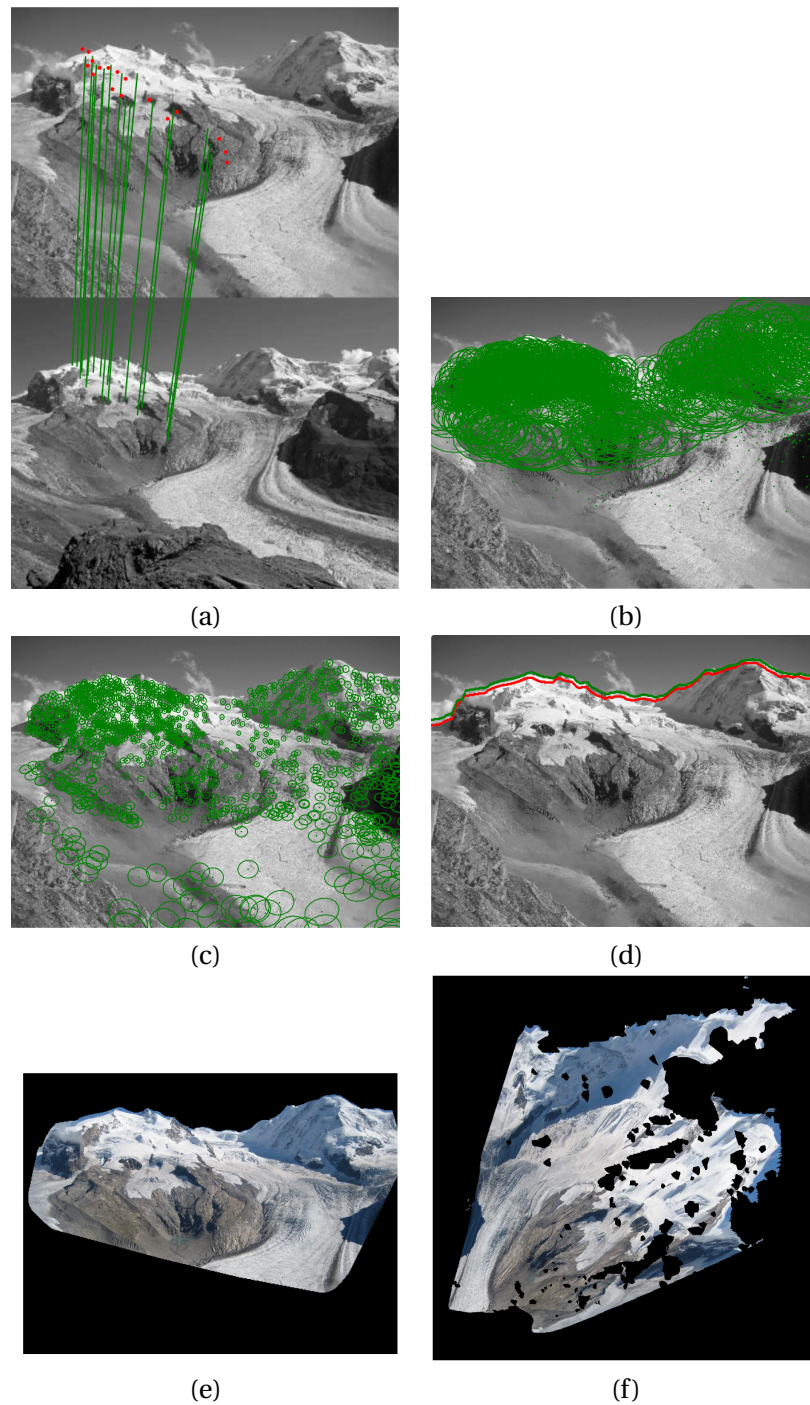


Figure 5.26 – Representation of the query images pose estimation workflow: (a) Initial matches found after the NNDR matching and RANSAC orientation. (b) Covariance ellipses at the first iteration. (c) Covariance ellipses after 10 iterations. (d) Horizon projected in the image before (red) and after (green) DTW matching. (e) Query picture masked with DEM and the SIFT features (only the area within the convex hull of the matched SIFT is conserved). (f) Query picture orthorectified.

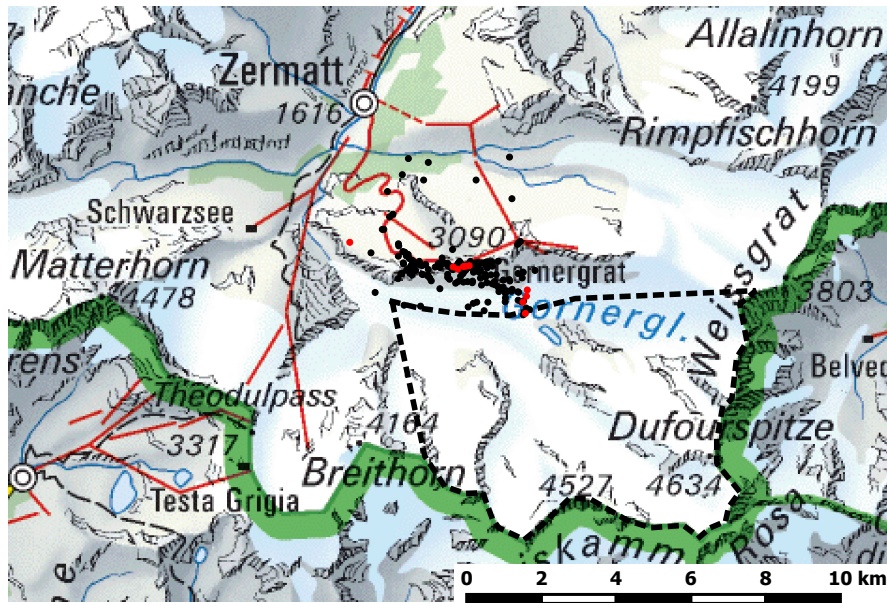


Figure 5.27 – Zermatt area. The red dots represent the GPS images location (the references), while black dots are the images with geotags only. The dashed line encloses the area of interest. Most of the images is shot from the Gornergrat ridge. (*Swisstopo*)

5.3.4.1 Description of the data

The area of interest is located in the southern Swiss Alps in the surroundings of the famous Zermatt ski resort. From Zermatt, a train brings people to the Gornergrat, a ski and hiking region in the middle of the highest Alps peaks. From this area, there is a great view of the Matterhorn and the Dufourspitze, which are among the most famous Swiss peaks. Many pictures shot in the area represent one of those two peaks. Figure 5.27 shows our case study area and the location of the images within our set extracted from the photo-sharing platform Panoramio 📍. This platform is mainly dedicated to "Photos of the world" and thus oriented to landscape scenery. On this platform, the assignment of a location to the picture is straightforward if it is shot with GPS-camera. Otherwise, the user can also indicate a place with a click on a map or only provide a toponym. Hence, the accuracy of the geotags is very heterogeneous. For instance, in the map of Figure 5.27, some inaccurate geotags are those located in the very North from where the area of interest is not visible.

In order to compute the camera pose correctly, the focal length must be estimated. A good estimate can be computed from the focal stored in the image metadata and from the sensor size, which is found in online camera databases. If the image metadata is incomplete, a normal lens is assumed (focal length equal to the diagonal of the image). In total, our set of images pointing to the area of interest is composed of 198 images, among which 10 have a GPS location and are thus the reference images. 118 images have the focal stored in the metadata, for the remaining 80 images a normal focal is assumed.

To compute both the skyline and the pixels world coordinate, a DEM is needed. We use the one provided by the Swiss Office of Topography, which has a 25m pixel size. Indeed, our focus is on objects far from the camera (beyond 1km), which makes the use of a higher-resolution DEM unnecessary. Some height inaccuracies are expected in glacier areas, since glaciers are dynamic and their footprint and height vary in time. However, the influence of the height inaccuracies is assumed to be smaller

than other perturbations. For the height differences to influence the pose estimation, it would require that every 2D-3D correspondences match exactly. However, the reference images are not expected to be perfectly georeferenced. Moreover, the key-point detection already comes with small shifts, which generate Z errors larger than the ones that can be found in the DEM.

A state of the art orthoimage with a 0.5m resolution also provided by Swisstopo is used to assess the accuracy of the georeferencing.

5.3.4.2 Results

Georeferencing accuracy To validate the success of the proposed workflow, we first measure the accuracy of the pixel localization. Indeed, our method minimizes the reprojection error (i.e the distance between a 3D key-point projected in the image and the corresponding 2D location), but because of distortions not taken into account explicitly (focal, principal point location, image distortion) and false positive matches, the estimated pose may differ from the real one, while still providing good pixel localization close to the correspondences. To measure the accuracy of the pixel localization, we applied the following scheme: the images are orthorectified with the DEM (i.e. according to the pose each pixel is projected on a triangulation of the DEM to retrieve its corresponding map coordinates), thus resulting in orthorectified images. Distances are then measured between similar landmarks found on both the rectified images and on an official orthoimage (accuracy < 1m). In the following, we will discuss the accuracy of the GPS images used as reference and then the accuracy of the remaining query images.

GPS images For each image, between 5 and 15 planar distances are measured and statistics are presented in Table 5.1. Two orientation methods are compared: horizon matching with DTW and user-defined GCP. The GCP digitization accuracy depends on the resolution of the oblique image. To provide a fair comparison, we used images with 500 pixel width for both methods.

First, we comment the user interaction required for both methods. It is much easier for a user to provide the initialization region of the horizon, (illustrated in Figure 5.3) than to detect precise GCP. Indeed, and specifically for this study case, it is not obvious to find similar rocks or cracks in oblique images and orthoimages. This reinforces the conclusion presented in chapter 5.1: matching linear features is a valuable improvement for monoplotters.

Second, we compare their accuracy: from Table 5.1, we observe no sensible difference in the accuracy depending on the orientation method. On average, the accuracy of the pixel localization is better than 50m, which is good according to the image resolution and the geometry of the problem (which was discussed in Chapter 4 and Figure 4.4). In particular, the ray is almost tangential in flat areas and close to the horizon. In these regions, a small inaccuracy of the pose generates large distortions. The worst accuracy observed, for both DTW and GCP, is measured for the same image (24694380), which may also indicate that either the GPS location or the focal estimation is incorrect. Images presented in Table 5.1 have easily extractable horizon (sky without clouds): the performances tend to decrease when clouds occlude the horizon.

Query image pose estimation Once the pose and the 3D coordinates have been estimated for all the GPS images, we can use them as references for the remaining query images. A pose is computed for the query images, for which RANSAC can find initial matches (100 images over the 188 query images).

Picture ID	DTW/GCP		
	Min. [m]	Max. [m]	Avg. [m]
24694380	62 / 32	289 / 447	149 / 156
47903616	6 / 6	73 / 76	44 / 40
58907261	10 / 25	76 / 96	44 / 58
65319402	13 / 10	57 / 70	36 / 41

Table 5.1 – Distances measured between recognizable landmarks found both in an orthoimage and in rectified **reference GPS** images. Reference images were oriented either with DTW (left) or GCP (right) and they are presented in Figure 5.28.

Picture ID	Min. dist. [m]	Max. dist. [m]	Av. dist. [m]
14653410	21	343	139
39408227	24	406	148
42359812	28	223	84
54833218	18	239	81
78060652	58	207	98
87063176	14	102	39
96861787	44	538	196

Table 5.2 – Distances measured between recognisable landmarks found on an orthoimage and on **rectified query images**. Those images are also those presented in Figure 5.28.

However, some poses are clearly incorrect (10 images): these incorrect poses are associated with images, for which RANSAC returns only false positives. At the end of the pose estimation process, inexact poses are also computed for images with few correspondences irregularly distributed in the image. These poorly distributed correspondences generate large uncertainty in areas without correspondences. However, by erasing image areas without correspondences, we somehow avoid very large distortions. Finally, the horizon matching step applied to refine the pose is useful for images without clouds, but may suffer from undesirable effects in the presence of clouds (see the discussion in the previous section). Statistics for seven query images are presented in Table 5.2. As expected, the accuracy decreases compared to the reference images. The bad geometric configuration generates even larger distortions in some regions of the image close to silhouette break lines.

Pose accuracy To assess the accuracy of the pose itself, we computed as reference the orientation of one image with GCP (the image is presented in Figure 5.28). Then, we started the pose estimation using a location shifted away from the "real" location estimated with GCP. For each shift distance, 10 pose estimations are conducted from different locations. The mean and standard deviation of the absolute difference between the computed parameters and the one obtained with GCP are presented in Table 5.3. In the table, we also summed the number of pose locations within a 100m radius of the ground truth location, they are reported between parentheses.

It appears that most of the poses started within a 500m radius reach a local minimum. We can see it also from the small standard deviations in ΔXY . This minimum is not centered on the real location, but 100m away and this shift can be explained by the registration errors of the reference images and some false positives detected. Beyond this threshold, i.e. for distances from 500m to 2km, the variance increases, and some poses do not converge to the minimum. Beyond 2km, computed poses hardly

5.3. Georeferencing an image collection

Mean	Dist.	ΔXY (# < 100m)	ΔZ	$\Delta \text{head.}$	Δtilt	Δroll
	50	85.5 (9)	14.7	0.4	0.5	0.5
	200	72.2 (10)	10.8	0.2	0.2	0.1
	500	85.3 (10)	35.9	0.4	0.7	0.5
	1000	232.7 (3)	95.6	1.7	1.3	1
	1500	227 (7)	132.6	1.7	1.4	0.8
	2000	253.2 (4)	144.4	2.1	1.9	2
	3000	2992.3 (0)	367.5	21.9	7.4	2.4
Std. dev.	Dist.	ΔXY	ΔZ	$\Delta \text{head.}$	Δtilt	Δroll
	50	15.6	7.2	0.3	0.6	0.8
	200	18.2	1.7	0.2	0.1	0.1
	500	14.9	46	0.3	0.9	1.
	1000	129.2	51.4	1.5	0.7	1.
	1500	275.5	179.2	3.4	1.7	1.
	2000	280.1	189.3	2.8	2.4	0.4
	3000	1877.1	540.9	13.5	7.8	0.4

Table 5.3 – Impact of the geotag on the pose accuracy. For a same image, the initialisation location is generated at different distances. Distances are measured in meters, angles in degrees.

converge. This show the importance of selecting reference images or, at least, to have a reference image in proximity (corresponding thus to a similar pose and orientation).

Visual assessment A video is available on the following link <https://youtu.be/87dHVDdlPSs>¹. In this video, the 100 images for which the pose was estimated are rendered on a shaded 3D model of the area, including those with poor matching. Holes in the reconstructed surface correspond to regions of the map which are hidden from the camera position. Quite often, patches of sky are visible close to the horizon; this illustrates the misregistration problem due to tangential geometry. An attentive screening will also show foreground objects projected on the background (a man with a hat, a bird on a fence, a lake). However, at the scale represented in this video, the registration is usually of good quality and at least represents a great improvement compared to a simple geotag.

5.3.5 Discussion

3DOF orientation of GPS reference images: The proposed workflow is composed of four stages. The first is the 3DOF orientation of images shot with a GPS-camera to texture the DEM with terrestrial images rather than orthoimages as presented in Section 5.2. We used the DTW-based skyline matching presented in Section 5.1. By applying this method, which involves the user, we ensure that the reference images are sufficiently well georeferenced. Indeed, we have few GPS-acquired images and we want to have all of them available as the reference. In term of time spent and skills, less than 20 seconds are required to provide a very good watershed initialization, while the digitization of GCP will require a skilled operator, a GIS and, for each GCP, at least the same amount of time as the one spent for the sky segmentation. The area studied in this application is perfect for horizon matching thanks to the

¹<https://youtu.be/87dHVDdlPSs>

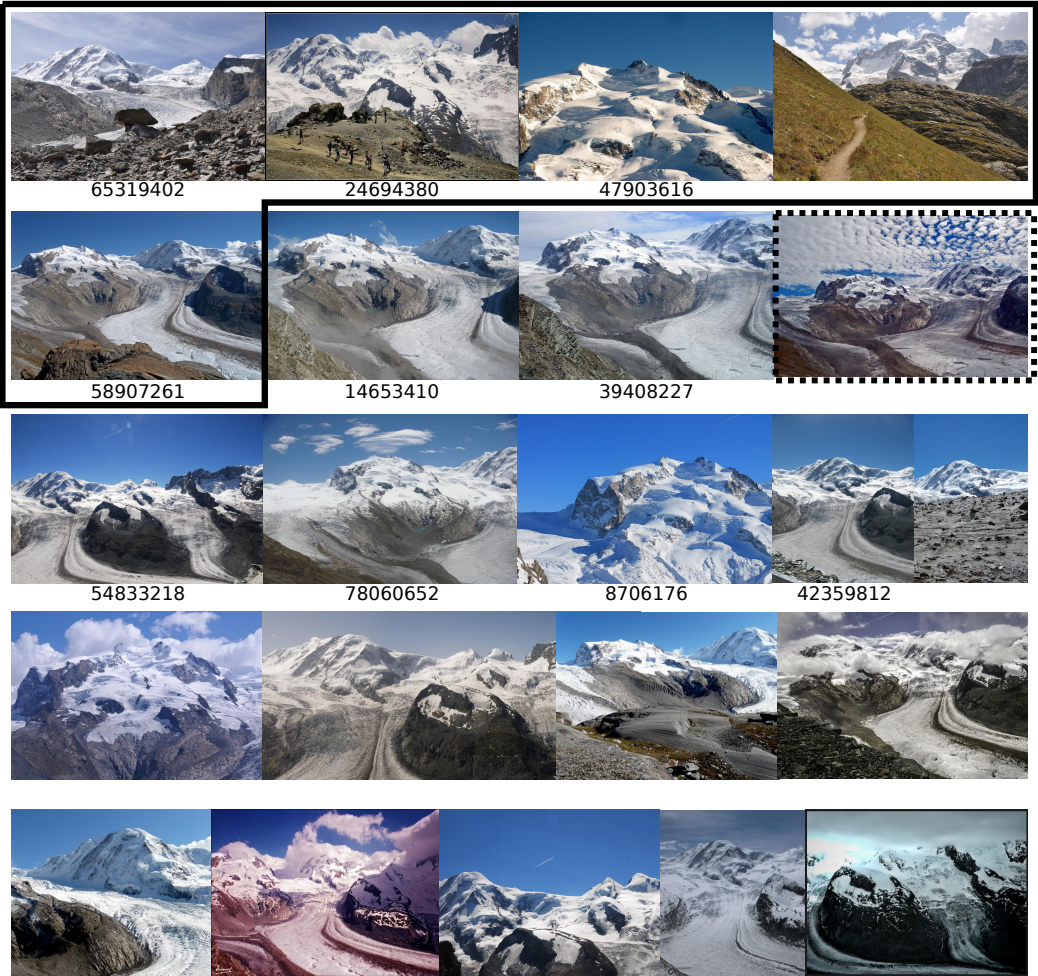


Figure 5.28 – Images used to compute statistics in Tables 5.1 and 5.2. GPS images are within the box, numbers correspond to the ones in the tables. The image within the dashed box is used for the statistic in Table 5.3 Unlabelled images are randomly selected in the dataset.

mountains that provide specific silhouettes. However, in the presence of other kinds of silhouettes (flat or with repetitive shapes) or in the presence of foreground objects perturbing the horizon (trees, buildings) DTW may fail. The 3DOF orientation with HOG features presented in Section 5.2 would be then a valid and automatic alternative to be applied in these cases. Moreover, if the images are particularly difficult to match automatically, the user can still constrain the matching with an azimuth range or provide some GCP. With this method, we assume that the GPS location stored in the metadata is exact. However, if the GPS is initializing during the picture capture, the GPS location could be worse than expected.

6DOF orientation of query images: The second stage is the 6DOF orientation of other images. This step is both a way to gain more images to be used as reference and a way to georeference query images. Its input is a pose prior, namely a **location** and an **orientation**.

The **location** part of the *apriori* orientation is based on the geotag provided by the user. Some web applications store the zoom level applied on the map by the user when clicking as a measure of the localization accuracy. However, the relation between the zoom and the accuracy is not straightforward and here we assumed for every geotag a standard deviation of 1000m. Zielstra and Hochmair (2013) studied image geotag accuracies for several types of landscapes and several areas of the world. We can learn from this work that in natural landscapes, users provide location much better than that.

For the **orientation**, we assume a horizontal camera (tilt and roll equal to zero) whereas the azimuth is estimated with a method strongly dependent on SIFT. This solution was privileged to reach an automated process, which is pertinent for the images most similar to the reference images. However, it compromises also our goal to provide a method less impacted by the various appearances of a landscape. At this level, we will consider three modifications:

- Ask the user to provide the azimuth, by providing him a 360° panorama, this step is fast and remains simple for the user.
- Estimate the azimuth with landscape matching. Both HOG matching and skyline matching are less sensitive to the various landscape appearances than SIFT.

Next, the picture are matched with **PEP-ALP** and relaxed SIFT matches. The main limitation here is that an overlap between the reference images and the query image is required. If the overlap is small or if the reference images are not perfectly oriented, the orientation of the query image is inaccurate. Our further development will consider the insertion of correspondences detected with a synthetic image generated from the DEM and rendered with a 3D image. It will provide matches more regularly distributed in the query image and not impacted by the accuracy of the reference images.

At this stage, more than 50% of the images are correctly georeferenced. We could increase easily this percentage by running a second pass of the process (using not only GPS images as reference, but also the newly georeferenced ones). The risk at this level is that the accuracy decreases (generally the alignment of the query image is worse than the reference images).

Left out images: Some images are not oriented. The bottleneck of our process is the initial estimation of the azimuth. We mentioned two solutions to be explored. The first involves the user, which is not necessarily a major issue per se in the context of supervised georeferencing. Moreover, this initial involvement provides a new reference image. Finally, an extension of this work would be to use the same processing for images without any geotags. A simple solution would use the correspondences detected between the query image and the most similar reference image as 2D-3D correspondences for the pose estimation with a closed-form solution.

Accuracy: In general, the accuracy around detected correspondences is coherent to an apriori expected accuracy (<50m). For this case study, we used low-resolution images, and improvements may still be observable by considering full resolution images. Accuracy tends to decrease in image regions where no or false correspondences are found (for instance close to the image borders) and in sinuous parts of the DEM, which generate large distortions during the orthorectification of the query images. To digitize the GCP used to compute the statistics, we choose the best quality images, which also often correspond to the images most reliably oriented. We can then imagine that the accuracy within the whole set could slightly decrease. Considering our setup, which involves very differing images shot with uncalibrated camera and rough pose, the results meet the expected accuracy.

Focal length: In our experiment, we estimate the focal length from the picture metadata. If unavailable, we assume a normal focal. An improvement of the actual PEP-ALP algorithm would be the possibility to relax also the focal parameter and thus solve the problem of 7DOF with priors.

Applications: One of the motivations for computing the pose of landscape images is related to environmental monitoring. To be effective, such monitoring requires a very high accuracy of the measurements of natural objects position and movements. According to statistics presented in Tables 5.1 and 5.2, the accuracy is not sufficient for environmental studies in most of the images. However, by providing our estimation as an initial pose, the task of manually georeferencing images with GCP becomes strongly facilitated. Moreover, once some GCP are provided by the user, we can take advantage of PEP-ALP and get more restrictive confidence ellipses, in order to propose more correspondences and a final better pose. Most of the database of landscape images are currently lists of images, sometimes associated with a map location. The orientation computed with our workflow could be useful in several scenarios to create more user-friendly image database browsers. First, the location and heading measured are accurate enough to detect visible points of interest and associate the images with appropriate toponyms and hence provide improved search facilities. Computed poses can also be used to overlay toponyms and other geographic layers in the images (as the mountain names). Finally, the visual alignment of the images and the landscape model is good and thus the images could be inserted into a virtual globe for an interactive 3D browsing of the collection. Summing up, we are not precise enough for precise spatial studies, but achieve the required level of detail to insert the images in a georeferenced database having the purpose to perform spatial queries and to augment the images with geographic information.

5.3.6 Conclusion

In this section, we presented a workflow to estimate the pose of a set of landscape images downloaded from a photo-sharing platform. Such a workflow is necessary to contextualize and extract information from pictures. These sets of images are characterized by a very approximate geolocation, by a low spatial coverage and by various landscape appearance. This setup is not limited to web-shared oblique landscape images, since it also corresponds, for example, to historical image databases or real-time orientation for markerless augmented reality. Two types of processing exist for this type of images:

- First, SfM-based methods are not directly compatible with the georeferencing of a single image and therefore to, the problematic of growing databases and supervised georeferencing. SfM is challenged if the images do not overlap sufficiently and represent varying appearance of the landscape and various viewpoint. Hence, images are left out.
- Second, registration of a single image with a landscape image has been studied mainly for 3DOF

pose estimation by Baboud et al. (2011); Chippendale et al. (2008). Their case studies are located in areas where they would gain from the existence of overlap between the images. Baatz et al. (2012a) provide a large-scale localization from the skyline which is promising, but not accurate and would gain from a final refinement.

The advantages of our method are then:

- The registration is performed in a geographic coordinate system, suitable for growing databases.
- This method is directly compatible with user interaction which can be necessary for the georeferencing of the reference images and the most dissimilar query images.
- It is also designed to be compatible with other methods (skyline matching and HOG matching) presented in this thesis. Hence, it is applicable to a wide range of images.

In our proposition, reference images are GPS images oriented with a DEM. They are used to recover automatically the full pose (orientation and location) of other images belonging to the same collection and having an approximate location. To reach this goal, we proposed an original workflow, called PEP-ALP, based on a Kalman filter and use the 3D landscape model to add more robustness to the SIFT matching. To the best of our knowledge, this method is the first to recover orientation and location of tourist images in rural areas. Since the user is involved only at the beginning of the process, i.e. during the orientation of the GPS located images, it remains reasonably close to an automatic routine.

The achieved accuracy is not comparable to the one of orthoimages generated via a classic acquisition and processing of photogrammetric images, which remains a limitation for the usage of our workflow for environmental studies. To improve the accuracy further, an increased involvement of the user would be necessary. Nonetheless, the pose is correct enough to open interesting opportunities for images database management, for example for advanced querying or augmented reality purposes. Moreover, it can be used for a wider variety of images collections and with a limited user involvement.

With the current workflow, 50% of the pictures are georeferenced. The goal is to provide a solution for every picture with a limited user supervision. Hence, further developments should provide a method for the detection of the initial azimuth which is independent of SIFT: (i) by matching the pictures with the landscape model (HOG matching), (ii) simply by asking the user to provide the initial azimuth. Moreover, the process is also designed to be extensible to images without geotags if an estimate of their location is obtained, for instance by measuring their similarity with already georeferenced images. Finally, if a cluster of images is well connected, we could further process the pictures with a bundle adjustment to gain accuracy.

5.4 Discussion of the proposed methods

In this Chapter, we presented three methods directly related to the goal of this thesis, that is the registration of a picture with a landscape model. In Section 5.1, we discussed the registration with the skyline and in Section 5.2 the registration with a 3D synthetic images. In Section 5.3, we also inserted in the landscape model georeferenced pictures, more similar to the query images than synthetic images. In the next Section (Section 5.4.1), we will compare the two first methods (which were tested with a similar database of pictures) and discuss their possible interaction. Next, in our experiments, we assumed that the focal length of the camera was known. In Section 5.4.2, we will discuss the impact of this hypothesis. The database of pictures acquired for these experiments is the focus of the next discussion (Section 5.4.3). In these contributions, we exploited as reference an accurate DEM and an accurate orthoimage. In Section 5.4.4, we will discuss how we could improve our experiments by considering other DEM and orthoimages and assess their impact on the georeferencing. Finally, other reference geographic data could be exploited as well: in Section 5.4.5, we will then comment the potential of other GIS data for the registration of landscape images.

5.4.1 Skyline and HOG registration, possible interaction

For the task of registering an image with a 360° panorama generated from the picture locations, the two methods proposed have comparable performances (82% DTW matching, 78% HOG matching). The DTW matching is semi-supervised whereas the HOG matching is entirely automatic. Some images of our database are presented in Figure 5.29. Images successfully oriented with the skyline but not with HOG are presented in the upper left part. Image successfully oriented with HOG but not with skyline are shown in the upper right box. Finally, images that are never correctly oriented are reproduced on the bottom. Dashed ellipses highlight regions which are not represented in the synthetic reference images because they are beyond the boundaries of the DEM.

Considering the settings of our experiments, it seems natural to expect the skyline matching to be more efficient than the visual matching. Indeed, the skyline detection is supervised to ensure that the skyline is correctly identified. Moreover, the skyline is neither perturbed by illumination nor by rendering distortions. Finally, our case study region is located in the Alps, where the morphology is ideal for skyline matching. However, we can expect the visual matching to be more appropriate than skyline matching in two situations. i) When the skyline is occluded by buildings or vegetation. In this situation, the skyline (considered as a whole in our method) cannot be correctly aligned with the synthetic skyline. In contrast, the matching with local patches can discard the occluded area of the image. ii) When the skyline is not accurately detected or too smooth to provide good tie points. In this situation, the consideration of the land cover can generate more discriminative correspondences.

One of the motivations to use local visual features is to be more robust to occlusion. The images C3, C4, D3, D4 in Figure 5.29 seem to confirm this behavior, occlusion of the skyline perturb the DTW matching but do not impact the matching of visual patches. The images C1, D1 and D3 have smooth skylines which explain why skyline matching fails. Apparently, for these images, the consideration of the land cover improves the registration.

In the second set of images, only oriented with skyline, the images A1, A2, B1, B2 capture large regions close to the camera where no HOG correspondences can be detected. The distortions engendered by the rendering of the synthetic view and the lack of discriminative patches may explain why HOG matching fails. The images A3 and B3 have backlight and the image A4 is crossed by a cable, situations which seem to perturb the HOG matching.

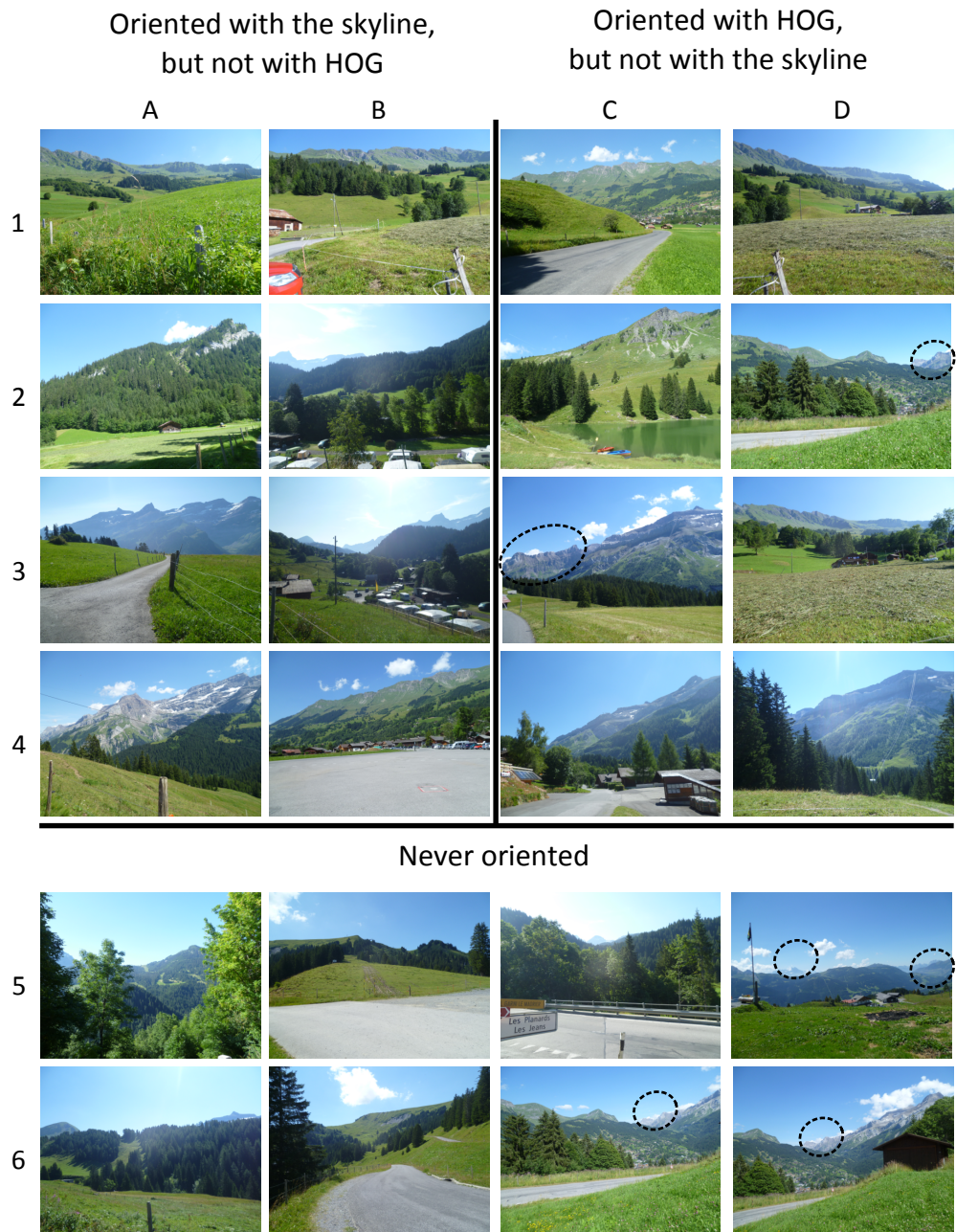


Figure 5.29 – Comparison of the matching of skyline and visual features. Ellipses show regions which are not reproduced in the reference synthetic images.

Chapter 5. Registration of a landscape image with a landscape model

To summarize, skylines and visual features have strengths and weaknesses. They are listed hereunder:

Skyline matching:

- ✓ The skyline geometry is neither impacted by illumination and seasonal variations nor by the image content (distortions in the foreground).
- ✓ Skyline description is compact.
- ✗✓ The skyline matching requires a skyline detector which is not trivial to implement, specifically to work with various type of images. In supervised conditions, the detection of the skyline with standard segmentation algorithm is simple and fast.
- ✗ In our implementation, the skyline is considered as a whole, it is then impacted by occlusion.
- ✗ If the skyline is too smooth, it is difficult to find good tie points.

HOG matching:

- ✓ The image processing to describe an image patch with HOG is simple.
- ✓ The matching is entirely automatic.
- ✓ Local features avoid the problem of occluded regions.
- ✗ HOG description is perturbed by significant illumination variations (at least backlight). We tested our method on images shot only in the summertime, we cannot discuss the robustness with respect to seasonal variation.

Considering our implementations, and since the performance are equivalent, the HOG matching will be preferred if the user cannot be involved or if the skyline is occluded. After an initial alignment with HOG, the pose provides an indication of the skyline position and shape. This prior will facilitate the automatic and accurate detection of the skyline in the query image. Hence, the skyline matching could be used to refine the pose. However, if a method to detect automatically the skyline is developed, in general, the skyline matching would be more appropriate than HOG, except in morphologically smooth regions (hilly rather than mountainous).

5.4.2 Focal length

In our experiments of the matching of a picture with a landscape model, we always assumed that the focal length was known. We choose this strategy to focus on the image matching rather than the geometry. This is a limitation to extend these methods to real collections of images which usually are recorded with various cameras. Indeed, for recent pictures the focal length can be extracted from the metadata, however for many collections (historical), it is uneasy to recover the exact focal length before the pose estimation. Hence, it would be desirable to extend our experiments and methods for unknown or approximate focal.

The skyline matching with DTW is by definition extensible to unknown focal. Indeed, the elasticity of the DTW can recover distortions such as induced by a scaling.

In our second experiment, we matched an image with a 360 panorama. If the scale is unknown, the

matching should also be implemented in the dimensions of the scales. The modifications that have to be introduced in the algorithm are small but there is a risk that more false positive will be detected. Finally, the relaxation of the focal within PEP-ALP is also straightforward and is only matter of relaxing one parameter in the Kalman filter. In order to avoid a divergence of the pose estimation, it is important that accurate correspondences, well distributed in the query image are extracted. Thus, within PEP-ALP the matching of the 3D key-points of the reference images should be mixed with correspondences detected from the synthetic image and hence gain correspondences where there is no overlap.

5.4.3 Datasets and benchmark datasets

Compared to pose estimation of images in an urban environment, few studies focus on landscape images. Hence, there is no benchmark dataset appropriate for our methods. Consequently, we acquired our own set, which is representative of various ratios of occlusion and foreground but contains only illumination variations occurring during summer days. Therefore, this dataset cannot be used to evaluate our methods for more challenging acquisitions such as various weather conditions, seasons, illuminations or drastically dissimilar images (historical, paints and drawings).

An ideal dataset would also associate the images with their ground truth location and orientation. These values should be sufficiently accurate to be considered as the reference and provide a fair comparison of the method performances. One should specifically ensure that the measurement of the camera angles is sufficiently accurate. In this thesis, we consider the GPS location as the Ground Truth, but the camera angles provided were insufficiently accurate and we had to generate manually ground truth azimuth and tilt.

Finally, the reference geographic data (DEM and orthoimages) should also be delivered to limit the time spent on the acquisition and processing of the reference data and also put every competitive research on a level playing field.

The creation of this benchmark would be very valuable for the comparison of the performances comparisons. Baatz et al. (2012a) provides a dataset which we did not use for a technical reason. Indeed, it would have required the download of orthoimages in multiple regions of Switzerland, a task which has not been automatized yet in our process. Moreover, the comparison of our method with the one that they propose is pointless since they do not share a similar goal (global vs local georeferencing). In contrast, the comparison with the studies of Chippendale et al. (2008) and Baboud et al. (2011) would have provided us with a more reliable evaluation. Hence, a limit of the evaluation of our method (and other authors in this field) is a reliable and comparable evaluation of the performance.

5.4.4 Reference data (DEM and orthoimages)

Along our experiments, we used as reference geographic data provided by the Swiss mapping agency which are known to be very accurate. They are the references for the height and appearance and may have an impact on the accuracy of the georeferencing.

DEM: Our reference 3D model has a 25m resolution. For a further research, we could assess the impact of the resolution on the georeferencing accuracy. On the one hand, there is the possibility to use a Digital Surface Model (DSM) that contains also the building and vegetation heights and have a higher resolution. It could especially have an impact on the georeferencing with the skyline in regions where a forest covers the mountains. Considering terrestrial pictures, the drawback of DSM is that the vegetation and buildings may occlude the background where the correspondences are detected. On

the other hand, global DEM, having a resolution of 30m to 90m, are available worldwide. They will generate distorted skylines and landscapes but would provide an assessment of the georeferencing accuracy in the worst scenario.

Orthoimage: To render our landscape model, we used an accurate and recent orthoimage acquired during summer campaigns. Hence, the orthoimage is neither acquired in the same year (landcover variations) nor at the same moment of the day (shadows and illumination variations) as our query images. This is why we detect correspondences with the HOG descriptor used for cross-domain matching. In our experiments, we did not assess the impact of the reference orthoimages. Especially, recent aerial sensors provide seasonal orthoimages and oblique views, which could improve the georeferencing. Hence, it would be interesting to complete our experiments using various resolutions of orthoimage recording various seasons. For the georeferencing of historical images, we could also consider rendering the DEM with historical orthoimages and hence reduce the temporal shift.

5.4.5 Alternative usage of landscape models as the reference

Currently, in the field of matching landscape images with geographic data, only DEM and derived skyline and morphological break-lines (depth discontinuities) have been used as reference. We have added to that a DEM textured with an orthoimage (and indirectly also a DEM textured with terrestrial images).

For instance, Ardeshir et al. (2014) detect easily recognizable objects stored in GIS databases, such as road signs, to locate a camera at a city scale. Alternatively, Baatz et al. (2012b) propose a 3DOF orientation based on land cover classes. These methods generally follow these steps:

1. Training images are acquired with ground truth classification.
2. A classifier is trained.
3. The query image is classified.
4. The classification result is compared and matched with the GIS database.

In the context of landscape images, this workflow has two challenges.

- **The training set** must be representative of the query images. We discussed the lack of benchmark dataset and it is difficult to generate a set representative of a large variety of images. If the images are not oriented, they must be annotated with the various classes or object present in the reference database.
- **The classification** (or recognition) should not be underestimated. The recognition of high-level features is still an active research area. Currently, the recognition of high-level objects addresses mainly the recognition of everyday objects (in computer vision) and the recognition of land cover classes in multi-spectral aerial images (in remote sensing).

Hence, without considering the camera orientation, the classification or recognition of objects in high oblique RGB terrestrial images of rural areas is already a challenge.

Hence, Hammoud et al. (2013); Koperski et al. (2013) propose a supervised recognition of particular geographic objects which are subsequently searched within a geographic database. Applied to

landscape images, we could imagine a similar process: the user clicks on visible summits, encloses forests, villages and lakes and delineates skyline, roads and rivers etc. If he has further knowledge of the area, he can also digitize semantic information, such as toponyms. The spatial distribution of the geographic objects could then be compared with a geographic database to identify regions having similar properties.

6 Learning prior:

Measuring attractiveness of a location

For the case studies presented in Produit et al. (2012, 2014b), we downloaded databases of shared pictures with their location and visualized them in a map. By screening these locations, we noticed some trends. The most obvious one is that the pictures are generally shot from a road or a path. This type of information could also be of interest for pose estimation of images. Specifically, in the previous chapter, we draw the conclusion that for the 6DOF orientation of an image having a loose prior (such as a region indicated by a toponym), an appropriate method could be to test several locations and estimate the one giving the most similar skyline or synthetic view. In this scenario, a measure of the attractiveness of each location could be used to discard unlikely locations and to promote the most attractive ones.

In this chapter, we developed a method to produce the map of the attractiveness. This research was partially presented in Produit et al. (2014a).

6.1 Introduction

In the previous chapters, the focus was on the pose estimation of a picture having a rough location and azimuth. These approaches suppose implicitly that the 3D space of possible locations is homogeneous. This hypothesis can be questioned. For instance, no landscape pictures can be shot from locations under the ground. Beyond this obvious consideration, we can also start considering location priors induced by the land cover if the images are shot from the ground level (most of the high oblique images that we want to georeference in this thesis).

In the previous chapters, we showed that 6DOF camera orientation is challenging because of the lack of efficient features detectors and descriptors to match a synthetic image and a real one. However, we also showed that one can measure the similarity between them, for instance with the skyline (Section 5.1) or a HOG description (Section 5.2). Hence, as a coarse solution, synthetic images (or panoramas) can be generated for a regular grid of locations. Then, their similarity with the picture can be measured to determine the most similar location. In this scheme, the matching could be improved by down-weighting or discarding locations according to their probability to be a shooting location. The measure of this probability is the objective of this chapter.

Schlieder and Matyas (2009) state that a picture location is the result of two spatial choices. The choice of the subject of the picture and the choice of the location. We can say that the choice of the location is driven by two factors: the accessibility and the field of view. The first factor is a physical

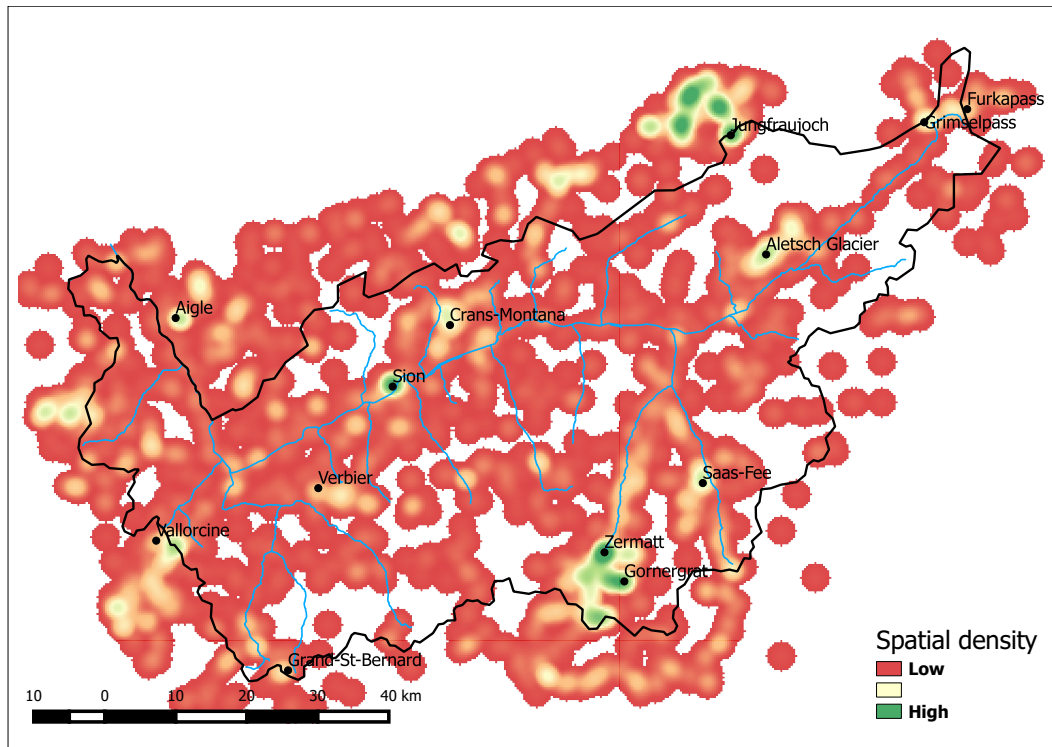


Figure 6.1 – Spatial density of GPS picture locations for our case study (Valais). Hot spots (clusters of high density) are pointed out in green. Low densities are red and transparent, if null. It is no surprise that, at the scale of the area, the spatial density highlights attractive regions for tourists: ski resorts (Crans-Montana and Zermatt), skiing areas (Gornergrat and Verbier), remarkable scenery (Aletsch glacier and Jungfrauoch), passes (Grand-St-Bernard and Furkapass), and cities (Sion). Note that such maps can be perturbed by the presence of very active photographers who upload more images than others, or by outliers (generating the circular patterns in the right part of the map). Finally, with the democratization of smartphones and GPS-enabled cameras, this map would probably be more complete today (data represented on the map is the state of the database in summer 2013).

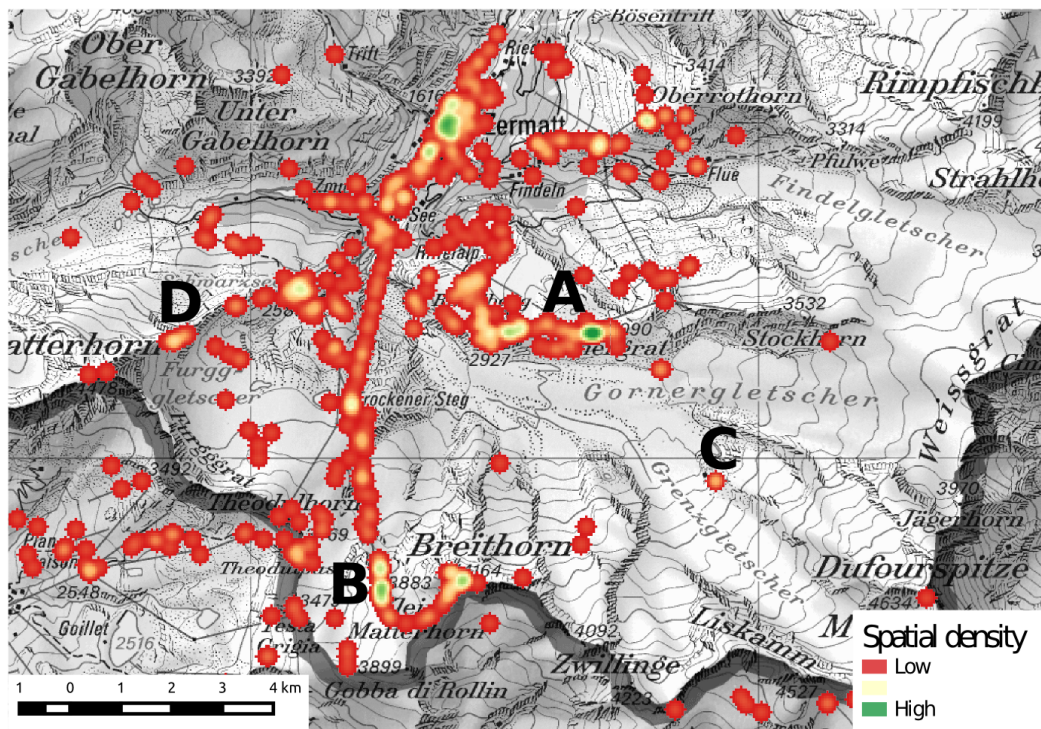


Figure 6.2 – Spatial density with a smaller bandwidth to study the region around Zermatt. Some points of interest are clearly visible in this density map. For instance, the cable car stations of the Gornergrat (A) and Klein Matterhorn (B). Oppositely, isolated locations induce small peaks of density which can be correlated with a point of interest (Matterhorn summit) but are generally not. The main drawback of this probability field is that there is no information in areas where no photo has been shot. For instance, we would expect a higher probability on the paths which leads to the Monte-Rosa (C) or Hörnli (D) huts. In contrast, the cliff which is on the south of the Gornergrat, has a high density even if it is not accessible.

The problem of the outliers is clear on this map: isolated pictures generate small density peaks, which do not necessarily reflect particular places. Hence, the density is trustful only above a certain density value. As discussed earlier, one can generate clusters of images by fixing a density threshold. These clusters can be used to enclose images associated with a same point of interest or expected to share some visual and textual properties. (*Swisstopo*)

Chapter 6. Learning prior: Measuring attractiveness of a location

constraint. That is to say, easily accessible locations have more chance to be shooting spots. The second factor is more artistic: locations open on a large landscape, or from which a point of interest is visible, are preferred. In this chapter, we will formulate some assumptions about the relations between the picture locations and a set of geographic indicators describing the accessibility and visibility of a location. We will validate our hypothesis with a database of pictures locations downloaded from the web (a subset of the data used in Wider et al. (2013)). Moreover, we will use the learned relations to measure the probability of finding a picture in every location of a map.

Many GIS scientists have taken the opportunities offered by the web-shared databases to study the spatial distribution of picture locations. For instance, they show that pictures are clustered around points of interest (Schlieder and Matyas, 2009). Using these statistics one can say that locations close to points of interest are more probable than others. Such probability can be approximated by a spatial density of pictures. At global scale, such as illustrated in Figure 6.1, the approximation of the picture location probability with a spatial density is perfectly valid. However, by zooming in the map, (see Figure 6.2) and thus reaching the scale of the pose estimation, the spatial density has undesirable behaviors. First, some images are misregistered or isolated and are not representative of a general trend. Second, the density does not take into account the land cover and morphology of the area, which can have a strong influence on the picture locations: some locations have very low probabilities (steep slopes and cliffs, heart of a forest, highways) while others are more likely to be shooting locations (paths, arrival of cable cars, summits). Finally, a location which is not in the vicinity of other pictures is associated with a probability equal to zero, but we cannot know if it is because of a real lack of attractiveness or of a lack of images in the database.

In this chapter, we will show how we can infer the probability of a map location to be a picture location, by taking into account their nature and morphology. To this end, we will model this problem as a Geographic One-Class problem which aims at separating places similar to picture locations from the other locations. Specifically, we will apply a Kernel Density Estimation in the space of geographic features to overcome the problem of the methods based on the spatial densities (computed in the space of geographic coordinates).

6.2 Review

Databases of web shared and geotagged data open great opportunities for researchers (Luo et al., 2011). A point in common of the works based on images is the need to detect spatial clusters of images. These clusters group together images sharing visual or semantic similarities. Scientists usually make use of a threshold of the spatial density to enclose a group of images constituting a spatial cluster (Li and Goodchild, 2012). Clusters are computed and detected at various scales, depending on the goal of the study. An example of a spatial density is presented in Figure 6.1 and Figure 6.2.

In the computer vision community, the availability of images with corresponding geotags is an opportunity to infer the location of pictures without geotags. Crandall et al. (2009) extracted clusters of high density with mean-shift clustering. Next, they associate a query image with a cluster (and thus a location) by measuring the similarity their textual and visual features. The main limitation of this approach is that an estimate of the location can only be computed for the pictures belonging to clusters, which are a small portion of the entire space. Thus, Tsung-Yi et al. (2013) propose to match the pictures directly with an orthoimage and a land cover map using visual features. In the GIS community, authors generally do not use the images content, but only their locations and tags. For instance, Li and Goodchild (2012) used density of pictures to draw the extent of particular locality and name them.

The density is used to extract the contour of spatial clusters. The tags frequencies within a cluster are analyzed to set the name of the locality. This work is similar (except the scale) to the one of Grothe and Schaab (2009) who delineate the extent of larger and fuzzy areas like the Alps, using Support Vector Machine and Kernel Density Estimation. In urban studies, collaborative databases have been widely used to analyze people behavior and general trends in the tourists flows (Sun et al., 2013; Popescu and Grefenstette, 2009; Girardin et al., 2008).

Generally, these studies use the pictures locations to detect particular spatial patterns (such as flows or highly attractive areas) and to delineate them geographically. Hence, they implicitly explain the clusters locations by the presence of one or several particular landmarks, which attract the photographers. For pose estimation, the spatial density can be used to get a first rough location of a picture, similarly to Crandall et al. (2009), who assign a picture to a cluster by measuring their textual and visual similarity. However, this type of priors shows some drawbacks.

- First, the cluster delineation is trustful at a global-scale but a cluster is too large for precise pose estimation. On the contrary, a spatial density estimation at local scale suffers from the presence of outliers (misregistered geotags) and loses its generalization power, since it does not reflect general trends but the sampling (see Figures 6.1 and 6.2). Particularly, using spatial density of images, one implicitly accepts that every image is shot from attractive spots or in the neighborhood of recorded shooting locations. The probability outside these areas, where no photographs has been yet shared or shot, is null.
- Second, the probabilities within a cluster depend only on the proximity of other pictures and not on the land cover which may intercede positively or negatively for a location.

Hence, our main contribution in this study is to compute a probability at a local-scale, for every map location, and not only in the neighborhood of attractive spots. Our goal is then quite similar to Tsung-Yi et al. (2013): both approaches use geographic data to provide an estimate which is not biased by the spatial density of pictures. However, the focuses are different: they estimate the location of an image content, whereas we want to provide an estimate of the probability that a location is a good spot to shoot a picture. To do so, we compute the density of image locations in the space of geographic features rather than in the 2D space generated by the geographic coordinates. By doing so, we also learn which combination of geographic features is appealing for the photographer and which ones are repulsive.

6.3 Problem formulation

Our hypothesis in this study is that the locations of pictures in rural areas are related to geographic indicators. These indicators describe the accessibility of the locations (proximity to a path or a road, proximity to a public transportation station) or their inaccessibility (cliffs, steep slopes, deep valley). Second, they can also describe the field of view (location on ridges or summits have larger field of view) or occlusion (forests). The combination of these indicators creates a good or bad location to take a picture. In this study, we will not consider the presence of point of interest as a driving factor nor pictures shot in urban areas (since they are supposed to be linked to other factory and thus other geographic features) and focus on the drivers of attractiveness of rural pictures only.

First, the entire map is discretized with a grid of locations Y . Each location is described by some geographic features z . We want to separate these locations in two sets, the set of positive locations ($Y = 1$), which are likely to be picture locations and the set of negative locations ($Y = 0$) which are

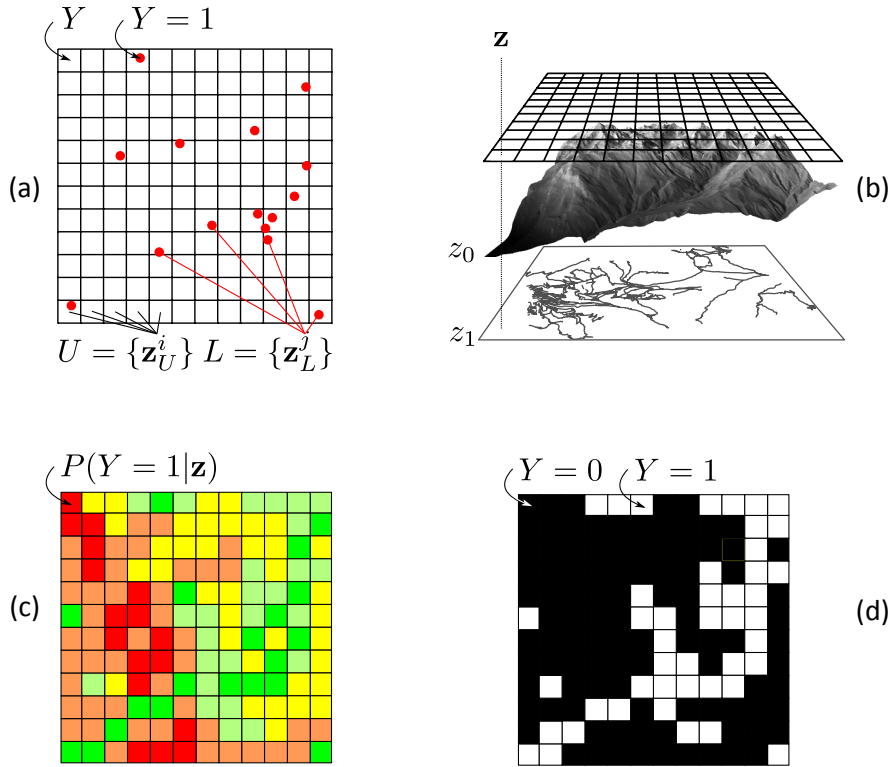


Figure 6.3 – General method: (a) The map is discretized, the cells constitute the unlabelled set U . The picture locations are stored in the labelled set L . (b) The geographic indicators of each cell and each picture location are computed. (c) The probability (or attractiveness) map is generated. (d) A threshold of the probability value separates picture locations from the rest.

unlikely to be picture locations. Second, we have a set of pictures downloaded from Flickr (flickr.com 🌐). Only the images localized by GPS are selected, to ensure that their position is sufficiently accurate. These picture locations compose the set of positives ($Y = 1$), we will also call it the labelled set hereafter. From this set, one can derive directly $P(z|Y = 1)$, that is the probability of a geographic description given that the location provide pictures. However, our goal is rather to measure the probability of any location (unlabelled locations) to be a picture location, given its geographic description $P(Y = 1|z)$. This value can be thresholded to generate a binary classifier. The general method is illustrated in Figure 6.3.

Hence, in the following, we will first discuss how this type of problem is treated in the literature and identify the method which suits the most our problematic. Second, we describe this method and demonstrate how it can be simplified to match our needs. Third, we will show how the distribution functions of the data are estimated. Fourth, we discuss the processing of the geographic features and finally, we will explain how the performance is evaluated.

6.3.1 One-Class Data and Geographic One-Class Data

This setting is specific to One-Class problems (OC). A One-Class problem is a classification task in which only the set of positive labelled data is available to train the classifier. For each incoming unlabelled

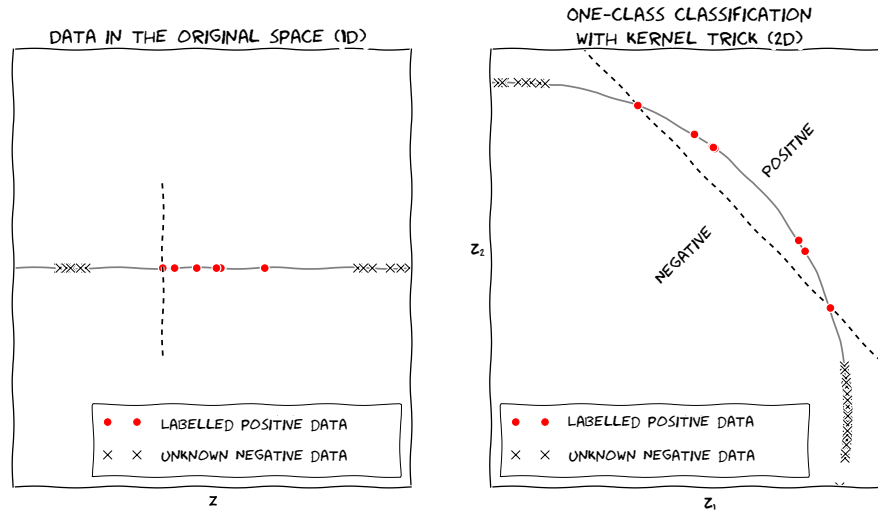


Figure 6.4 – In the original space (1D), positive data are hardly separated from negative data with a linear classifier. In the higher dimensional space (2D), the data are projected on a sphere and a linear classifier can successfully separate the positive from the rest. The distance to the hyper-plane is an estimate of the probability.

data, we want to classify it as positive (similar to the training data) or negative (dissimilar to the training data). In such problems, no (or few unrepresentative) negative data are available to train a binary classifier. During the last years, kernel methods have been widely used for OC problems (Schölkopf et al., 1999; Tax and Duin, 2004; Hoffmann, 2007). In this study, we will present the results obtained with the One-class SVM (OCSVM, Schölkopf et al. (1999)) and the Kernel PCA (KPCA) for novelty detection (KPCA-nd, Hoffmann (2007)). Both methods use the kernel trick to project the original data on a higher dimensional hypersphere. In this high dimensional space, the data are more likely separated by a linear model (Schölkopf and Smola, 2002).

In geography, the related data are called Geographic One-Class Data (GOCD, Guo et al. (2011)). Usually, they refer to data representing the presence of a phenomenon observed in several locations (a picture location). However, for every other location of the map, the absence of the phenomenon can either be explained by an inappropriate location (inappropriate location to shoot a picture) or simply because this location is not sampled (appropriate location, but no picture is recorded). Compared to OC problems, GOCD problems are slightly different in the sense that in OC the unlabelled data to be classified are not known at the training time. On the contrary, in GOCD, every unlabelled location and related features are available beforehand. Guo et al. (2011) compares OCSVM (Schölkopf et al., 1999), Maximum Entropy (MAXTENT) (Jaynes, 1957) and Positive and Unlabelled Learning (PUL) (Elkan and Noto, 2008) for GOCD problems. The two last methods provided the most accurate results. In our study, we will consider the PUL approach, which can be simplified to match our problem. The advantage of the GOCD approaches, such as PUL, compared to traditional One-Class method, is that they take advantage of the existing unlabelled data to improve the classifier.

The two approaches are illustrated in Figures 6.4 and 6.5. In Figure 6.4, OCSVM and KPCA-nd project the data in a space of higher dimension where a hyperplane (dashed line) separates positive and negative data (because a linear classifier cannot classify correctly the data in the original space). The unlabelled data are not available for the choice of the hyperplane location. It is then chosen to fit the labelled data. The distance of a point to the hyperplane gives the measure of the probability to be

Chapter 6. Learning prior: Measuring attractiveness of a location

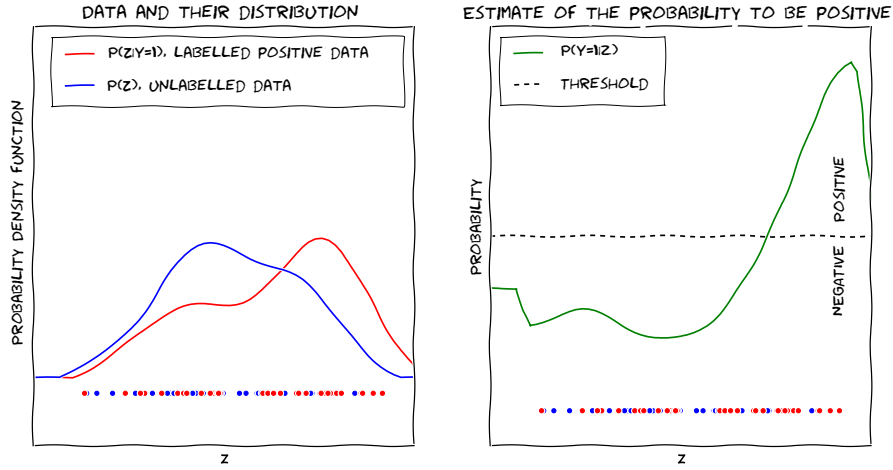


Figure 6.5 – In PUL approach, the difference between the distribution of the unlabelled data and the labelled data is used to estimate the probability. A threshold is fixed to separate the positive from the rest.

positive or negative. In comparison, in Figure 6.5, PUL compares the distributions of the labelled data and the unlabelled data to measure the probability of each unlabelled data to be positive. This value can be thresholded (dashed line) to classify the unlabelled data.

6.4 Method

The detailed method is illustrated in Figure 6.6, the method has seven steps:

- A. Every location is described with geographic indicators.
- B. The indicators are preprocessed (normalized) and optionally decorrelated (with a PCA or a KPCA).
- C. Four sets are formed, the set containing the unlabelled locations is U and is regularly sampled in the region. The set containing the picture locations constitutes the labelled set L . This set is subdivided in L_T to train the method, L_t to test and choose the free parameters of the method and L_V which is used only to validate the method.
- D. From these sets, we will derive and compare the distribution of all locations ($P(z)$) and the picture location ($P(z|Y = 1)$). To do so, we will estimate the distribution from the data with kernel density estimation (KDE).
- E. The probability of each location to be a picture location given its geographic features ($P(Y = 1|z)$) is computed from these two distributions. This value is mapped and also thresholded by a threshold λ to classify each location as positive or negative.
- F. We will formulate a cost function whose goal is to measure how a set of parameters fits the testing set L_t . The cost function is based on the recall of the testing set, r_{L_t} (the ratio of correctly classified locations) and the ratio of unlabelled locations classified as positive, r_U . This cost indicates the best set of parameters.
- G. Finally, the independent set is classified and is used to validate the method.

Thus, the next sections will discuss each step in the order proposed in the Figure 6.6.

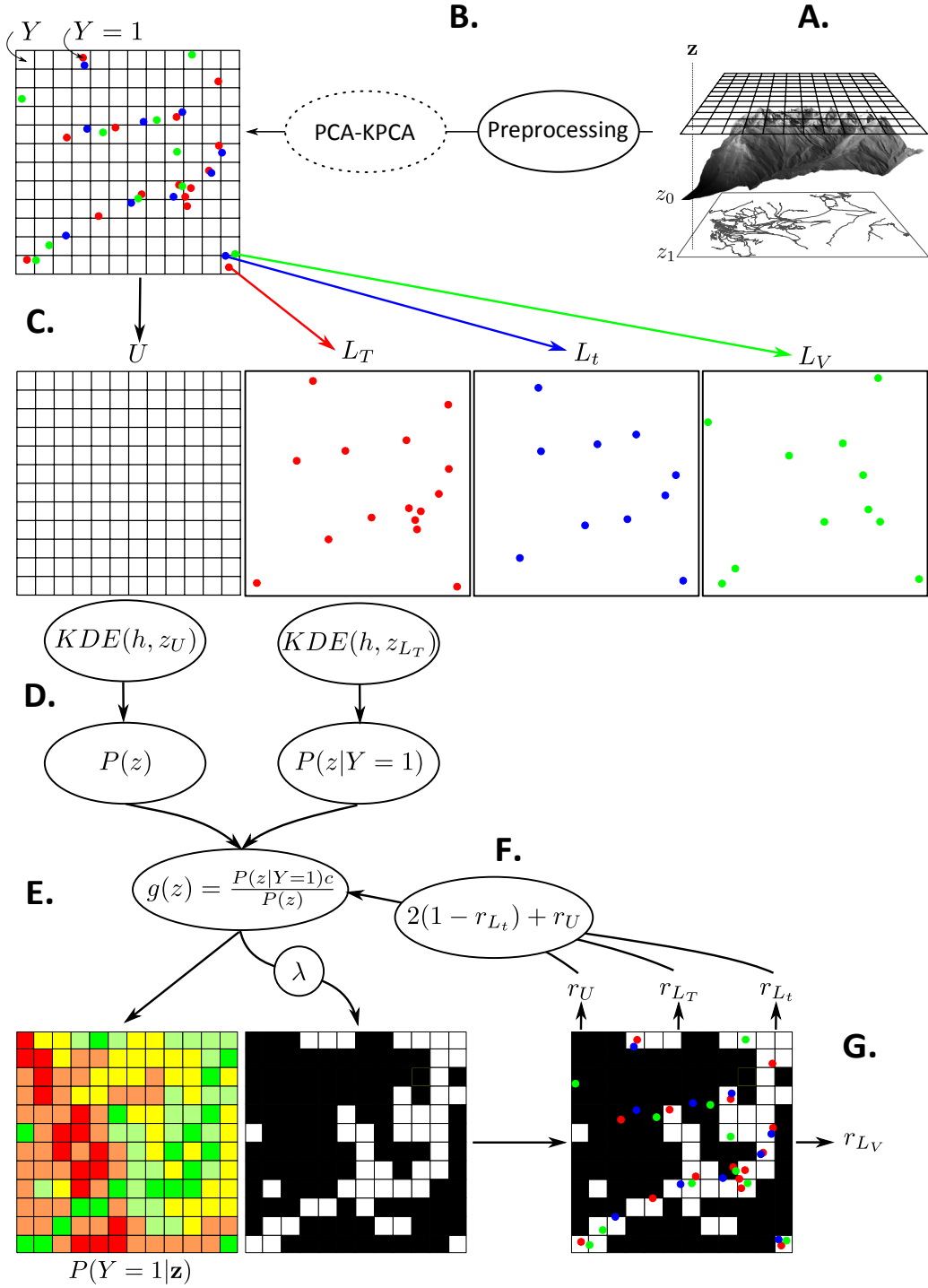


Figure 6.6 – Seven steps of our method. A. Geographic indicators computation. B. Geographic indicators preprocessing and decorrelation. C. Separation of the sets. D. Estimation of the distributions. E. Estimation of the probability and of the threshold. F. Cost function to select the best set of parameters. G. Validation

6.4.1 Geographic indicators computation

A regular grid and the pictures locations are overlaid on the region. For each location, the corresponding set of geographic indicators is extracted. The processing is done in SAGA-GIS, chosen for its efficiency. The geographic indicators are detailed in Table 6.1, 124.

6.4.2 Geographic indicators preprocessing and decorrelation

First, the geographic indicators are normalized. Next, we propose two strategies to extract the significant features and their weight. A classic approach is to perform a Principal Component Analysis (PCA). The PCA detects the axes (or features) which explain the most of the variability of the data. These axes are also decorrelated. We will also test the non-linear equivalent of the PCA, the KPCA (Schölkopf et al., 1998) to suppress non-linear relations between the geographic features. The extraction of geographic features with PCA and KPCA are optional, the process can also be applied directly to the normalized geographic indicators.

6.4.3 Labelled data sets separation

We separate the labelled set L in three subsets of roughly the same size. To do so and to avoid spatial correlation, we overlay a bigger grid on the region, each cell (and the belonging picture locations) is randomly attributed to the training set L_T , the testing set L_t or the validation set L_V . The three sets are mapped in Figure 6.13, 135.

6.4.4 Distribution estimation with kernel density estimation

To estimate the probability density function of the locations with respect to the geographic features z , we will apply a Kernel Density Estimation (KDE, also named Parzen window, Parzen (1962)).

KDE is a non-parametric method for the estimation of the Probability Density Function (PDF) of a dataset. KDE estimates the density of data by applying a local smoothing filter. It has some desirable features. First, it limits the impact of outliers. Second, it does not require prior information about the distribution of the data, such as a probability distribution.

The KDE function in Equation 6.1 is used to evaluate the density at a location z^j given a data set $X = \{z_X^0 \dots z_X^i \dots z_X^n\}$. This data set is composed of n pictures described with the geographic feature z_X^i . In our problem, we will evaluate the density of $L_T = \{z_{L_T}^i\}$ and $U = \{z_U^i\}$.

$$\hat{f}(z^j, z_X) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{z^j - z_X^i}{h}\right) \quad (6.1)$$

where $K(x)$ is a local smoothing operator, or kernel function, and h is the bandwidth of the kernel function. Among the different kernel functions, we used the Gaussian function:

$$K(x) = (2\pi)^{-d/2} \exp\left(-\frac{1}{2}x^2\right) \quad (6.2)$$

Scott's rule, applied to the training set dimension, is used to compute the bandwidth h (Scott and David, 1979):

$$h = n^{\frac{-1}{d+4}} \sigma_{z_X} \quad (6.3)$$

where σ_{z_X} is the covariance of the positive dataset. This rule uses the dimension d of the geographic feature and the number of data n to estimate a reasonable h . The choice of the appropriate kernel function has less influence on the results than the choice of the proper bandwidth. Indeed, if the bandwidth is too small, the density is over-fitted to the positive data set and sensitive to the outliers: the generalization power becomes weak. On the contrary, if the bandwidth is too large, the density is oversmoothed and the local peaks disappear. In our method, KDE is applied to the labelled data L_T to estimate $P(z|Y = 1)$ and to the unlabelled data set U to estimate $P(z)$.

6.4.5 Probability estimation and classification

In the PUL model, the labelling process, s (the sampling of the data), is also modelled. If $s = 1$ the location is labelled, if $s = 0$ the location is not labelled. Hence, we can assume that the probability to be a positive location if the location is labelled is 1. Since the sampling is usually performed by an expert this assumption is generally considered as valid.

$$P(Y = 1|z, s = 1) = 1 \quad (6.4)$$

Similarly, the probability to be a negative location, if the location is labelled is 0. Indeed, the sampling considers only positive locations.

$$P(Y = 0|z, s = 1) = 0 \quad (6.5)$$

Finally, the probability to be labelled if the location is positive is independent of z and is a constant c . That is to say, every location on the map has the same probability to be sampled.

$$P(s = 1|z, Y = 1) = P(s = 1, Y = 1) = c \quad (6.6)$$

This model is well adapted to ideal cases. However, as stated by Dudík et al. (2005), the sampling is spatially biased by the accessibility of the location. In practice, it is difficult to guarantee that the labelling is independent of the geographic features or that the labelled set contains only positive examples.

Going back to our problem, the bias engendered by the accessibility is exactly what we want to measure. We can stop considering s and assume that $Y = 1$ reflects exactly the labelling process, which we want to be dependent of the geographic features. Hence, using the Bayes rule, we will simplify the problem as:

Chapter 6. Learning prior: Measuring attractiveness of a location

$$P(Y = 1|z) = \frac{P(z|Y = 1)P(Y = 1)}{P(z)} \quad (6.7)$$

Where $P(z|Y = 1)$ is estimated from the set of labelled data L_T and $P(z)$ from the unlabelled data U . $P(Y = 1)$ is an unknown constant a . However, we are not interested in the exact value of the probability and a value proportional to the probability for the classification and the mapping is sufficient for our purpose:

$$g(z) = P(Y = 1|z) = a \frac{P(z|Y = 1)}{P(z)} \quad (6.8)$$

$g(z)$ is the value measuring the attractiveness that we compute and map for each location z . Finally, for the classification, we will set a threshold on $g(z)$ to delineate the border between the locations similar to picture locations ($Y = 1$) and the locations dissimilar to picture locations ($Y = 0$).

$$h(z) = \begin{cases} 0 & \text{if } g(z) < \lambda \\ 1 & \text{if } g(z) > \lambda \end{cases} \quad (6.9)$$

6.4.6 Performance evaluation and setup

Several parameters need to be fitted by this method. First, we will have to choose a set of geographic indicators. Second, they will be decorrelated with a PCA for which a number of principal components to retain n_{pca} has to be fixed. Finally, the probability is estimated and we have to fix the threshold λ for the classification. Thus, there are many possible settings and we have to measure their performance.

A way to select these parameters is the division of the training set in two subsets. The first is used for the training (L_T) and the second is used to verify the appropriateness of the classification result (L_t). In classification problems, the parameters are chosen to maximize the classification score (the ratio of testing data correctly classified). However, in OC problem, one can reach high classification score with loose boundaries. In Figure 6.5, it will correspond to pull the boundary at the bottom: the entire test set is correctly classified, but many negative data are classified as positive as well. Hence, in Guo et al. (2011) the positive and unlabelled score F_{pu} is maximized:

$$F_{pu} = r_{L_t}^2 / r_U \quad (6.10)$$

Where r_{L_t} is the recall (the proportion of correctly predicted data in the testing set) and r_U is the ratio of positive predicted locations in the unlabelled set (U). One want to have a high r_{L_t} but also a small ratio of unlabelled data classified as positive. However, in our problem, it appears that this criterion privileges solutions which have a very small r_U and a medium recall (see Figure 6.10).

In this study, we will minimize a simple and intelligible criterion which is:

$$C = 2(1 - r_{L_t}) + r_U \quad (6.11)$$

The goal of the criterion is first to avoid the overfitting on the training set. Hence, we involve the testing set in the criterion to ensure that the chosen parameters perform similarly for the training and testing sets. Then, we want a high recall on the testing set ($1 - r_{L_t}$ close to zero) and a small amount of unlabelled location classified as picture locations (r_U is small). We increase the weight of the testing set to force the retained solutions to have r_{L_t} close to 1.

6.4.7 Validation

Finally, we keep the set L_V away from the training (while L_t is used to fit the parameters). Its role is then to ensure that our method is valid on an independent set. We will thus measure r_{L_V} , the recall on the validation set. However, as discussed above, in One-Class problems it is easy to have high recall by adopting a low threshold (every picture location is correctly classified). r_{L_V} has to be evaluated together with r_U .

6.5 Data

The experiments consider one of the political regions of Switzerland located in the Swiss Alps: "Valais-Wallis". The area is the first part of the Rhône watershed bounded by Geneva Lake to the West and by some of the highest summits of the Alps. It encloses two of the most attractive tourist spots in Switzerland: Zermatt and the Aletsch glacier. The altitude gradient is considerable between the lowest area on the Geneva lake shore (450m) and the highest summit the "Dufourspitze" (4634m). The area is a valley, whose bottom hosts small to mid-sized villages. Climbing the flanks, low altitudes are generally covered with vineyards or forest depending on the orientation (500-700m); then forests, mountain villages and resorts are found in the range from 700 to 1400m; above 1400m, pastures and slopes dedicated to ski give access to the highest peaks, playground for the alpinists. The political border encloses then an area, which attracts tourists interested in nature and mountains activities.

The Swiss Topographic Agency (Swisstopo) provides a DEM and a Topographic Landscape Model (TLM). The TLM contains vector layers of several territorial objects. These two datasets are the source of our geographic indicators. Regarding the TLM, we selected only the "roads", other transportation facilities, "forests" and "lakes", which are expected to have an impact on picture locations. The geographic indicators are described in Table 6.1. The image locations are extracted from the Flickr database and were provided by Wider et al. (2013). These images are filtered to keep only locations outside built areas and geotagged with a GPS device. As stated above, we want to learn the good places in term of landscape features: images taken in built-up areas tend to capture the presence of a village or a touristic attraction rather than a natural landscape. The set L of image locations retained contains 2683 points, which are then separated into the three subsets represented in Figure 6.13, 135. The first set, circles, is used to train the methods L_T ; the second one, triangles, is used to fit the free parameters of the methods L_t ; the third one, stars, is used for the validation to compute independent statistics of the results L_V . To generate these three sets, a grid with a 5km cell side is generated over the area. Then, each grid cell is randomly attributed to one of the subsets, in order to obtain spatially uncorrelated labelled sets of approximately equal size.

Chapter 6. Learning prior: Measuring attractiveness of a location

Name	Abbrev.	Data	Description
Altitude	Z	DEM	Altitude of the location.
Curvature	Curv	DEM	Curvature measured in the location.
Slope	Slope	DEM	Slope measured in percent.
Visible Sky	Sky	DEM	The amount of visible sky. It is maximal in a flat area and minimal in a incised valley.
Positive Openness	PosOp	DEM	Measures the solid angle intercepted by the relief. It is quite similar to the curvature.
Negative Openness	NegOp	DEM	Measures the solid angle within the relief.
Distance Lake	DLake	Hydrology	Distance to the nearest lake.
Presence of a Lake	BLake	Hydrology	Binary indicator of the presence or absence of a lake.
Distance Road	DRoad	Road	Distance to the nearest road.
Presence of Road	BRoad	Road	Binary indicator of the presence or absence of a road.
Road Type	TRoad	Road	Type of road. Smallest values indicate trails, largest values highways.
Distance Forest	DFor	Land cover	Distance to the nearest forest (negative within a forest).
Presence of a Forest	BFor	Land cover	Binary indicator of the presence or absence of a forest.
Distance to cable car	DLift	Public Transp.	Distance to the nearest cable car or ski lift.

Table 6.1 – Description of the geographic indicators.

Finally, a regular grid of points separated by 100m and covering the whole region constitutes the unlabelled set U , which is used both for the comparison with L and for the mapping.

6.6 Results

First, we will observe the difference between the distributions of the picture locations and the unlabelled locations. This comparison will provide us a first indication whether the geographic indicators chosen are appropriate to distinguish both sets (Section 6.6.1). Second, we will analyze a typical experiment which includes every geographic indicator (Section 6.6.2). Third, we will conduct various experiments using different combinations of geographic features and we will compare the performance of our method with respect to OCSVM and KPCA (Section 6.6.3). Finally, we will present and comment resulting probability maps (Section 6.6.4). These two last sections constitute the validation of our method as defined in Section 1.4.3 (p. 9). We will use an independent validation set to ensure that our method can correctly detect these picture locations and will analyse the map resulting from the classification.

6.6.1 Distribution of the locations with respect to the geographic indicators

Based on the histograms presented in Figure 6.7, we will explain how the distribution of picture locations differs from the distribution of the unlabelled locations.

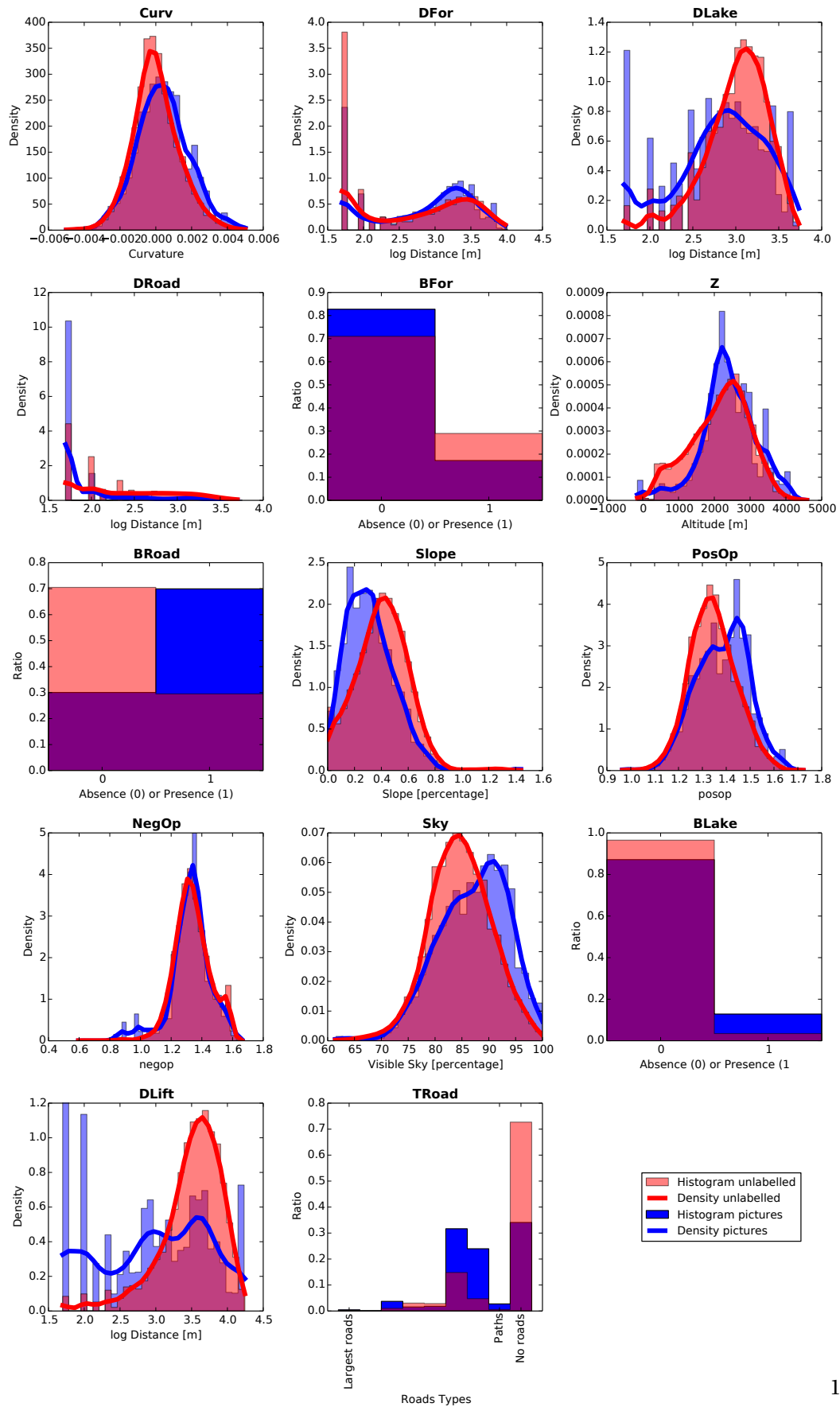


Figure 6.7 – Histograms and kernel density of the picture locations (blue) compared to the unlabelled set (red).

Chapter 6. Learning prior:

Measuring attractiveness of a location

Altitude (Z): Since we are focusing on landscape photography, few images are shot between 450 and 1500m (where the main attractions are the cities and the villages discarded from the dataset). On the contrary, the range between 1800 to 2200m (over the forest limit, where the ski slopes are found) is very attractive. There is a small mode above 3500m probably representing pictures taken by alpinists at high altitude.

Curvature (Curv): The histogram of the pictures is slightly shifted to the right in the direction of the positive curvatures (convex area); indeed ridges are more attractive than the valley bottoms and apparently also than flat areas.

Slope (Slope): The flatter locations with slopes going from 5% to 30% are preferred.

Visible Sky (Sky): This indicator confirms the result observed with the curvature: cells with a high ratio of visible sky (>90%) are more often chosen.

Positive openness (PosOp): The behavior of this indicator is similar to the curvature. Locations with larger viewsheds are preferred.

Negative openness (NegOp): No clear tendency appears for this indicator.

Distance to the nearest lake (DLake): People tend to take more pictures in a radius of 2000m around the lakes.

Distance to the nearest road (DRoad) and presence of roads (BRoad): Cells close to roads and paths are more attractive. Approximately 70% of the pictures are taken from a cell containing a road or a path.

Roads Types (TRoad): The roads are classified according to their width going from motorway to spur of trails. Cells which are crossed by several types of roads are assigned to the widest road. Path and unpaved roads are privileged by picture locations.

Distance to the nearest forest (DFor) and presence of forest (BFor): The unlabelled and picture location distributions are very similar. However, within the forest (negative values) and close to the forests (<700m) images are slightly less represented than grid cells. The trend is inverted between 700m and 3km, where more images are found. More than 80% of the pictures are shot outside of the forests.

Distance to the nearest cable car (DLift): Half of the pictures are taken within a range of 1500m surrounding a lift.

In order to measure the significance of these features, the unlabelled data and the picture locations distributions are compared. For each of the geographic feature chosen, their distribution diverges (tested by a Kolmogorov-Smirnov test with $\alpha = 1\%$).

One can see from the distribution of the indicators that some of them are correlated. A correlation matrix for the non-binary indicators is presented in Figure 6.8. First, we can focus on the morphological indicators. As expected the curvature is correlated with the positive openness and the visible sky. Altitude is negatively correlated with the negative openness. Low altitudes have high negative openness and inversely, it makes sense since deep valleys are mainly found at low altitudes. The distance-based indicators also present some similarities. The distance to the nearest road and the distance to the nearest forest are correlated. Indeed, the road and path network is denser at lower altitude where the forests are also found. Hence, these two latter are correlated with altitude.

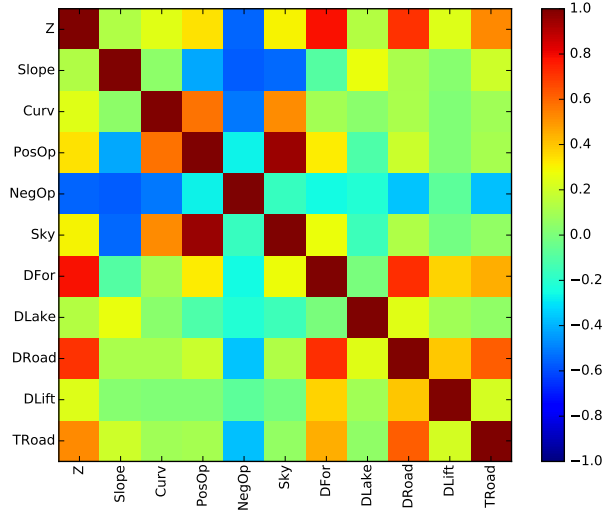


Figure 6.8 – Correlation matrix of the geographic features. Morphological indicators are on the left and vector-based indicators on the right.

6.6.2 Analysis of a PCA of every geographic features

For a general understanding of the method, we will discuss the result of the classification for all the geographic indicators projected with a PCA. With this setting, there is a small number of parameters to be fixed (the number of principal components n_{PCA} and the constant $P(Y = 1) = a$) and the linear relations are suppressed.

In Figure 6.9, the recall for the three sets of image locations (r_{L_T} , r_{L_t} , and r_{L_v}) as well as the ratio of positive of the unlabelled set (r_u) are computed for $n_{pca} \in [1 \dots n_{GI}]$ (where n_{GI} is the number of geographic indicators). It appears that if only a few principal components are selected, the unlabelled locations are not well separated from the image locations. Hence, the recall is very high, but the number of location classified as positive in the unlabelled set is also very high. Once that the number of principal components is increased, the difference between the unlabelled locations and the image locations becomes also more pronounced. However, the classification becomes more and more specific to the training set. The role of the cost function (Equation 6.11) is then to fix the limit until which the difference between the training set and the testing set is acceptable to avoid overfitting. It also ensures that the ratio of unlabelled locations classified as positive remains small.

In Figure 6.10, we show the impact of the threshold value λ of the Equation 6.9. For that, we fix $\lambda = 0.5$ and let $a = P(Y = 1)$ variate (Equation 6.8). Thus, a acts here as a scaling factor. This result is presented for the best solution detected in Figure 6.9: $n_{PCA} = 8$ components. A small a generates a small amount of regular cells classified as image locations, but a considerable amount of false negative is also generated (many pictures of the testing set are misclassified). This types of solutions are the ones that maximize the F_{pu} criterion and justify the use of another criterion (remind that F_{pu} is to be maximized, Equation 6.10). Oppositely, if a rises, the number of false positive decreases but the ratio of unlabelled locations classified as positive rises also. One can see that the difference between the three pictures location sets does not evolve much with respect to a . The recall curves tend to remain parallel. The goal of the criterion is here to find the trade-off between the recall r_{L_t} and the ratio of unlabelled location r_U classified as positive. We can also notice that the validation set seems to be more similar to

Chapter 6. Learning prior: Measuring attractiveness of a location

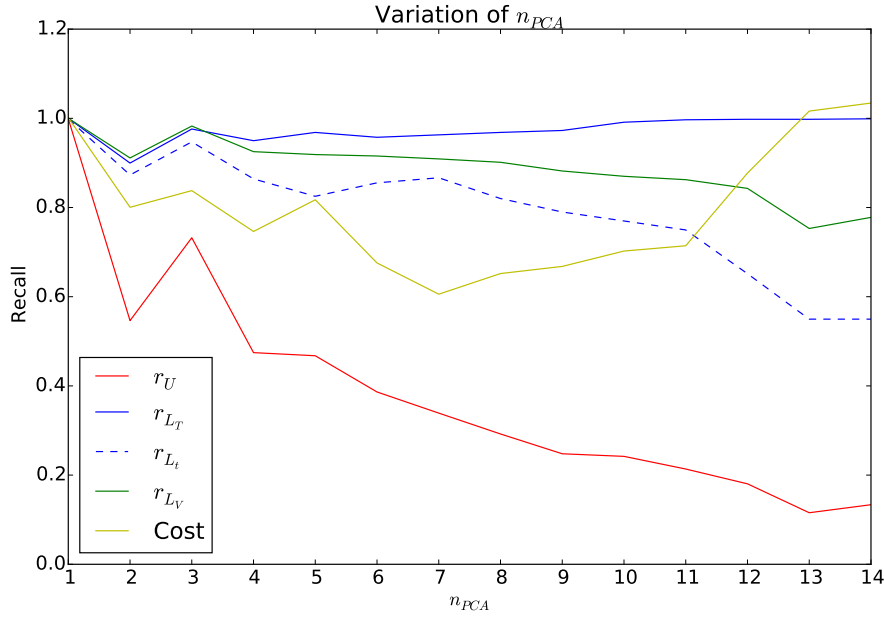


Figure 6.9 – Influence of n_{PCA} on the recall of the labelled sets, unlabelled set and cost function. If n_{PCA} is small the sets are not separated, if n_{PCA} is close to the number of geographic indicators, the classification overfits the training set.

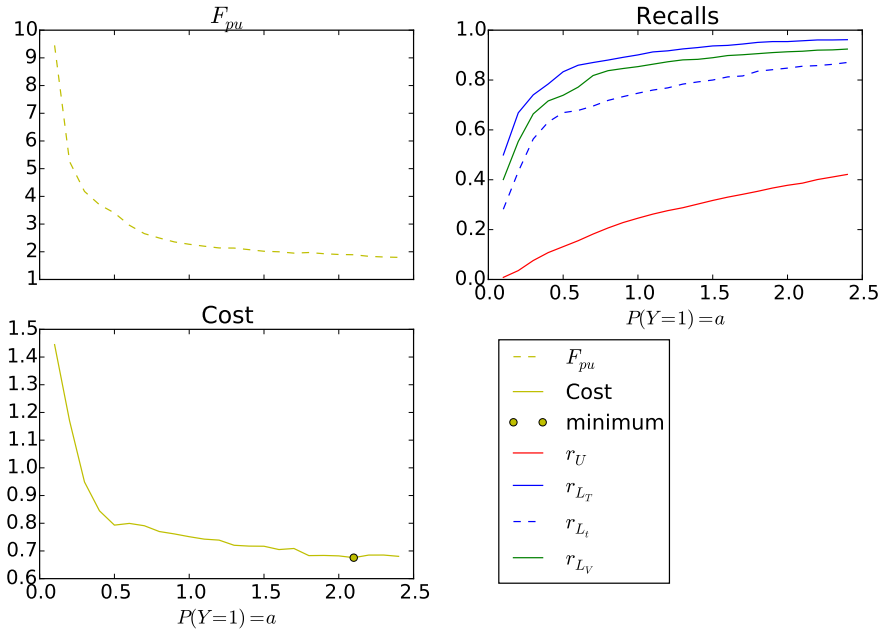


Figure 6.10 – Influence of the threshold on the recall of the labelled sets, unlabelled set and cost functions. The x -axis shows the value of $a = P(Y = 1)$ and the threshold λ is set to 0.5 (Equations 6.8 and 6.9).

No.	Name	Z	Curv	Slope	Sky	PosOp	NegOp	DLake	BLake	DRoad	BRoad	TRoad	DFor	BFor	DLift
1	All	•	•	•	•	•	•	•	•	•	•	•	•	•	•
2	DEM	•	•	•	•	•	•								
3	Dist							•		•			•		•
4	Bin								•		•	•		•	
5	Scenic	•	•	•	•			•					•		
6	MostCorr	•		•	•			•		•		•	•		•
7	MC+BRoad	•		•	•			•			•	•	•		•
8	MC+BFor	•		•	•			•		•		•		•	•
9	MC+BLake	•		•	•				•	•		•	•		•
10	4C+DFor	•		•	•					•			•		
11	4C+DLake	•		•	•			•		•					
12	4C+BRoad	•		•	•					•	•				
13	4C+TRoad	•		•	•					•		•			
14	4C+NegOp	•		•	•		•			•					
15	4C+DLift	•		•	•					•					•

Table 6.2 – . Experiments and selected geographic indicators.

the training set than the testing set. Hence, in the next results, we can expect to have a better recall on the validation set than in the testing set.

6.6.3 Numerical results for various combination of geographic features

6.6.3.1 Experiment descriptions

In Table 6.2, we report 15 experiments (rows), each one considering various combinations of geographic features for GOCD. The first experiment (**All**) mixes all the features described above. The three next experiments consider respectively the morphological indicators only (**DEM**), the distance-based indicators only (**Dist**) and the binary indicators only (**Bin**). In the fifth experiment, the geographic indicators related to landscape only are selected, without accessibility-related indicators. This experiment can be considered as a measure of the "scenic-ness". In the next experiment, based on the histograms, we select visually the best uncorrelated indicators (MostCorr). In the experiments 7-8-9, we use the same combination as MostCorr but we replace respectively the DRoad by RType, DFor by BFor and DLake by BLake. Finally, for the last experiments (10-15), we took the four indicators that appear the most correlated with the pictures (DRoad, Slope, Z and Sky) and add another indicator (DFor, DLake, BRoad, TRoad, NegOp, DLift).

6.6.3.2 Comparison of the methods

We will compare five methods. The first three (KDE, KDE(PCA), KDE(KPCA)) are the ones presented earlier with or without projection of the original indicators. Their parameters are chosen to rise the best cost. Next, the One-Class method OCSVM (Schölkopf et al., 1999) and KPCA-nd (Hoffmann, 2007), which do not consider the unlabelled data, are presented as comparison. OCSVM and KPCA-nd also have various parameters to set, we chose those which maximized F_{pu} .

In the experiments, KDE(PCA) is better than KDE only if the number of components selected by the

**Chapter 6. Learning prior:
Measuring attractiveness of a location**

	KDE					KDE(PCA)					KDE(KPCA)				
	r_{LT}	r_{Lt}	r_{LV}	r_U	Cost	r_{LT}	r_{Lt}	r_{LV}	r_U	Cost	r_{LT}	r_{Lt}	r_{LV}	r_U	Cost
All	1.0	0.59	0.81	0.15	0.96	0.96	0.87	0.91	0.37	0.64	0.95	0.84	0.9	0.27	0.59
DEM	0.99	0.72	0.89	0.7	1.25	0.91	0.83	0.78	0.53	0.87	0.94	0.87	0.84	0.57	0.84
Dist	0.89	0.84	0.92	0.36	0.67	0.89	0.84	0.92	0.36	0.67	0.93	0.9	0.94	0.44	0.65
Bin	0.97	0.98	0.97	0.85	0.9	0.97	0.99	0.98	0.87	0.89	0.97	0.98	0.97	0.85	0.88
Scenic	0.98	0.78	0.83	0.5	0.94	0.98	0.78	0.83	0.5	0.94	0.99	0.84	0.86	0.56	0.88
MostCorr	0.98	0.75	0.85	0.2	0.7	0.95	0.88	0.93	0.39	0.62	0.96	0.84	0.89	0.21	0.54
MC+BRoad	0.99	0.8	0.89	0.28	0.68	0.93	0.86	0.9	0.34	0.61	0.93	0.89	0.93	0.37	0.58
MC+BFor	0.97	0.78	0.88	0.22	0.66	0.94	0.89	0.92	0.35	0.58	0.93	0.83	0.89	0.21	0.56
MC+BLake	0.99	0.87	0.92	0.39	0.66	0.93	0.89	0.93	0.37	0.59	0.94	0.86	0.91	0.31	0.59
4C+DFor	0.91	0.8	0.81	0.24	0.64	0.91	0.8	0.81	0.24	0.64	0.89	0.81	0.81	0.27	0.64
4C+DLake	0.89	0.78	0.8	0.22	0.66	0.89	0.78	0.8	0.22	0.66	0.94	0.83	0.87	0.3	0.64
4C+BRoad	0.88	0.8	0.81	0.25	0.66	0.88	0.8	0.81	0.25	0.66	0.93	0.87	0.89	0.38	0.65
4C+TRoad	0.94	0.84	0.88	0.34	0.67	0.94	0.84	0.88	0.34	0.67	0.87	0.81	0.84	0.26	0.64
4C+NegOp	0.95	0.69	0.85	0.32	0.94	0.9	0.77	0.84	0.27	0.72	0.86	0.79	0.82	0.25	0.66
4C+DLift	0.97	0.85	0.9	0.38	0.67	0.97	0.85	0.9	0.38	0.67	0.92	0.89	0.89	0.34	0.56

	OCSVM					KPCA-nd				
	r_{LT}	r_{Lt}	r_{LV}	r_U	Cost	r_{LT}	r_{Lt}	r_{LV}	r_U	Cost
All	0.95	0.89	0.96	0.95	1.18	1.0	1.0	1.0	1.0	1.0
DEM	0.95	0.88	0.99	0.95	1.19	1.0	1.0	1.0	1.0	1.0
Dist	0.95	0.94	0.98	0.93	1.05	1.0	1.0	1.0	1.0	1.0
Bin	0.91	0.92	0.91	0.82	0.98	0.96	0.97	0.97	0.86	0.92
Scenic	0.94	0.92	0.9	0.93	1.09	1.0	1.0	1.0	1.0	1.0
MostCorr	0.95	0.91	0.97	0.95	1.13	1.0	1.0	1.0	1.0	1.0
MC+BRoad	0.95	0.92	0.97	0.96	1.11	1.0	1.0	1.0	1.0	1.0
MC+BFor	0.95	0.92	0.96	0.95	1.11	1.0	1.0	1.0	1.0	1.0
MC+BLake	0.95	0.9	0.95	0.94	1.13	1.0	1.0	1.0	1.0	1.0
4C+DFor	0.9	0.85	0.9	0.81	1.1	0.97	0.93	0.96	0.76	0.89
4C+DLake	0.94	0.92	0.92	0.84	0.99	0.94	0.9	0.93	0.65	0.85
4C+BRoad	0.79	0.79	0.84	0.64	1.05	0.9	0.85	0.91	0.56	0.86
4C+TRoad	0.96	0.92	0.94	0.94	1.1	0.91	0.82	0.86	0.57	0.93
4C+NegOp	0.96	0.88	0.97	0.93	1.17	1.0	1.0	1.0	1.0	1.0
4C+DLift	0.95	0.93	0.96	0.95	1.08	1.0	1.0	1.0	1.0	1.0

Table 6.3 – . 15 Experiments for the three methods proposed in this work (KDE, KDE(PCA), KDE(KPCA)) dedicated to GOCD and the standard methods OCSVM and KPCA dedicated to OC problems (which do not consider the unlabelled data).

cost function is smaller than the number of geographic features. For instance, in some experiments, the cost is similar for both methods (**Dist** etc.). These cases correspond to a number of principal components selected equal to the number of dimensions. Hence, the benefit of the PCA appears generally for a number of geographic features higher than five. In these conditions, the PCA selects uncorrelated axes and improves the results. The KDE reaches the best cost for experiment **4C+DFor** for which 81% of the validation set is correctly classified and 25% of the unlabelled locations are classified as positive. However, KDE(PCA) has the best performance for experiment **MC+BFor** with 92% of recall and 32% of unlabelled locations detected as likely locations.

KDE(KPCA) is always better than KDE(PCA): the projection in the space of higher dimension is generally valuable. The best experiment here is reached by the indicators selected to be mostly correlated with the picture locations (**MostCorr**) for which we obtain $r_{LV} = 0.89\%$ and $r_U = 0.21\%$.

These three methods are better than both KPCA-nd and OCSVM, which experience difficulties to limit the amount of unlabelled locations classified as positive. For instance, the experiment number **4C+DLake** is the best for KPCA for which 65% of the regular locations are positive. For all the other experiments, this set is even closer to the validation set. Hence, we can say that if the unlabelled locations are not taken into account during the classification, these classifiers have difficulty to extract the uniqueness of the picture locations.

6.6.3.3 Features selection

The goal of this classification is not only to separate the pictures locations from the others but also to extract the relevant features. In the experiment **All**, we insert all the geographic features and in the experiment **MostCorr**, we select the geographic features ourselves. Interestingly, both settings induce close performance for KDE(PCA) and KDE(KPCA) while the KDE in the original space shows a good performance only for the second set of features. Hence, especially after a kernel projection, PCA reduces the number of dimensions (which is good for processing) and selects the best geographic features to improve the performance.

6.6.3.4 Cost function

We showed in the previous chapter that the choice of a proper cost function is very important in one-class problems. Hence, for the evaluation of the results it is important to be interested not only in the recall r_{LV} but also in r_U . In the best experiment, r_{LV} is 0.89 and r_U is 0.21, the cost function is then able to indicate a solution which has a high recall and is also able to generate few unlabelled locations classified as positive.

6.6.3.5 Geographic indicators

Eight geographic indicators are selected for the experiment **MostCorr**, which induces the smallest cost. Most of them are related to the accessibility (DRoad, DLift, TRoad) or inaccessibility (Slope, Z). However, some others are more related to the visibility or *scenic-ness* (DFor, DLake, Sky).

6.6.4 Evaluation of the map

6.6.4.1 Evaluation of one of the most attractive area in the Alps: Zermatt

Probabilities for the Zermatt area are presented on the map in Figure 6.11. The dots represent the training set, the triangles the testing set and the stars the validation set. Red symbols represent misclassified locations. Image locations are superposed to the results of KDE(KPCA) for the combination **MostCorr**. The colors correspond to the probability estimate of $p(Y = 1|z)$ in each 100m cell of the map. A misclassification (red marker) corresponds then to a picture located in an area of low probability (in red). The validation set is very specific in this area and located mainly in the skiing and touristic area of the *Gornergrat*. Hence, we have a particularly high recall for this region. The misclassifications are mainly pictures shot by people adept of mountaineering. In other words, locations represented in red are generally in remote areas and away from a trail. Note that, in the very south of the area, the DEM is corrupted and the indicators (and thus the probability estimate) are not representative.

One can see that the paths have a great influence on the probability estimate and they appear as linear patterns. However, their values are not constant and may change according to the morphology or the altitude of the locations (**A** highlights a path along which the probability varies). Cable-cars generate also linear, but more diffuse, patterns (**C**). This could be linked to the fact that they draw a general directions of the ski slopes (from which skiers can deviate). Here also, their influence can be mitigated by the morphology, as in the zone highlighted by the arrow **B**. Finally, away from the paths and the lines of cable cars, the morphology has the main influence. Ridges, summits and passes, as well as flatter areas (on the glacier) generate favorable probabilities.

6.6.4.2 Evaluation of the entire area

The Figure 6.12 is a North-South transect of the valley. In the plain, the probabilities are generally low, but influenced by the proximity of roads. One exception is easily explicable by the presence of two small lakes (**1**). Then, for two similar flanks (**2-3**) in altitude and orientation, the steeper one without roads (**2**) is less probable. The marker (**4**) shows the diffuse influence of a cable car compared with the influence of a trail (**5**). The trail has also a higher probability than the small road within a forest (**6**). The ellipse (**7**) highlights the low probabilities in incised valleys. On its left, a skiing area appears clearly. In the ellipse **8** and **9**, one can see that ridges, summits and flat regions on glaciers are slightly positive. The highly positive region within the marker **10** is hardly explainable. Our guess is that it is morphologically similar to tourist regions, such as Zermatt, and inherits their high probability. In a general view, the probabilities are more contrasted in low to middle altitude than in the higher altitudes (above 2500m) where less training pictures are available and the influence of the morphological indicators predominates.

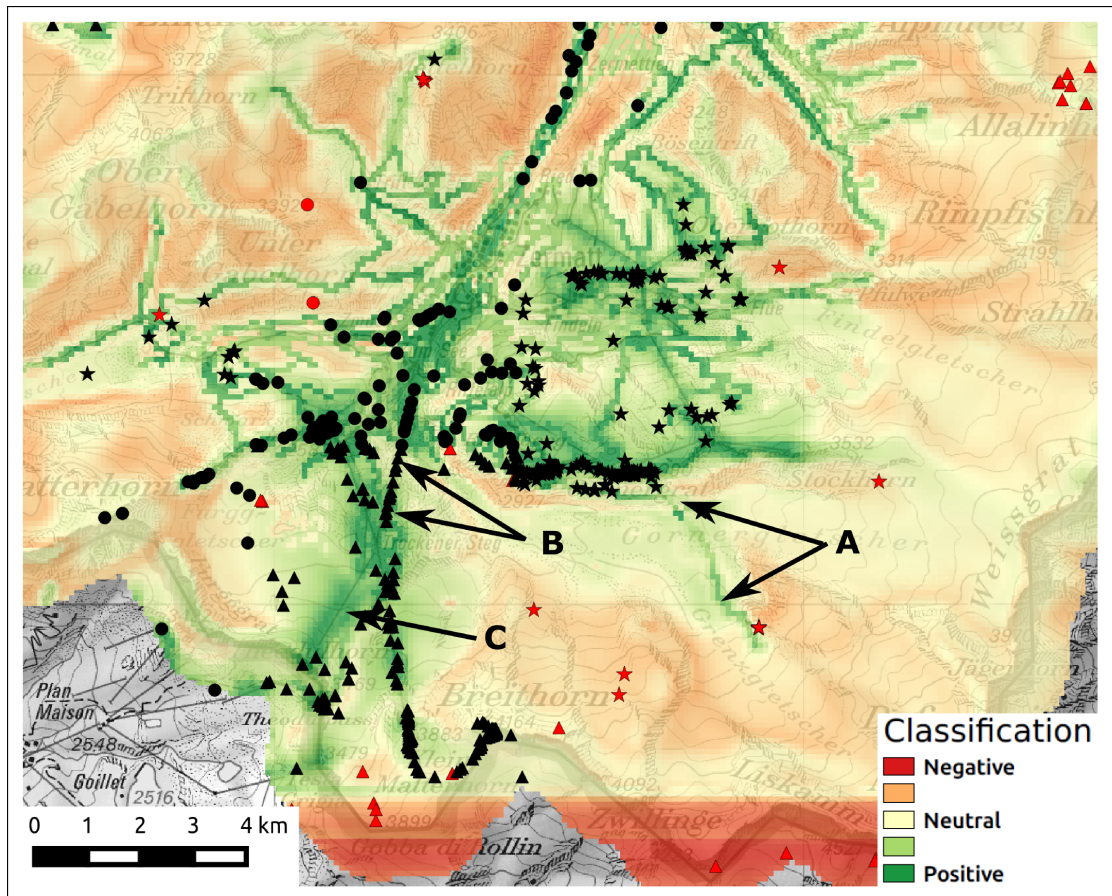


Figure 6.11 – Probability map resulting from the **MostCorr** experiment. Red cells are unlikely, whereas green cells are attractive to shoot pictures. The red band on the bottom of the figure is not representative and due to a problem of the DEM in this area. Circles are training locations, triangles testing locations, and stars validation locations. Red markers correspond to misclassified locations. (*Swisstopo*)

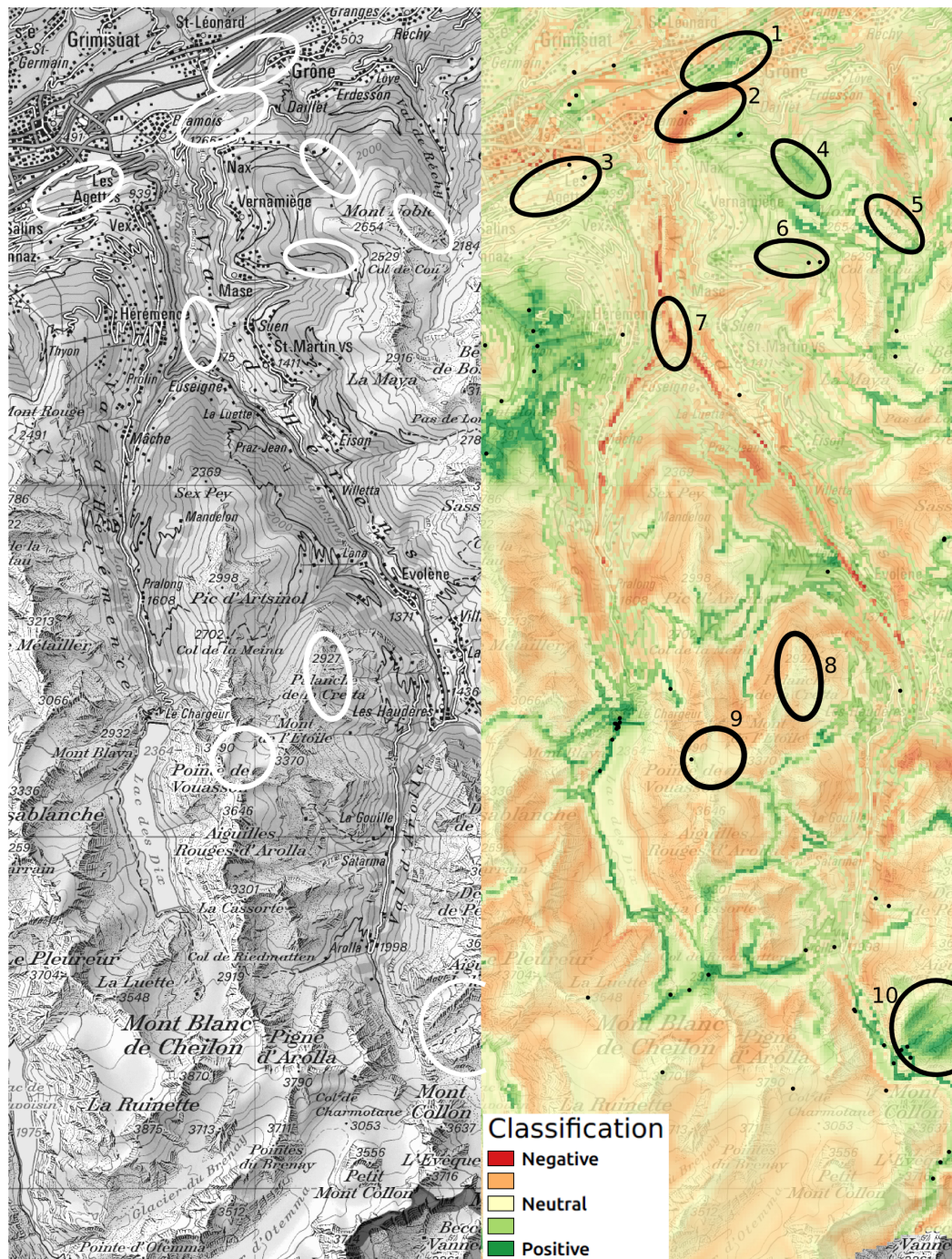


Figure 6.12 – Estimate of the probability for a N-S transect resulting from the **MostCorr** experiment. The topographic map is shown on the left, the corresponding classification result on the right. The top of the map is within the valley, the bottom reaches some high summits of the Alps. Red cells are unlikely, whereas green cells are attractive to shoot pictures. (*Swisstopo*)

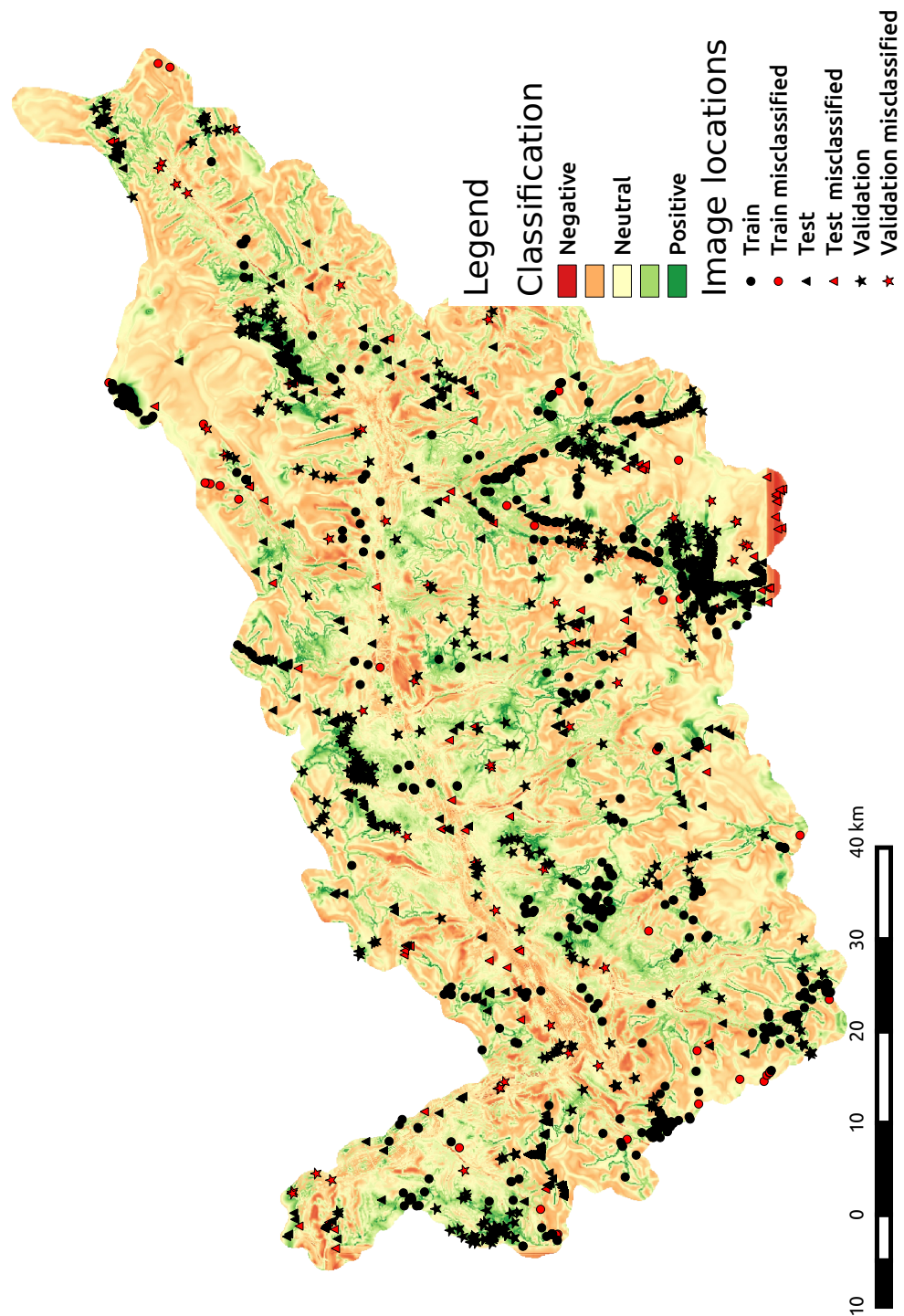


Figure 6.13 – Estimate of the probability for the entire area. The training, testing and validation set are represented with dots of various shapes.

6.7 Discussion

6.7.1 Method

The improvement between the proposed methods (KDE, KDE(PCA) and KDE(KPCA)) and the standard ones (OCSVM, KPCA-nd) is mainly due to the insertion of unlabelled data in the classification process. Our method is easy to implement and shows good results without fine tuning. The force of our method is to learn attractive combinations of geographic indicators. The projection of the geographic indicators in a higher dimensional space generated by the kernel projection seems to improve the results. Here, we propose a simple approach to benefit from the kernel projection, which can probably be defined more rigorously. Specifically, Munoz and Moguerza (2004); Cremers et al. (2003) suggest that the KDE in the original space of the geographic feature is strongly related to One-Class SVM and KPCA-nd. Hence, the classification with these methods would be equivalent to the setting of a density threshold in the original space. These affirmations would then suggest that the performance of KDE and KDE(PCA) could reach the one of KDE(KPCA) if a more appropriate bandwidth h is chosen rather than fixing it with Scott's rule.

We proposed a cost function to determine the best set of parameters for a method. We choose this function to promote high recall and a low ratio of unlabelled location classified as positive. Another strategy could be to fix the expected recall, (for instance, estimate that 10% of the picture locations are not related to the proposed geographic features, fix r_{L_t} to 90%) and choose the parameters which minimize r_u . However, in our context, we did not have this estimation. Enriched of this experiment, we could now consider fixing r_{L_t} to 90%. This would provide also a more reliable comparison of the experiments.

6.7.2 Geographic indicators

In this study, we consider the choice of an appropriate location to shoot a picture and consider only indirectly the visibility of a point of interest (with the indicator related to the field of view).

The accessibility is a factor, which impacts the distribution of the pictures locations. At global scale, areas reachable by roads or cable cars are preferred to remote locations. At the local scale, cells crossed by a path are more likely to be a picture location. The slope also influences the accessibility. For instance, flat regions are preferred to cliffs. Appropriate locations are also distinguished by a large field of view. Thus, ridges are more suitable for photography than incised valleys or locations in the middle of a forest. Finally, scenic locations are attractive to photographers. However, it is more difficult to generate an indicator that describes the *scenic-ness* of a place. The indicator, which is certainly the most directly linked to that, is the proximity of a lake. Other indicators such as the visible sky and curvature are more related to the field of vision. In our approach, we let the data drive the mixing and weighting of these indicators to generate the map of the likely locations to shoot pictures. Our method is not thought to the selection of the relevant indicators. On the contrary, with the PCA and KPCA, we want to be able to insert various indicators and let the process weight and select the important ones. We showed that if the indicators are selected (by studying histograms) to be highly correlated with the picture location, it improves the result obtained with all the indicators. However, the improvement is not drastic and proves that PCA and KPCA can reduce efficiently the dimensions of the data to improve the probability estimate.

We selected indicators expected to be related to picture locations. The histograms presented earlier

prove that they are. However, other indicators could have been computed (proximity to a summit, proximity to glacier, ski runs etc.) or variation of the ones proposed (DEM-based indicators at finer or coarser scale), but we tried to keep the list small and varied to avoid too much correlation and heavy processing. Finally, we did not insert indicators related to the tourism (proximity to a point of interest, proximity to restaurants) to avoid to get results similar to spatial density estimation and keep the results generalizable to less touristic regions.

6.8 Conclusion

In the previous chapter, we discuss the potential of 3D landscape to determine the fine orientation and location of a picture. In this chapter, also we would like to draw the attention on the potential of geographic data, but to determine a probable location of a picture. Pose estimation assumes that every location on the map is equiprobable. However, for example in term of accessibility, it seems obvious that easily reachable locations are more likely to be a shooting spot.

State of the art methods that produce attractiveness maps or determine potential locations of an image are based on the spatial density or spatial interpolation from a training set of picture locations. However, we saw that the spatial distribution of the pictures shared by the public is biased by points of interests where a lot of pictures are shot, while most of the regions are only sparsely photographed. Hence, these methods are not suitable for areas with low density of images or no images at all.

Rather than using the spatial dimension to interpolate such probability, we propose to compute the probability in a space made of geographic features mostly related to accessibility and visibility. With this space, we can compute a probability in every location of a map and not only in the vicinity of other pictures. Our method is then close to Tsung-Yi et al. (2013), who compute the probability in a space made of visual features. The data set used to train our one-class classifier is made of web-shared images, which are accurately localized with a GPS. We show that this problem can be solved by a Geographic One-Class approach. Specifically, it can be seen as a one-class classification task, in which the locations which are likely to be shooting locations have to be separated from the rest. In addition, our GOC approach also includes the distribution of the unlabelled locations (i.e. every location, including good locations to take pictures) to improve the classification. We propose a method based on the KDE to estimate the PDF of both unlabelled and pictures locations. The Bayesian relation between them is then used to estimate the probability of a location given its geographic features. The geographic indicators proposed are correlated. We compare then the result of our method applied to the original dimensions, with two projections of the original dimension: a PCA projection and a projection in a higher-dimensional space (KPCA). The results are evaluated with a cost function which synthesizes the recall on the testing set and the amount of positive location in the unlabelled set (that we want to keep small). In the best experiment, the cost function highlights a combination in which 89% of the validation locations are correctly classified and only 21% of the unlabelled locations belong to the positive set.

The next step would be to use these learned relations to improve the georeferencing of pictures. In this thesis, we did not propose a robust key-point detector and descriptor, which can trustfully detect correspondences in an image and in a 3D landscape model. Hence, we can imagine a workflow in two steps for the 6DOF orientation of a camera. First, the region where a picture is shot is estimated (by an operator or with the metadata, circle in Figure 6.14(b)). For each location in the region, a panorama is computed and its similarity to the query image measured. In this process, the probability measured in this chapter can be used either to discard unlikely locations or to promote likely ones. The space

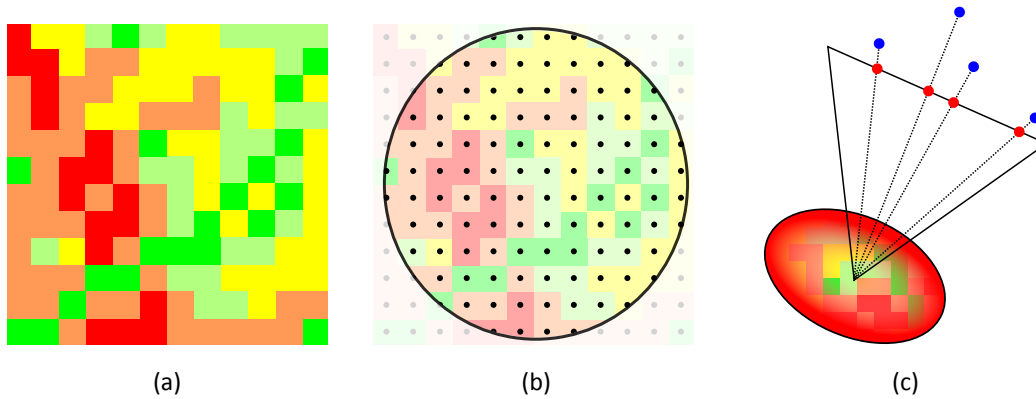


Figure 6.14 – Potential utilisation of the map of attractiveness: (a) Map, (b) Potential locations of a picture, a synthetic image is generated for each location and compared to the query picture. The probability values are used to promote or discard locations. (c) 2D-3D correspondences locate a picture in a confidence ellipse. The probability values highlight most probable locations in this region.

of possible locations is then reduced, which could lead to a better efficiency (less computationally intensive and more accurate). At this level, if there is an indication of what is visible in the picture (provided by the metadata or the operator), the subset of possible locations could be further reduced by a visibility analysis.

Next, 2D-3D correspondences are detected (Figure 6.14(c)), they are typically not as accurate as the ones digitized by an operator. Hence, the pose estimated is associated with a large confidence region. A combination of the confidence ellipse computed with the 2D-3D correspondences and the probability proposed in this Chapter could highlight the most probable location(s).

This method could also be valuable for localization at larger scale. Usually, studies which aims at retrieving a picture location at large scale, train the algorithm with a database of pictures shared on the web. They relate visual features of the query image with geolocated visual features of the training database. In this way, they are affected by the inhomogeneous spatial distribution of the pictures. As proposed by Lee et al. (2015); Tsung-Yi et al. (2013), if visual features of the images are linked to geographic features, the bias engendered by the sampling is avoided and can result in a spatially improved estimate of the localization. Thus, we can see a two-steps extension to our work. The first step is to classify the pictures according to categories and reproduce our study. Hence, a picture will be firstly associated to a category (e.g. "mountain+winter") which limits its possible locations. In the second step, we could learn a relation between the visual content and geographic attributes. Indeed, with the rise of pictures shot with smartphones, providing an azimuth estimate, and methods such as developed in this thesis, we will also be able to relate the visual content with visible regions. For instance, Chippendale et al. (2008) map the attractiveness of a landscape by measuring how many pictures are looking at a region. This second step would join the work of Tsung-Yi et al. (2013) who associate visual contents with geographic features to measure the visual content probability. However, it could be improved by considering the accurate visible regions and by studying also the location probability.

7 Conclusion

7.1 Goals

The goal of this thesis is to propose and implement solutions for the georeferencing of landscape images. The issue under scrutiny is a single image, which is to be oriented in a virtual globe efficiently. The methods developed should have a broad range of application and thus be able to:

- solve the georeferencing in different kinds of rural regions,
- be robust to various landscape appearances (such as those induced by daylight and seasonal variations and various camera types),
- be compatible with user supervision (that may still be necessary for the georeferencing of the most challenging images),
- take into account overlapping images georeferenced previously.

To answer these needs, we want to rely on reference data available everywhere: geographic data such as DEM describing the landscape morphology and orthoimages describing the land cover. The correspondences between these geographic data and a picture are easier to detect through a 3D rendering, which generates synthetic views with a natural viewpoint.

First, we aim to provide a computer program dedicated to the supervised georeferencing of images and enabled with a 3D browser. This program should also have other functionalities in anticipation of the exploitation of the georeferenced pictures (picture augmentation and geographic information extraction).

Second, we aim at exploring the possibility of automatically registering a picture with the 3D landscape model using image matching methods. In order to reduce the space of possible orientations and locations of the picture, we want to provide georeferencing methods both for the situations where the picture location is perfectly known (localized with a GPS) and for those for which we have only a reliable estimation of the location (such as a click on a map or an approximate orientation in a virtual globe). The methods developed should be insensitive to the various appearances that the landscape can have in a picture.

Throughout this research, we noticed that a likely solution to determine the location of a picture at larger scale (such as a region indicated by a toponym) would be to measure the similarity of the picture

with the 3D landscape model for multiple possible locations. Hence, we want to measure the likeliness of a location to be a shooting spot given its geographic description. This indicator could then reduce the amount of locations to be tested or to influence the pose estimation to promote likely locations.

7.2 Results: discussion of the objectives

The objectives and hypotheses are listed in Section 1.4 (p. 7). Hereafter, we discuss whether the conditions of validation are respected, they were formulated in a way that they ensure that the objectives are achieved and the hypotheses confirmed.

7.2.1 Monoplotter

A. Obj. 1: Facilitate the acquisition of 2D-3D correspondences:

- ✓ **Provide a 3D browser (to render geographic data with a natural viewpoint):** In our first contribution, we implemented a monoplotter with a 3D interface. The 3D environment can be browsed to estimate an initial pose of an image.
- ✓ **Provide GCP digitization in the 3D environment:** The 3D interface can also be clicked to digitize 3D GCP. Hence, the 3D interface mixes the user friendliness of virtual globes and the accuracy of pose estimation with GCP.
- ~ **Provide Ground Control Lines as an alternative to GCP:** We provided a method to align and extract correspondences from a 2D skyline and a 3D skyline extracted from a 3D model. A more general function for the pose estimation from linear features digitized by the operator (such as rivers, roads or forest contours etc.) can be directly derived from the skyline matching with DTW.

We did not express the conditions of validation for the monoplotter. Specifically, if it is further developed, we should consider to rigorously prove that 3D views ease the acquisition of GCP (for instance by performing an analysis of the time required by users to georeference a picture). However, our own experiences, experiences of our students and a discussion with C. Bozzini (author of Bozzini et al. (2012)), corroborate this statement.

✓ A. Obj. 2: Provide a direct interaction between a picture and the GIS:

Our monoplotter is implemented for one of the most widespread open-source GIS. Unlike other implementations, it is enabled with all the functions to overlay geographic layers on the picture (picture augmentation) and is capable of projecting pictures information in the GIS (monoplotting, orthorectification). It will require an additional investment to be proofed and consolidated, but it is already a valid alternative to other monoplotters.

7.2.2 Registration with a 3D landscape model

B. Val. 1: Skylines extracted from a real image and a landscape model are aligned and correspondences detected. The correspondences are used to retrieve the 3DOF and 6DOF orientation of a database of images having a ground-truth orientation:

- ✓ **Skyline Matching:** We propose DTW matching to align the skyline detected in a picture and in a 3D model. The warping behavior of DTW is appropriate to match deformed skylines (scale change, distortions due to the rendering of a panorama).
- ✓ **3DOF orientation:** We used DTW to align skylines in the scenario of a 3DOF orientation and successfully retrieve the orientation of a database of images (82% of the azimuths are within 5° of the ground truth). The alignment may fail if the skyline is too smooth, if the 3D model is incomplete or if the skyline is obstructed.
- ~ **6DOF orientation:** We slightly modified the method to recover the 6DOF orientation of the same database of images. Specifically, we shifted the real camera location by 100, 250, 500, and 1000m and tried to recover the ground truth location. The result is more mitigated. Firstly, our pictures have a low resolution, resulting in smoothed and poorly discriminative skylines. Secondly, in hilly reliefs, skyline variations observed in short-range displacement ($>200\text{m}$) are more than a simple deformation of the shape of the skyline (that the warping behavior of DTW could recover). Indeed, it appears that the skyline can change drastically and only a part of the skyline remains similar in the picture and in the 3D model. In these conditions, the alignment with DTW, which considers the entire skyline, becomes impossible. Thus, we can say that DTW is appropriate for skyline matching, but we observed that variations of the landscape with a displacement of the viewpoint may generate a new skyline shape, which can be beyond the capacity of DTW. This observation would suggest the implementation of another method to find the 6DOF orientation of a picture (which measures the skyline similarity for a grid of locations or considers local skyline features, like in Baatz et al. (2012a)).

Contributions: The warping capacity is the main difference and advantage of our skyline matching algorithm. It proves to ease the convergence of the skyline alignment. Compared to the more rigid methods found in the literature, it is also promising for the computation of the focal or the location of the camera (if the problem of occlusion is overpassed).

B. Val. 2: A metric to measure the similarity of (patches of) real and synthetic images is provided. This similarity measure provides 2D-3D correspondences exploited to retrieve the 3DOF and 6DOF orientation of a database of images having a ground-truth orientation:

- ✓ **Measure of appearance similarity:** We propose HOG description to detect correspondences between a real and a synthetic image. Typically, it provides noisy correspondences, which have to be processed further to eliminate false positives.
- ✓ **3DOF:** We compared a global matching (the entire picture is described with HOG) and a multi-scale and geometrically constrained matching of local patches. The local matching is more efficient and compatible with a real 3DOF orientation (whereas the global matching only provides the azimuth and the tilt). With this method, we recovered 62% (global matching) and 78% (local matching) of the ground truth azimuths.
- ✗ **6DOF:** Because of a lack of time, we did not test the 6DOF orientation. Similarly to skyline matching, we should consider two pose estimation schemes. For fine orientation and if there is a reliable prior location and orientation, we could use the PEP-ALP (the case study will thus be similar to Produit et al. (2012), but the method would gain from the developments presented in this thesis). At larger scale, we would need to generate panoramas for a grid of locations and determine the best location and alignment.

Contributions: The matching with a landscape model as proposed in this contribution is a way to take into account not only the depth discontinuities but also the land cover. The implementation remains simple and does not require a training step, but only the computation of the best parameters for the HOG descriptors. They can be optimized only once if the benchmark dataset is representative of a large variety of images and landscapes. We also demonstrate that the knowledge of geographic information (distance) can be injected in the matching process to promote or down weight regions of the reference pictures supposed to disturb the matching.

B. Val. 3: The methods are integrated to georeference a collection of various (accuracy of the geotag, appearance) and overlapping images. The quality of the georeferencing is assessed (accuracy of the localisation of a pixel):

- ✓ **Integration:** In Section 5.3, we integrated the skyline matching and PEP-ALP. Firstly, the skyline matching was exploited to orient images localized with GPS (3DOF orientation). Secondly, it was used to refine the 6DOF orientation of pictures having a less accurate prior location. At this stage, HOG matching was not inserted in the workflow but it could replace or interact with the skyline matching.

- ~ **Various images:** We downloaded a collection of images from a photo-sharing web-site (Panoramio 📍). This collection consists of images geotagged with various accuracies (GPS or user click on a map) and of images of various appearances (seasonal and diurnal variations, several types of cameras). For their pose estimation and inspired by the works of Moreno-Noguer et al. (2008); Serradell et al. (2010), we developed PEP-ALP a robust method for the simultaneous pose estimation and matching which relies on a Kalman filter (successive pose estimation) and on variance propagation (geometrically constrained matching). The geometric constraint induced by the DEM and pose prior limits the potential corresponding 2D-3D key-points and thus the measure of their appearance similarity can be relaxed. At this stage, our workflow still relies on SIFT to compute the initial azimuth of a picture. This limitation does not allow us to ensure that the relaxation of the appearance similarities (applied only in the next step) is pertinent to georeference pictures having a different appearance than the reference images. A required further development would be to estimate the initial orientation independently of SIFT, for example by a matching to the synthetic model or simply with an azimuth indication of the user.

- ✓ **Accuracy assessment:** The georeferencing is not accurate enough to extract geographic information from the pictures but sufficient to locate an image in a virtual globe or overlay the image with geographic data.

Contributions: SfM is the state-of-the-art for the georeferencing of pictures, but it is a solution only for particular set of overlapping and similar images. Hence, to orient every images of a collection, the involvement of a user has to be considered. The matching with a landscape model can reduce the user workload. We propose that the user provides an approximate location and orientation which is then further refined with the comparison with the landscape model. The Kalman filter implemented for the iterative pose estimation is a robust solution which enables the relaxation of the threshold of similarity set to detect correspondences. Hence, various images (and apparently even a synthetic image) can be used as reference.

7.2.3 Learning location priors

C. Val. 1: An independent validation set is used to verify the pertinence of the method, the chosen indicators and the resulting map.

- ✓ **Method:** We developed a method to learn the attractiveness (or repulsiveness) of a location described by geographic features. This method is not biased by the spatial proximity of the points of interest.
- ✓ **Geographic indicators and Map:** According to our hypotheses, we computed geographic indicators related to accessibility and morphology. We showed statistics demonstrating that the distribution of unlabelled locations and picture locations with respect to geographic features are dissimilar. With the best settings, 89% of the independent validation set was correctly classified whereas only 21% of the unlabelled locations were classified as picture locations.

Contributions: In this chapter, we developed a method to measure the photographic attractiveness of a location. Unlike, general method proposed to assess and map the distribution of pictures, we did not perform a spatial density (showing only clusters around points of interest), but a density in the space of geographic features. Hence, a value taking into account the landcover and landscape morphology is computed in every location of a map at local scale.

To sum up, in this thesis, we propose an integrated approach for the georeferencing of landscape images. Each contribution can be considered as a method to improve the monoplottter:

- by facilitating the acquisition of GCP,
- by providing linear features as GCP (currently only skyline),
- by automatically aligning an image with a 3D landscape model,
- by considering images already georeferenced in the database and images shared on the web as the reference for the pose estimation,
- by considering images already georeferenced in the database to learn trends about the preferred shooting locations.

7.3 Future research and applications

7.3.1 Further research

This work only starts filling the gap left by the matching of real images with 3D synthetic images generated from a DEM textured with an orthoimage. The continuation of this topic could be the validation of our methods with more challenging images, for example having seasonal and temporal variations. We would also suggest using these methods to retrieve the location of the picture by testing multiple panoramas generated for potential locations of the picture. The final goal is then to find an appropriate descriptor and matching strategy, which can be used to retrieve the pose of an image within a region (i.e. the pose of an image having a loose prior such as a toponym). In this context, we are inspired by two approaches. The first is proposed by Baatz et al. (2012a) and describes a method using local skyline features to retrieve the orientation of a picture at large scale (entire Switzerland).

The second is presented in Aubry et al. (2014). This method matches images of different kinds (drawing and paintings) with the 3D rendered model of a city.

Although the skyline matching would benefit from the implementation of a sky detector, the two methods proposed to determine the 3DOF orientation of a picture are already effective. Hence, in the context of the georeferencing of a collection of images one after the other (Section 5.3), the orientation of the initial reference images (i.e. the GPS images having a known location in our case study) could be entirely automatized either with HOG or skyline matching (either separately or jointly). The current limitation of our workflow for the pose estimation of new images is the initial azimuth guess. This step could also gain from the HOG matching (less sensitive than SIFT to varying image quality) or from a further development of our geometrically constrained matching to test multiple azimuths (as in Moreno-Noguer et al. (2008); Serradell et al. (2010)) from which our scheme is inspired).

The matching of remarkable high-level geographic objects detected in the image (roads, peaks, rivers . . . , labelled or digitized in an image by an operator) with those contained in GIS databases (Koperski et al., 2013; Hammoud et al., 2013) is a field of study that could benefit from the knowledge of GIS scientists. Such a function would be valuable for the supervised georeferencing of images, for instance prior to the use of the monoplottter. Specifically, for the georeferencing of an image with loose prior and located in a region unknown to the user, the labelling and digitization of geographic objects could suggest potential poses of a picture and reduce the time spent browsing a virtual globe. Moreover, this research area is complementary to current research in remote sensing and computer vision working towards the recognition of these high-level objects.

Finally, the mapping of the attractiveness was a side product of the thesis. However, a deeper literature review shows us that our approach to avoid the bias engendered by the spatial clustering of pictures (involving geographic features covering homogeneously the space) is pertinent and close to other works (Tsung-Yi et al., 2013; Lee et al., 2015). Inspired by these studies, we would also insert visual features in the process. In this way, we will compute a map of the potential locations of a picture (rather than a map of the potential locations of all pictures). Our study is one of the first taking advantages of the accuracy of the GPS-based localisation of the images shared on the web. With the rise of pictures shot with smartphones and tablets and methods to orient pictures, such as developed in this thesis, we will soon also be able to learn trends about image orientations and image contents (relation between the visual feature of the picture and corresponding locations on the geographic models). For instance, we could learn what makes a scene photogenic and thus derive a map of the most photogenic places or propose customised locations to photographers regarding their preferences.

7.3.2 Applications

In the short term, we will consolidate the Pic2Map plugin and maintain it. Next, we plan to integrate the presented tools in this plugin or, even better, in a web implementation of this monoplottter. Thus, we could carry our methods to the next stage, which is their interaction and also the interaction with the user. In addition, a web implementation will address needs such as the crowdsourcing of the georeferencing of image archives.

If it is further developed, the georeferencing of web-shared collections would be an upgrade of the "*photo tours*", offered by Google Map (a browsable group of images oriented with SfM). Indeed, this browser of pictures is currently only roughly georeferenced and can consequently not be augmented with contextual geographic data or inserted in a 3D map. With our approach, every *tour* would be

inserted in a geographic coordinates system and thus the visitor could navigate the entire collection and the entire space in a single 3D browser.

7.4 Concluding words

Georeferencing is a hot topic in computer vision and robotic vision. The accurate offline georeferencing of collection of images acquired simultaneously or having a similar appearance was widely studied (Strecha et al., 2010; Agarwal et al., 2011). Currently, many applications are available for researchers, professionals and amateurs to generate 3D models from pictures. Emerging from the robotic and the need of a UAV to map his environment to chose the best flight plan, the online mapping from a single camera is also well underway (Pizzoli et al., 2014). However, in general, the goal of these methods is the mapping and the generation of 3D models and the georeferencing is a side product.

Hence, there are situations, for which these mapping techniques do not directly provide a solution. First, there is the issue of markerless augmented reality, that is the orientation of a mobile computer with respect to geographic data. In this context, the orientation sensors currently mounted on customer cameras or tablets are not accurate enough to provide an exact alignment with geographic information. The refinement of the orientation can be obtained by the alignment of the visual content of the picture with a geographic reference. The matching with a landscape model or with georeferenced key-points is a promising method to achieve a real-time alignment.

Second, considering the offline georeferencing of a collection of images, the usual workflow relies on the detection of tie-points across the images. The tie-point matching becomes suboptimal when the images are acquired non-synchronously. Indeed, during a day, the weather conditions, the sun elevation and the shadows may induce important appearance variations. Throughout the year, the textures and colors of the landscape vary and at a longer time scale, the land cover is dynamic (forest, glacier, rivers, built areas evolve). Moreover, the quality of the images recorded by cameras has also dramatically evolved. These variations of the landscape appearance challenge the detection of tie-points across photographs (and it is not to mention drawings and paintings). Finally, the landscape is heterogeneously covered by photographs, some areas are entirely covered while others have never been recorded. The overlapping rate is therefore very variable and SfM-based georeferencing works best when the redundancy is high. Finally, after the relative orientation, the absolute orientation requires geotagged images or GCP to scale and orient the model to fit with a world coordinate system.

From these considerations, one can understand the pertinence of considering a landscape model in georeferencing methods:

- Landscape models cover the entire earth.
- A DEM is the only reference data, which can be assumed to be stable over time (except in particular types of land cover such as landslides and glaciers).
- Matches detected with a landscape model can be inserted in SfM to reduce drifts. They are also exploitable where there is no overlap between pictures and can thus link independent clusters of images.
- Finally, when developing a method to match a picture with a synthetic 3D image, we have to be robust to appearance variations induced by the rendering. Thus, the registration should rely on the matching of landscape landmarks (morphological and land cover) less impacted by

landscape appearance than local image features (key-points). Such method could provide a solution for images left out by SfM.

Specifically, a current issue is the georeferencing of picture collections such as web-shared pictures and archives. As GIS scientists, we can provide tools harnessing the wealth of photograph collections. In this scenario, we are interested in the georeferencing rather than the 3D reconstruction and consequently, we want methods that limit the number of images not georeferenced. For this task, we should not be reluctant to involve the users. For instance, recent experiences show that the crowd takes interests in the georeferencing of cultural properties (Kowal and Pridal, 2012). A key issue is to provide to the volunteers a georeferencing platform adapted to their skills (Haklay, 2013). The registration with a landscape model is certainly a mean to reach this goal by easing and automatizing some georeferencing steps. The georeferencing by volunteers is also a way to disseminate the cultural and environmental value of photographs. Indeed, archivists, geographers and environmental scientists are aware of the value of photograph collections as witnesses of the short-term and medium-term past and are the main users of georeferencing tools. However, we can also share these snapshots of the past time with the general public. Current media offers unlimited possibilities to develop platforms for personal and intuitive experiences. We can imagine virtual visits of photograph collections through space and time in a virtual globe or even provide *in situ* windows open onto the past with augmented reality.

For such applications, the georeferencing is currently the bottleneck. This challenging task can only be achieved by integrating all the knowledge available. This thesis is a first step in this direction. It incorporates user knowledge and geographic data to design georeferencing methods available for a wide variety of images and locations with a limited user supervision.

Bibliography

- Y.I. Abdel-Aziz and H.M. Karara. Direct linear transformation from comparator coordinates in close-range photogrammetry. In *ASP Symposium on Close-Range Photogrammetry*, 1971.
- S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.
- S. Ardeshir, A. R. Zamir, A. Torroella, and M. Shah. GIS-assisted object detection and geospatial localization. In *ECCV*, pages 602–617. Springer, 2014.
- M. Aubry, B. C Russell, and J. Sivic. Painting-to-3d model alignment via discriminative visual elements. *ACM Transactions on Graphics (TOG)*, 33(2):14, 2014.
- G. Baatz, O. Saurer, K. Köser, and M. Pollefeys. Large scale visual geo-localization of images in mountainous terrain. In *ECCV*, pages 517–530. Springer, 2012a.
- G. Baatz, O. Saurer, K. Koser, and M. Pollefeys. Leveraging topographic maps for image to terrain alignment. In *Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, pages 487–492. IEEE, 2012b.
- L. Baboud, M. Cadík, E. Eisemann, and H.P. Seidel. Automatic photo-to-terrain alignment for the annotation of mountain pictures. In *CVPR*, 2011.
- S. Bae, A. Agarwala, and F. Durand. Computational rephotography. *ACM Transactions on Graphics*, 29(3):24, 2010.
- M. Bansal, K. Daniilidis, and H. Sawhney. Ultra-wide baseline facade matching for geo-localization. In *ECCV*, pages 175–186. Springer, 2012.
- R. Behringer. Registration for outdoor augmented reality applications using computer vision techniques and hybrid sensors. In *Virtual Reality*, pages 244–251. IEEE, 1999.
- D.J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, 1994.
- A. Björk, K. Kjaer, N. Korsgaard, and S. Khan. An aerial view of 80 years of climate-related glacier fluctuations in southeast Greenland. *Nature Geoscience*, 5:427–432, 2012.
- N. S. Boroujeni, S. A. Etemad, and A. Whitehead. Robust horizon detection using segmentation for uav applications. In *Computer and Robot Vision (CRV)*, pages 346–352. IEEE, 2012.

Bibliography

- C. Bozzini, M. Conedera, and P. Krebs. A new monoplottting tool to extract georeferenced vector data and orthorectified raster data from oblique non-metric photographs. *International Journal of Heritage in the Digital Era*, 1(3):499–518, 2012.
- C. Bozzini, M. Conedera, and P. Krebs. A new tool for facilitating the retrieval and recording of the place name cultural heritage. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1(2):115–118, 2013.
- G. Bradski. The opencv library. *Dr. Dobb's Journal of Software Tools*, 2000.
- T. Cham, A. Ciptadi, W. Tan, M. Pham, and L. Chia. Estimating camera pose from a single urban ground-view omnidirectional image and a 2D building outline map. In *CVPR*, pages 366–373. IEEE, 2010.
- Pl Chippendale, M. Zanin, and C. Andreatta. Spatial and temporal attractiveness analysis through geo-referenced photo alignment. In *Geoscience and Remote Sensing Symposium, IGARSS*, volume 2, pages II–1116. IEEE, 2008.
- E. Church. Analytical computations in aerial photogrammetry. 1936.
- V. Coors, T. Huch, and U. Kretschmer. Matching buildings: Pose estimation in an urban environment. In *International Symposium on Augmented Reality (ISAR)*, pages 89–92. IEEE, 2000.
- J. G. Corripio. Snow surface albedo estimation using terrestrial photography. *International Journal of Remote Sensing*, 25(24):5705–5729, December 2004.
- F. Cozman and E. Krotkov. Position estimation from outdoor visual landmarks for teleoperation of lunar rovers. In *Workshop on Applications of Computer Vision, WACV*, pages 156–161. IEEE, 1996.
- D. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the world's photos. In *Conference on World Wide Web*, pages 761–770. ACM, 2009.
- D. Cremers, T. Kohlberger, and C. Schnörr. Shape statistics in kernel space for variational image segmentation. *Pattern Recognition*, 36(9):1929–1943, September 2003.
- B. Crouzy, R. Forclaz, B. Sovilla, J. Corripio, and P. Perona. Quantifying snowfall and avalanche release synchronization: A case study. *Journal of Geophysical Research: Earth Surface*, 120(2):183–199, 2015. ISSN 2169-9011.
- N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893. IEEE, 2005.
- M. Dawood, C. Cappelle, M. El Najjar, M. Khalil, and D. Pomorski. Vehicle geo-localization based on IMM-UKF data fusion using a GPS receiver, a video camera and a 3D city model. In *Intelligent Vehicles Symposium (IV)*, pages 510–515. IEEE, 2011.
- M. Dudík, S. Phillips, and R. Schapire. Correcting sample selection bias in maximum entropy density estimation. In *Advances in neural information processing systems*, pages 323–330, 2005.
- C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In *International Conference on Knowledge Discovery and Data Mining*, 2008.
- M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

- M. Fluehler, J. Niederöst, and D. Akca. *Development of an educational software system for the digital monoplottting*. ETH, Eidgenössische Technische Hochschule Zürich, Institute of Geodesy and Photogrammetry, 2005.
- W. Förstner. A feature based correspondence algorithm for image matching. *International Archives of Photogrammetry and Remote Sensing*, 26(3):150–166, 1986.
- W. Förstner. Computer vision and remote sensing - lessons learned. In *Photogrammetric Week 2009*, pages 241–249, 2009.
- F. Girardin, F. Calabrese, F. Fiore, C. Ratti, and J. Blat. Digital footprinting: Uncovering tourists with user-generated content. *Pervasive Computing*, 7(4):36–43, 2008.
- C. Grothe and J. Schaab. Automated footprint generation from geotags with kernel density estimation and support vector machines. *Spatial Cognition & Computation*, 9(3):195–211, 2009.
- A. Gruen and T. Huang. *Calibration and Orientation of Cameras in Computer Vision*. Springer Science & Business Media, 2001.
- Q. Guo, W. Li, Y. Liu, and D. Tong. Predicting potential distributions of geographic events using one-class data: concepts and methods. *International Journal of Geographical Information Science*, 25(10):1697–1715, 2011.
- N. Haala and J. Böhm. A multi-sensor system for positioning in urban environments. *Journal of Photogrammetry and Remote Sensing, ISPRS*, 58(1-2):31–42, 2003.
- M. Haklay. Citizen science and volunteered geographic information: Overview and typology of participation. In *Crowdsourcing Geographic Knowledge*, pages 105–122. Springer, 2013.
- R. Hammoud, S. Kuzdeba, B. Berard, V. Tom, R. Ivey, R. Bostwick, J. HandUber, L. Vinciguerra, N. Shnidman, and B. Smiley. Overhead-based image and video geo-localization framework. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 320–327. IEEE, 2013.
- C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.
- R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.
- J. Hays and A. Efros. Im2gps: estimating geographic information from a single image. In *CVPR*, 2008.
- H. Hoffmann. Kernel PCA for novelty detection. *Pattern Recognition*, 40(3):863–874, 2007.
- E. Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.
- R. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.
- K. Koeser, B. Bartczak, and R. Koch. An analysis-by-synthesis camera tracking approach based on free-form surfaces. In *Pattern Recognition*, pages 122–131. Springer, 2007.
- K. Koperski, C. Tusk, K. Johnson, and G. Marchisio. Fusion of GIS and remote sensing data for geolocation of photographs. In *Geoscience and Remote Sensing Symposium, IGARSS. IEEE*, 2013.

Bibliography

- K. Kowal and P. Pridal. Online georeferencing for libraries: The british library implementation of georeferencer for spatial metadata enhancement and public engagement. *Journal of Map & Geography Libraries*, 8(3):276–289, 2012.
- K. Kraus. *Photogrammetry: geometry from images and laser scans*. Walter de Gruyter, 2007.
- J. Lagrange. *Leçons élémentaires sur les Mathématiques, données à l'École normale, en 1795*. 1812.
- S. Lee, H. Zhang, D. Crandall, and I. N. Bloomington. Predicting geo-informative attributes in large-scale image collections using convolutional neural networks. In *Winter Conference on Applications of Computer Vision*. IEEE, 2015.
- V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An accurate o(n) solution to the PnP problem. *International Journal of Computer Vision*, 81(2):155–166, 2009.
- A. Li, V. Morariu, and L. Davis. Planar structure matching under projective uncertainty for geolocation. In *ECCV*, pages 265–280. Springer, 2014.
- L. Li and M. Goodchild. Constructing places from spatial footprints. In *International Workshop on Crowdsourced and Volunteered Geographic Information*, pages 15–21. ACM, 2012.
- Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide pose estimation using 3D point clouds. In *ECCV*. 2012.
- D. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- B. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.
- J. Luo, D. Joshi, J. Yu, and A. Gallagher. Geotagging in multimedia and computer vision—a survey. *Multimedia Tools and Applications*, 51(1):187–211, 2011.
- D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *CVPR*, pages 1–8. IEEE, 2007.
- T. Melen. Extracting physical camera parameters from the 3 x 3 direct linear transformation matrix. In *Optical 3D Measurement Techniques II: Applications in Inspection, Quality Control, and Robotics*, pages 355–365. International Society for Optics and Photonics, 1994.
- A. Messerli and A. Grinsted. Image GeoRectification and feature tracking toolbox: ImGRAFT. *Geoscientific Instrumentation, Methods and Data Systems Discussions*, 4(2):491–513, 2014.
- E.M. Mikhail, J.S. Bethel, and J.C. McGlone. *Introduction to modern photogrammetry*, volume 31. Wiley New York, 2001.
- K. Mikolajczyk, B. Leibe, and B. Schiele. Local features for object class recognition. In *ICCV*, volume 2, pages 1792–1799. IEEE, 2005.
- G. Milani. Pic2map: intégration de photographies dans qgis. Master's thesis, EPFL, 2014.
- G. Monge. *Géométrie descriptive. Leçons données aux écoles normales, l'an 3 de la République; par Gaspard Monge,...* Baudouin, imprimeur du corps législatif et de l'institut national, 1798.

- F. Moreno-Noguer, V. Lepetit, and P. Fua. Pose priors for simultaneously solving alignment and correspondence. In *ECCV*, pages 405–418. Springer, 2008.
- A. Munoz and J. Moguerza. One-class support vector machines and density estimation: the precise relation. In *Progress in Pattern Recognition, Image Analysis and Applications*, pages 216–223. Springer, 2004.
- P. Naval Jr, M. Mukunoki, M. Minoh, and K. Ikeda. Estimating camera position and orientation from geographical map and mountain image. In *38th Research Meeting of the Pattern Sensing Group, Society of Instrument and Control Engineers*. Citeseer, 1997.
- E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.
- M. Pizzoli, C. Forster, and D. Scaramuzza. Remode: Probabilistic, monocular dense reconstruction in real time. In *Robotics and Automation (ICRA)*, pages 2609–2616. IEEE, 2014.
- A. Popescu and G. Grefenstette. Deducing trip related information from flickr. In *International conference on World Wide Web*, pages 1183–1184. ACM, 2009.
- T. Produit and D. Tuia. An open tool to register landscape oblique images and generate their synthetic models. In *Open Source Geospatial Research and Education Symposium (OGRS)*, 2012.
- T. Produit, D. Tuia, F. Golay, and C. Strecha. Pose estimation of landscape images using DEM and orthophotos. In *Int. Conference on Computer Vision in Remote Sensing, CVRS*, 2012.
- T. Produit, D. Tuia, F. De Morsier, and F. Golay. Picture density in the space of geographic features: mapping landscape attractiveness. In *International Workshop on Environmental Multimedia Retrieval (in conjunction with ACM ICMR)*, 2014a.
- T. Produit, D. Tuia, V. Lepetit, and F. Golay. Pose estimation of web-shared landscape pictures. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1:127–134, 2014b.
- G. Reitmayr and T. W. Drummond. Going out: robust model-based tracking for outdoor augmented reality. In *International Symposium on Mixed and Augmented Reality, ISMAR*, pages 109–118. IEEE, 2006.
- F. Remondino. Detectors and descriptors for photogrammetric applications. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(3):49–54, 2006.
- F. Remondino, S. Del Pizzo, T. Kersten, and S. Troisi. Low-cost and open-source solutions for automated image orientation : A critical overview. In Marinos Ioannides, Dieter Fritsch, Johanna Leissner, Rob Davies, Fabio Remondino, and Rossella Caffo, editors, *Progress in Cultural Heritage Preservation*, volume 7616 of *Lecture Notes in Computer Science*, pages 40–54. Springer, 2012.
- M. A. Ruzon and C. Tomasi. Edge, junction, and corner detection using color distributions. *Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1281–1295, 2001.
- T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2D-to-3D matching. In *ICCV*, pages 667–674. IEEE, 2011.
- T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, volume 6, page 7, 2012.

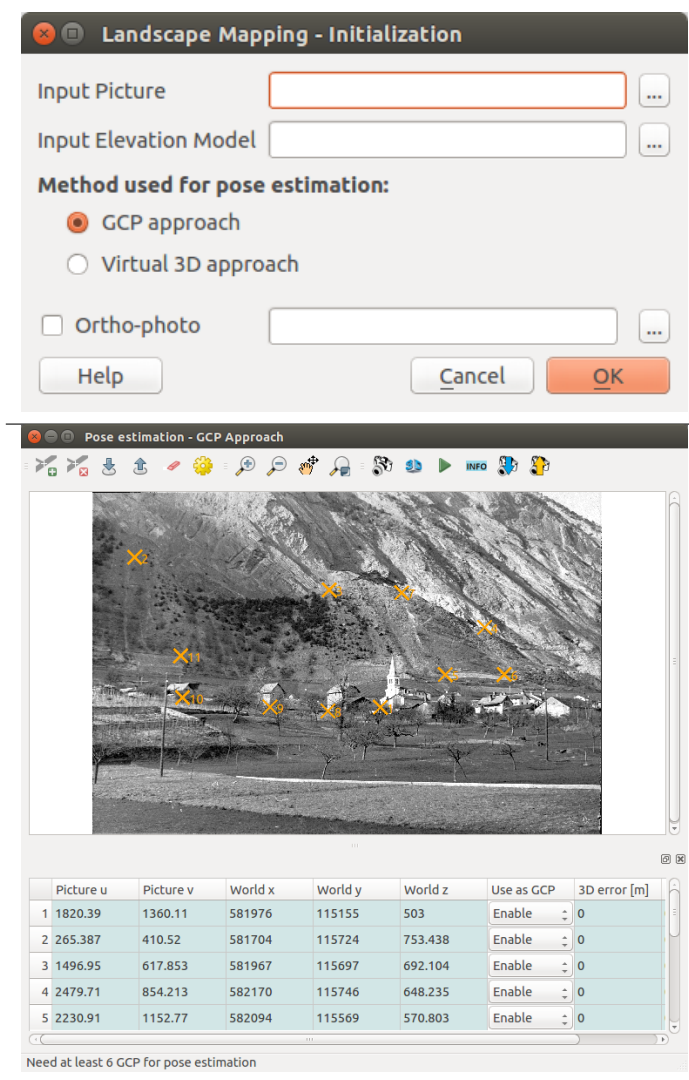
Bibliography

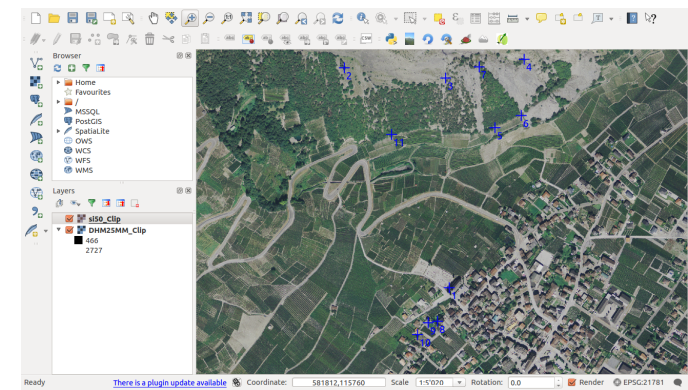
- C. Scapozza, C. Lambiel, C. Bozzini, S. Mari, and M. Conedera. Assessing the rock glacier kinematics on three different timescales: a case study from the southern swiss alps. *Earth Surface Processes and Landforms*, 2014.
- G. Schindler, P. Krishnamurthy, R. Lubliner, Y. Liu, and F. Dellaert. Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In *CVPR*, pages 1–7. IEEE, 2008.
- C. Schlieder and C. Matyas. Photographing a city: An analysis of place concepts based on spatial choices. *Spatial Cognition & Computation*, 9(3):212–228, 2009.
- B. Schölkopf and A. Smola. *Learning with Kernels*. MIT press, Cambridge (MA), 2002.
- B. Schölkopf, R.C. Williamson, A.J. Smola, J. Shawe-Taylor, and J.C. Platt. Support vector method for novelty detection. In *NIPS*, volume 12, pages 582–588, 1999.
- B. Schölkopf, A. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.
- M. Schumann, S. Achilles, S. Müller, and Arbeitsgruppe Computergraphik. Analysis by synthesis techniques for markerless tracking. In *6th Workshop on Virtual and Augmented Reality, GI Workgroup VR/AR*, 2009.
- S. Scott and W. David. On optimal and data-based histograms. *Biometrika*, 66(3):605–610, 1979.
- E. Serradell, M. Özuysal, V. Lepetit, P. Fua, and F. Moreno-Noguer. Combining geometric and appearance priors for robust homography estimation. In *ECCV*, pages 58–72. Springer, 2010.
- J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition*, pages 593–600. IEEE, 1994.
- A. Shrivastava, T. Malisiewicz, A. Gupta, and A. Efros. Data-driven visual similarity for cross-domain image matching. page 1. ACM Press, 2011.
- J. Sivic and A. Zisserman. Efficient visual search for objects in videos. *Proceedings of the IEEE*, 96(4):548–566, 2008.
- N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. *ACM transactions on graphics (TOG)*, 25(3):835–846, 2006.
- G. Sourimant, T. Collet, V. Jantet, L. Morin, and K. Bouatouch. Toward automatic GIS–video initial registration: And application to buildings textures computation. *annals of telecommunications - annales des télécommunications*, 67(1-2):1–13, 2012.
- F. Stein and G. Medioni. Map-based localization using the panoramic horizon. In *International Conference on Robotics and Automation*, pages 2631–2637. IEEE, 1992.
- C. Strecha, T. Pylvanainen, and P. Fua. Dynamic and scalable large scale image reconstruction. In *CVPR*, pages 406–413. IEEE, 2010.
- Y. Sun, H. Fan, M. Bakillah, and A. Zipf. Road-based travel recommendation using geo-tagged images. *Computers, Environment and Urban Systems*, 2013.
- L. Svärm, O. Enqvist, M. Oskarsson, and F. Kahl. Accurate localization and pose estimation for large 3d models. *CVPR*, pages 532–539, 2014.

- R. Szeliski. *Computer vision: algorithms and applications*. Springer, 2010.
- R. Talluri and J. Aggarwal. Position estimation for an autonomous mobile robot in an outdoor environment. *Robotics and Automation, IEEE Transactions on*, 8(5):573–584, 1992.
- C. Tardivo. Topofotografia aerea. *International Archives of Photogrammetry*, 4:14, 1913.
- D. Tax and R. Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004.
- L. Tsung-Yi, S. Belongie, and J. Hays. Cross-view image geolocalization. In *CVPR*, pages 891–898, 2013.
- T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trend in Computer Graphics and Vision*, 3(3):177–280, 2007.
- R.. Webb. *Repeat Photography: Methods and Applications in the Natural Sciences*. Island Press, 2010.
- T. Wider, D. Palacio, and R. Purves. Georeferencing images using tags: application with flickr. In *AGILE International Conference on Geographic Information Science*, 2013.
- S. Wiesmann, L. Steiner, M. Pozzi, C. Bozzini, A. Bauder, and L. Hurni. Reconstructing historic glacier states based on terrestrial oblique photographs. *Proceedings–AutoCarto*, pages 16–18, 2012.
- J. Woo, K. Son, T. Li, G. Kim, and I. Kweon. Vision-based UAV navigation in mountain area. In *MVA*, pages 236–239, 2007.
- L. Xie and S. Newsam. Im2map: Deriving maps from georeferenced community contributed photo collections. In *International Workshop on Social Media*, pages 29–34. ACM, 2011.
- D. Zielstra and H. Hochmair. Positional accuracy analysis of flickr and panoramio images for selected world regions. *Journal of Spatial Science*, 58(2):251–273, 2013.
- D. Ziou, S. Tabbone, et al. Edge detection techniques-an overview. *Pattern Recognition and Image Analysis*, 8:537–559, 1998.

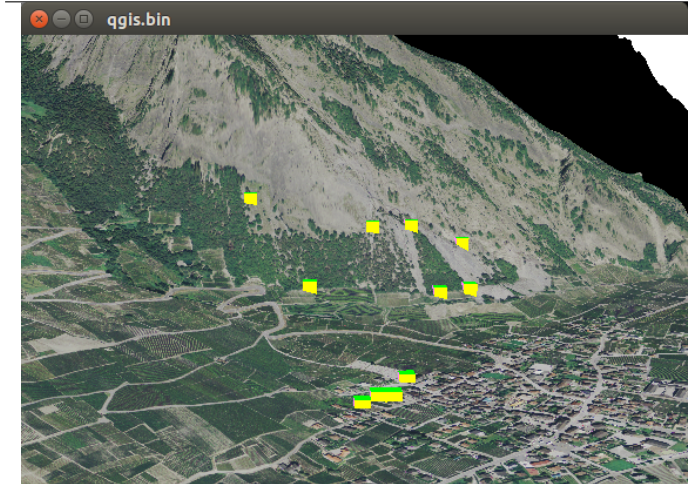
Appendix

A Pic2map snapshots

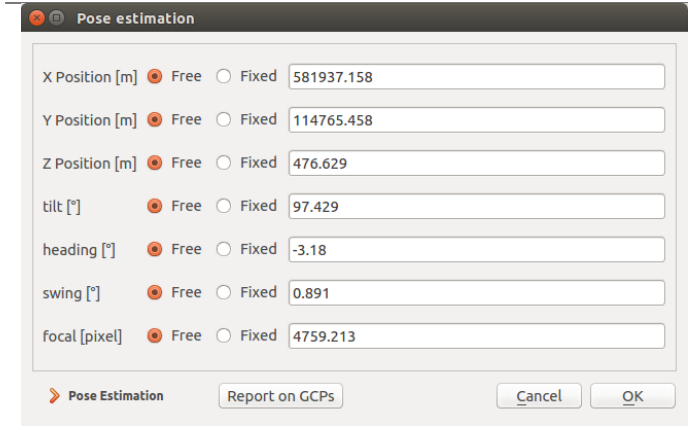




Digitalization of the GCP: The corresponding locations are clicked in the GIS.

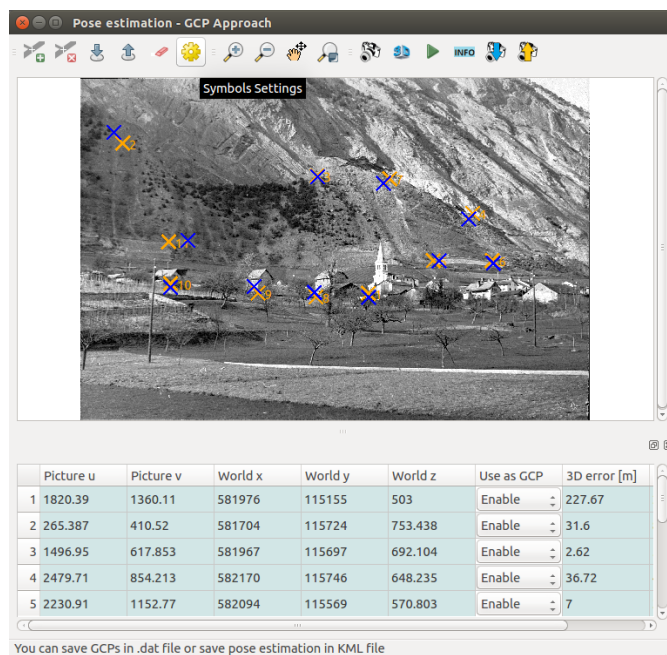


Digitalization of the GCP: Alternatively, they can also be clicked in the 3D view.

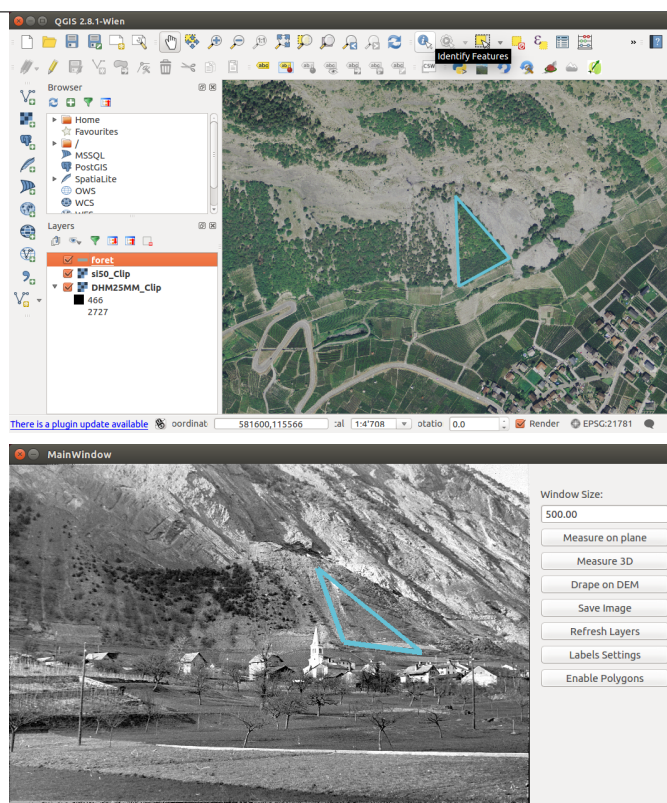


Pose estimation: Known parameters can be fixed.

A. Pic2map snapshots



Pose estimation: the 3D control points are projected in the image. The differences are listed in the table on the bottom of the photograph.



Monoplotting: An object drawn in the GIS is projected in the picture (and inversely). The ortho-rectified picture is generated with the button "Drape on DEM"

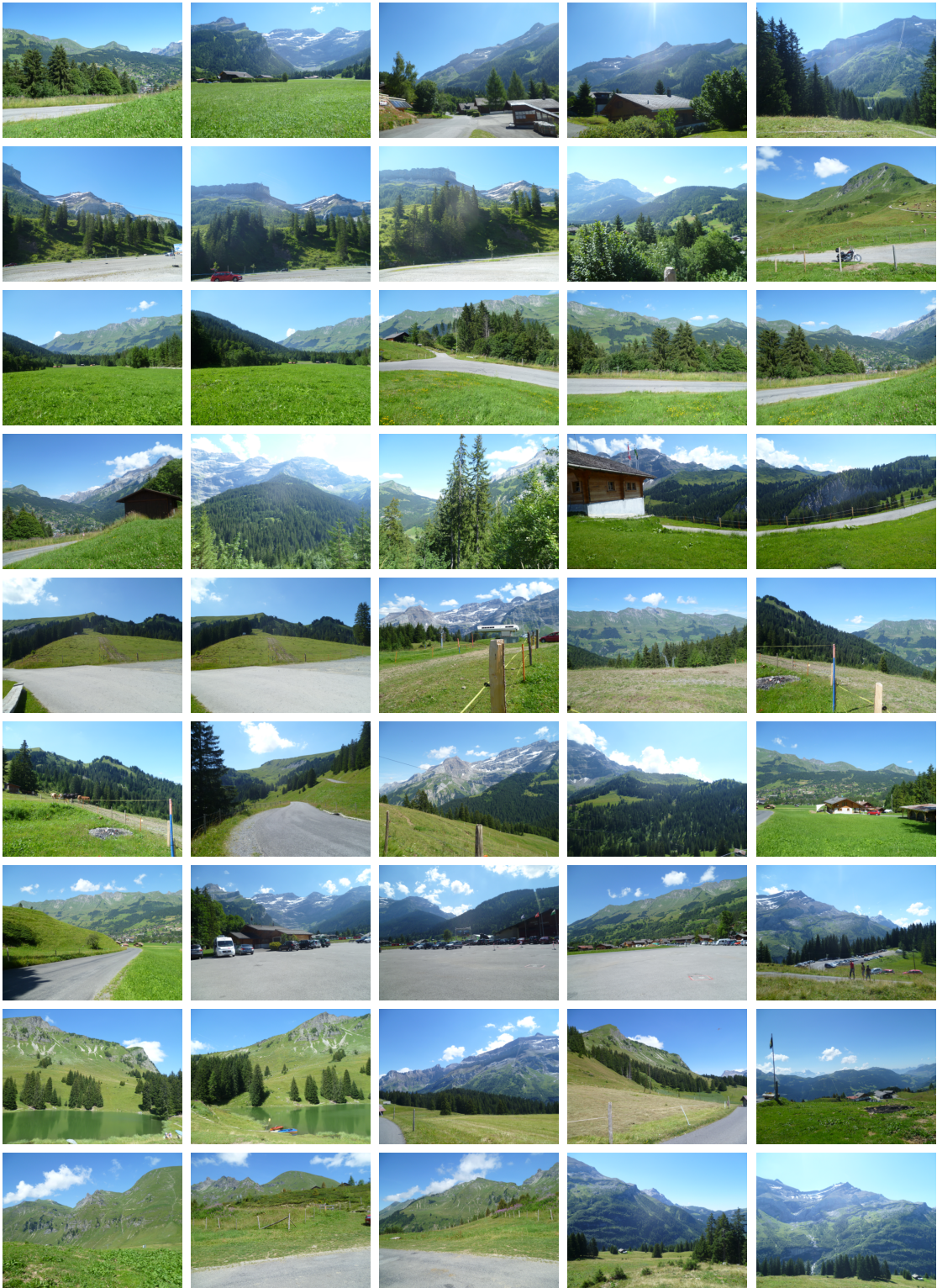
B Les Diablerets dataset



B. Les Diablerets dataset



Appendix



C Skyline matching pseudo-codes

C.1 3DOF orientation

Data: Query image 2D skyline features = u_Q , Location T , threshold for the limit of the roll = t , number of azimuth to test = n , DEM, Orthoimage

Result: query image pose = \mathbf{r} in \mathbb{R}^3

```

 $u_R, X_R \leftarrow \text{Panorama}(T, \text{DEM}, \text{Orthoimage});$                                 /* Generate a 360 Panorama */
for  $i \leftarrow 0$  to  $n$  do;                                                    /* Loop to test azimuths */

    azimuth[i]  $\leftarrow i * 360 / n$ ;
     $u_R^s, X_R^s \leftarrow \text{Subset}(\text{azimuth}, u_R, X_R);$                         /* Extract piece of skyline */
    for  $j \leftarrow 1$  to 5 do
        if  $j=1$  then
             $u_Q, u_R^s \leftarrow \text{AlignPrincipalDirection}(u_Q, u_R^s);$ 
        end
        if  $(j=1)$  or  $(j=5)$  then
             $\text{Cost}() \leftarrow \text{EuclideanXY}();$ 
        else
             $\text{Cost}() \leftarrow \text{EuclideanY}();$ 
        end
        error, correspondences  $\leftarrow \text{DtwMatch}(u_Q, u_R^s, \text{Cost}());$ 
         $p \leftarrow \text{OrientationThreeDOF}(\text{correspondences});$                     /* Compute camera angles */
         $u_R^s = \text{Projection}(p, X_R^s);$                                           /* Re-compute image coordinates of the skyline */
    end
    Error[i], correspondences  $\leftarrow \text{DtwMatch}(u_Q, u_R^s, \text{Cost}());$ 
     $R[i] \leftarrow \text{OrientationThreeDOF}(\text{correspondences});$ 
end
Error,  $R \leftarrow \text{ThresholdSwing}(\text{Error}, R, t);$                             /* Discard impossible swing values */
 $\mathbf{r} \leftarrow \text{Min}(\text{Error}, R);$                                               /* Detect best alignment and best pose */

```

C.2 6DOF orientation

Data: Query image 2D skyline features = u_Q , Location prior T^0 , Azimuth prior = α^0 , threshold, DEM, Orthoimage

Result: query image pose = \mathbf{p} in \mathbb{R}^6

```

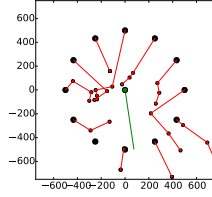
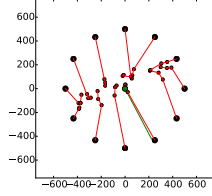
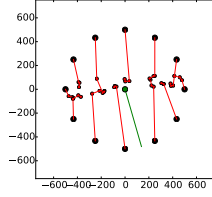
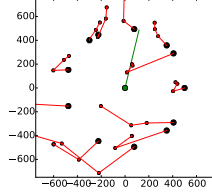
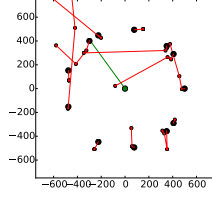
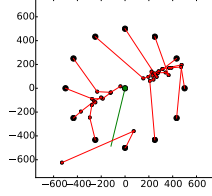
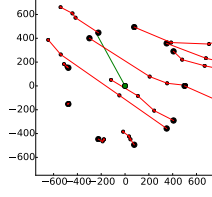
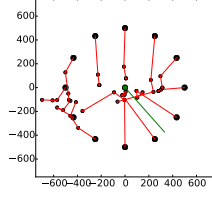
 $u_R, X_R \leftarrow \text{Panorama}(T^0, \alpha^0, \text{DEM}, \text{Orthoimage});$            /* Generate a slice of Panorama */
correspondences,  $r, u_R^s, X_R^s \leftarrow \text{DTWThreeDOF}(u_Q, u_R, X_R);$  /* Use the 3DOF skyline matching */
 $\mathbf{p} \leftarrow \text{PosePrior}(r, T^0);$                                      /* Initial pose */
for  $i \leftarrow 0$  to  $n$  do;                                         /* kalman iterations */

     $\mathbf{p} \leftarrow \text{Kalman}(\mathbf{p}, \Sigma_p, \text{correspondences});$ 
     $\mathbf{p} \leftarrow \text{UpdateAltitude}(\mathbf{p}, \text{DEM});$                        /* Put Z on the ground level */
     $\mathbf{p} \leftarrow \text{OrientationThreeDOF}(\mathbf{p}, \text{correspondences});$       /* Update camera angles */
     $u_R^s = \text{Projection}(\mathbf{p}, X_R^s);$                                /* Re-compute skyline coordinates */
    if  $i \text{ MultipleOf } (5)$  then
         $u_R^s, X_R^s \leftarrow \text{Panorama}(\mathbf{p}, \text{DEM}, \text{Orthoimage});$  /* Generate new Panorama slice */
    end
    error, correspondences  $\leftarrow \text{DtwMatch}(u_Q, u_R^s, \text{Cost}());$  /* Get new correspondences */
end
return  $\mathbf{p};$ 

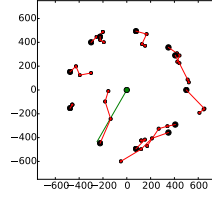
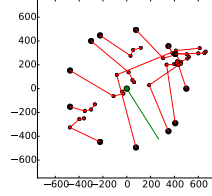
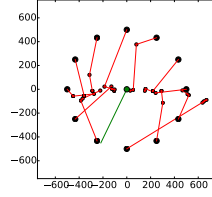
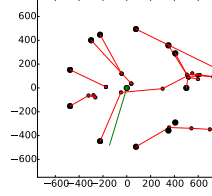
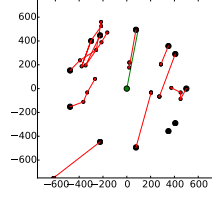
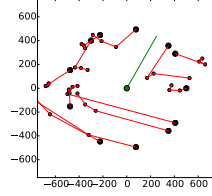
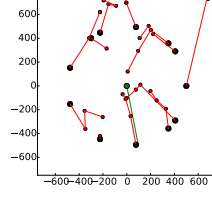
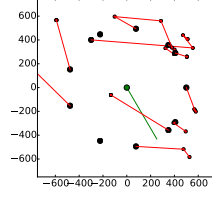
```

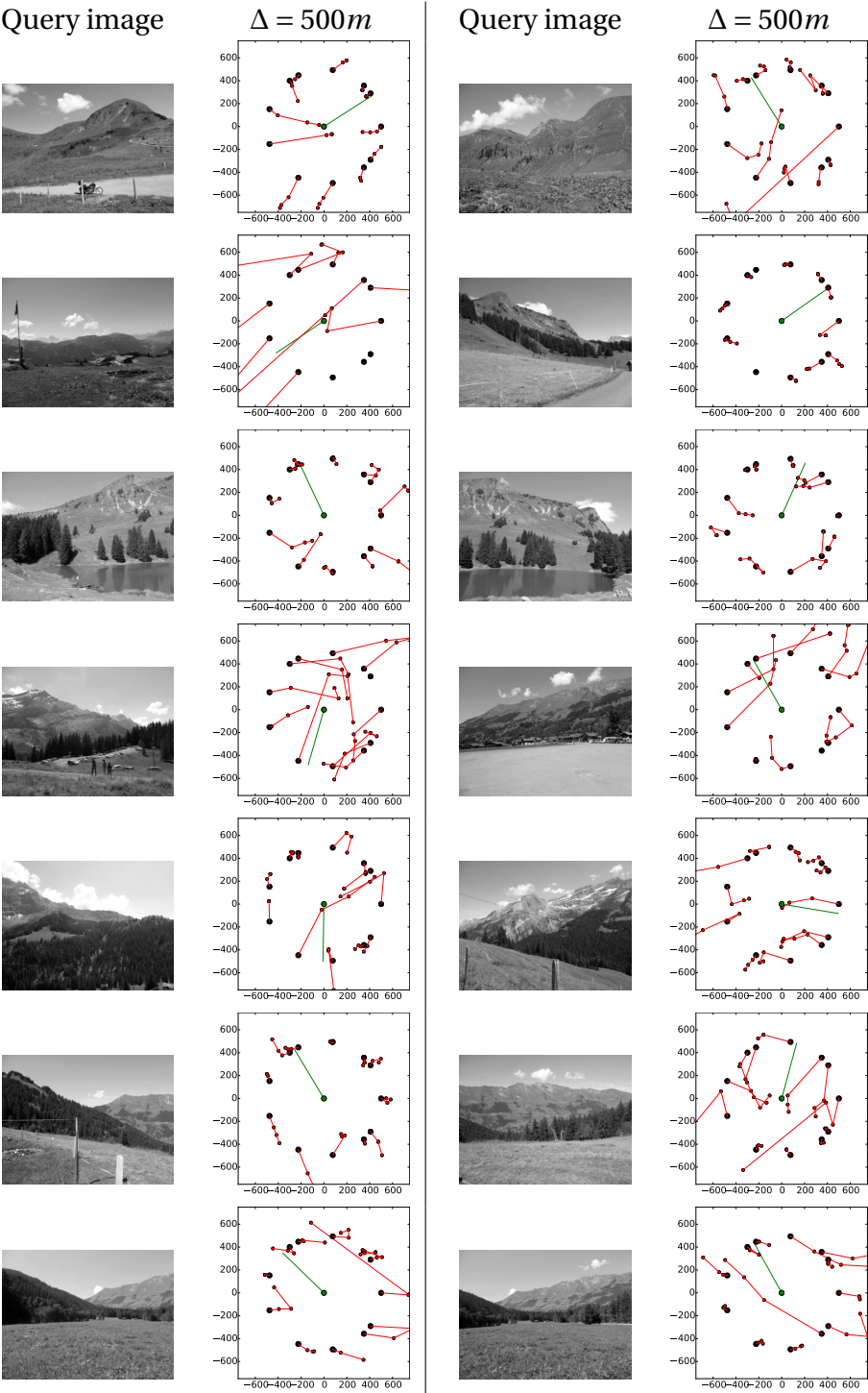
D Skyline matching results

Query image

 $\Delta = 500m$ 

Query image

 $\Delta = 500m$ 



E HOG matching pseudo-codes

E.1 Global matching

Data: Query image = Q , Location = T , DEM, Orthoimage

Result: query image pose = \mathbf{r} in \mathbb{R}^3

```

 $R_{im}, R_{XYZ} \leftarrow \text{Panorama}(T, \text{DEM}, \text{Orthoimage});$            /* Generate 360 Panorama */
 $Q^s, R^s \leftarrow \text{Scale}(R_{im}, Q);$ 
 $d^Q \leftarrow \text{HOG}(Q^s);$                                            /* Compute HOG description */
for  $i, j \leftarrow 0, 0$  to  $n, m$  do ;                               /* Test each location of the panorama */

    |  $R_e^s \leftarrow \text{PanoramaExtract}(i, j, R^s);$ 
    |  $d^R \leftarrow \text{HOG}(R_e^s);$ 
    |  $D[i, j] \leftarrow \text{DistanceCorrelation}(d^Q, d^R);$            /* Measure distance */
end
locations  $\leftarrow \text{LocalMinima}(D, \text{minimalDist}, \text{numberOfMin}=1);$  /* Get minimal distance(s) */
XYZ  $\leftarrow R_{XYZ}[\text{locations}];$                                    /* Get corresponding world coordinates */
azimuth, tilt  $\leftarrow \text{CameraDirection}(T, \text{XYZ});$                /* Compute viewing direction */
 $\mathbf{r}[1] \leftarrow \text{azimuth};$ 
 $\mathbf{r}[2] \leftarrow \text{tilt};$ 
 $\mathbf{r}[3] \leftarrow \text{roll}=0;$ 
return  $\mathbf{r};$ 

```

E.2 Multi-scale matching

Data: Query image = Q , Location = T , DEM, Orthoimage

Result: query image pose = \mathbf{r} in \mathbb{R}^3

```

 $R_{im}, R_{XYZ} \leftarrow \text{Panorama}(T, \text{DEM}, \text{Orthoimage});$                                 /* Generate 360 Panorama */
 $\text{maskD} \leftarrow \text{DistanceMask}(R_{XYZ}, 500);$                                 /* Compute a distance-based mask */
 $\text{sizes} \leftarrow [300, 275, 250, 225, 175, 150];$                                 /* Size of the patches */
 $\text{correspondences} = [];$ 
for  $s$  in  $\text{sizes}$  do
     $R_{points} \leftarrow \text{PanoramaDensePoints}(\text{mask}, s, \text{distance});$  /* Regular locations of the panorama */
     $Q_{points} \leftarrow \text{QueryDensePoints}(s, \text{distance});$  /* Regular locations of the query image */
     $Q^s, R^s \leftarrow \text{Scale}(R_{im}, Q, s);$                                 /* Scale to match HOG patch size */
     $d^R \leftarrow \text{HOG}(R_{points}, R^s);$                                 /* Description of the locations */
     $d^Q \leftarrow \text{HOG}(Q_{points}, R^s);$ 
    for  $i$  in  $\text{Size}(R_{points})$  do;                                /* Each location of the panorama... */
         $D \leftarrow \text{DistanceCorrelation}(d^Q, d^R[i]);$  /* ...compared with each location of the query image */
         $Q_{locations} \leftarrow \text{LocalMinima}(D, \text{minimalDist}, \text{minimalHogDist}, \text{numberOfMin}=3);$  /* Get n local minima in the query image */
         $R_{location} \leftarrow R_{points}[i];$ 
         $\text{correspondences} \leftarrow \text{AddCorrespondences}(\text{correspondences}, Q_{locations}, R_{location});$  /* Create three potential correspondences. */
    end
end
 $\text{geomCorrespondences} \leftarrow \text{TranslationTest}(\text{correspondences}, \text{minDist});$  /* Each potential translation is tested */
 $\mathbf{r} = \text{Ransac}(\text{geomCorrespondences}, \text{model} = \text{3DOF orientation});$  /* Ransac to compute the 3DOF orientation */
return  $\mathbf{r};$ 

```

F Georeferencing a collection of images.

Data: list of reference images = **Rims**, query image = **Qim**, **DEM**
Result: query image pose = **p**

```

/* Extract features description and location from query image */
 $d^Q, u^Q$  = ExtractSIFTfeatures (Qim);
/* Initialisation */
 $X_{all}^R = \{\}$ ;
 $d_{all}^R = \{\}$ ;
/* Extract features from reference images */
for Rim in Rims do
     $d^R, u^R, X^R$  = ExtractSIFTfeatures (Rim)/* Reference images pixels have 3D
    coordinates */
    /* Stores 3D locations and descriptions */
    add  $X^R$  to  $X_{all}^R$ ;
    add  $d^R$  to  $d_{all}^R$ ;
end
/* SIFT feature matching using Nearest Neighbor Distance Ratio */
MatchesNNDR  $\leftarrow$  SIFTNNDR ( $d^Q, d_{all}^R$ , NNDRthreshold);
/* Get a priori camera location from Geotag */
 $T_0 = (Qim.X, Qim.Y, Qim.Z)$ 
/* RANSAC to compute initial pose */
MatchesRANSAC,  $p \leftarrow$  RANSAC (MatchesNNDR. $d^Q$ , MatchesNNDR. $d_{all}^Q$ , model = orientation,  $T_0$ );
/* Kalman iterations */
SIFTthreshold  $\leftarrow$  max(EuclideanDistance (MatchesRANSAC. $d^Q$ , MatchesRANSAC. $d^R$ ));
 $\Sigma_0^P \leftarrow$  diag (1000m, 1000m, 300m,  $10^0$ ,  $10^0$ ,  $1^0$ );
/* Covariance matrix initialization */
 $u_{all}^R \leftarrow$  project ( $p, X_{all}^R$ );
 $x, \Sigma^P \leftarrow$  Kalman ( $p, \Sigma_0^P$ , MatchesRANSAC. $u^Q$ , MatchesRANSAC. $u^R$ , MatchesRANSAC. $X^R$ );
while  $n \leq nIteration$  do
     $\Sigma^u \leftarrow$  propagationVariance ( $p, u_{all}^R, X_{all}, \Sigma^P$ );
     $u_{all}^R \leftarrow$  project ( $p, X_{all}^R$ );
    ellipses  $\leftarrow$  drawEllipses ( $u_{all}^R, \Sigma^u$ , nSigma)
    ellipsesMatches  $\leftarrow$  withinEllipses ( $u_Q$ , ellipses);
    SIFTMatches  $\leftarrow$  thresholdSIFT (ellipseMatches. $d^Q$ , ellipseMatches. $d_{all}^R$ , SIFTthreshold);
     $p, MatchesRANSAC \leftarrow$  RANSAC ( $p$ , SIFTMatches. $u^Q$ , SIFTMatches. $X_{all}^R$ , model = camera orientation);
     $p, \Sigma^P \leftarrow$  Kalman ( $p, \Sigma^P$ , MatchesRANSAC. $u^Q$ , MatchesRANSAC. $u_{all}^R$ , MatchesRANSAC. $X_{all}^R$ );
end
/* Fine orientation with horizon */
 $h^R, H^R \leftarrow$  HorizonFromDEM ( $p$ , DEM)/* Get reference horizon */
 $\Sigma^u \leftarrow$  propagationVariance ( $p, h^R, H^R, \Sigma^P$ );
hMask = HorizonMask ( $h^R, \Sigma^u$ );
 $h^Q \leftarrow$  Sobel (Qim, hMask);
MatchesDTW  $\leftarrow$  DTW ( $h^Q, h^R$ );
 $p \leftarrow$  Orientation ( $p$ , MatchesDTW. $h^Q$ , MatchesDTW. $H^R$ );

```

Curriculum Vitae

Produit Timothée

30 years old

Ing. ETH.

timothee.produit@gmail.com

Grand-Rue 31a, 1814 La Tour-de-Peilz, Switzerland

www: <http://people.epfl.ch/timothee.produit> 🌐

www: <https://sites.google.com/site/produittim/home> 🌐

Education

- **EPFL** Lausanne
In progress: PhD in the GIS laboratory 2010 - present
– PhD topic: Registration of single landscape photographs with 3D landscape models
- **EPFL** Lausanne
Master of Science in MSc Environmental Sciences and Engineering 2007 - 2009
– Master Thesis: A novel GIS method to determine an urban centrality index applied to the Barcelona metropolitan area (supervisors: Dr. S. Joost, Prof. F. Golay)
– ISGO prize for an innovative Master Thesis in Geomatics
- **EPFL** Lausanne
BSc in Environmental Sciences and Engineering 2004 - 2007
- **Lycée Collège des Creusets** Sion
Federal Matura 1999-2004

Work Experience

- **EPFL** Lausanne
Eatlas du Valais: a web-based atlas for canton Wallis 🌐 August 2010 - August 2012
- **EPFL** Lausanne
Responsible for IT (computers, servers and web-sites management) 2010 - present
- **Swisstopo** Bern
Internship in the Image Data and Height Models group August 2009 - August 2010
- **EPFL** Lausanne
Teaching Assistant 2008-2015
– GIS course: first assistant
– Supervision of three master theses
– Quantitative Method: undergraduate assistant
– Topometry: undergraduate assistant

- | | |
|------------------------------|-------------|
| • Geosat SA | Sion |
| • <i>Surveyor internship</i> | Summer 2008 |
| • Stéphane Bessero SA | Fully |
| • <i>Surveyor internship</i> | Summer 2007 |
| • Nivalp SA | Grimisuat |
| • <i>Forestry internship</i> | Summer 2006 |

Academic Projects, selection of publications

- **T. Produit**, D. Tuia, V. Lepetit and F. Golay. Pose estimation of web-shared landscape pictures. *Photogrammetric computer vision (ISPRS-PCV)*, Zurich 2014
- **T. Produit**, D. Tuia, F. De Morsier and F. Golay. Picture density in the space of geographic features: mapping landscape attractiveness *International Workshop on Environmental Multimedia Retrieval (EMR)*, Glasgow 2014
- **T. Produit** and D. Tuia. An open tool to register landscape oblique images and generate their synthetic models. *Open Source Geospatial Research and Education Symposium (OGRS)*, Yverdon-les-Bains 2012
- **T. Produit**, D. Tuia, F. Golay and C. Strecha. Pose estimation of landscape images using DEM and orthophotos. *Int. Conference on Computer Vision in Remote Sensing (CVRS)*, Xiamen, 2012
- **T. Produit**, N. Lachance-Bernard, E. Strano, S. Porta and S. Joost. A modified Kernel Density Estimation applied to the Barcelona urban network. *International Conference on Computational Science and Its Applications (ICCSA)*, Fukuoka 2010
- **T. Produit**, F. Golay, N. Lachance-Bernard, E. Strano and S. Joost (Dirs.) A novel GIS method to determine an urban centrality index applied to the Barcelona metropolitan area. *EPFL Master Thesis*, 2009
- Semester Project: Recover ALS strip differences by ICP, *TOPO lab - EPFL*
- Design Project: Evaluation of Lidar acquisition method in terrestrial cinematic, *TOPO lab - EPFL, Collaboration with Helimap System SA*

Skills

- Programming and Markup Languages
 - **Expert:** Python, SQL, Matlab
 - **Intermediate:** Java, \LaTeX , HTML, PHP
- Software
 - **Expert:** ArcGIS, QGIS, SAGA GIS
 - **Intermediate:** Web-mapping (Geoserver, Openlayer etc.)
- Languages
 - **English:** B2 level of the European standard
 - **German:** B1 level of the European standard