université PARIS-SACLAY

ENS CACHAN

Earth Observation and Stereo Vision

A dissertation presented
by

Carlo de Franchis

in fulfillment of the requirements
for the degree of Doctor of Philosophy
in the subject of Applied Mathematics

Committee in charge

President	François Goulette	-	Mines ParisTech, FR
Examiners	Hélène de Boissezon	-	Centre National d'Études Spatiales (CNES), FR
	Pablo d'Angelo	-	German Aerospace Center (DLR), DE
	Jean-Marc Delvit	-	Centre National d'Études Spatiales (CNES), FR
Referees	Pascal Monasse	-	École des Ponts ParisTech, FR
	Marc Pierrot Deseilligny	-	Institut Géographique National (IGN), FR
Advisors	Gabriele Facciolo	-	École Normale Supérieure de Cachan, FR
	Jean-Michel Morel	-	École Normale Supérieure de Cachan, FR
Invited	Julien Michel	-	Centre National d'Études Spatiales (CNES), FR

Defended at École Normale Supérieure de Cachan on the

5 October 2015

Numéro National de Thèse (NNT): 2015SACLN002

This work includes references to online resources. Permanent URLs were provided when possible and the availability of these resources was verified in September 2015, but the progressive obsolescence of these references is inevitable.

revision d4b3470
produced on Friday 18th December, 2015 at 11:12 with LaTeX

Abstract

This thesis deals with the problem of computing accurate digital elevation models of the Earth's surface from optical images taken by pushbroom observation satellites. It takes advantage of the collaboration of the defendant with CNES (the French Space Agency) on the development of stereo vision tools for Pléiades, the first Earth observation satellite producing quasi simultaneous stereo pairs or triplets with small baseline.

The first chapter describes a simple pushbroom camera model for observation satellites orbiting around the Earth and addresses the correction of the acquisition geometry by involving extrinsic information. This chapter proposes a new algorithm to refine the orientation parameters from a set of ground control points, applicable to all pushbroom satellites.

With the goal of testing for satellite imaging the thriving exploration of stereo matching by the computer vision community, the second chapter explores the adaptation of the theory of epipolar rectification to pushbroom images. Epipolar rectification is traditionally used in stereo to reduce the matching computational cost, and permits to test for satellite imaging the most competitive computer vision algorithms. The next chapter discusses the effect of geometric calibration errors on the rectification accuracy and proposes a method to cancel its impact on stereo matching.

The fourth chapter describes and analyzes a detailed implementation of the Semi-Global Matching (SGM) algorithm, which is currently among the top-ranked stereo vision algorithms. Based on a recently proposed interpretation of SGM as a min-sum Belief Propagation algorithm, a variant is proposed that allows to reduce by a factor five the energy gap of SGM with respect to reference algorithms for Markov Random Fields with truncated smoothness terms.

By wrapping together the algorithmic blocks described in the previous chapters, the fifth chapter describes `S2P`, a complete stereo pipeline for producing digital elevation models from satellite images. As an application, a landscape evolution model is presented in the sixth chapter. The model is used to simulate numerically the fine structure of the river networks on digital elevation models obtained from Pléiades Earth observation images.

The source code of the `S2P` stereo pipeline[1] is distributed as open source. To ensure reproducibility, the algorithms implemented in each block of the `S2P` pipeline are submitted to the IPOL journal, with detailed descriptions and analysis, documented source codes and online demonstrations.

[1] https://github.com/carlodef/s2p

Résumé

Cette thèse étudie les problèmes posés par l'estimation automatique de modèles numériques d'élévation de la surface terrestre à partir de photographies prises par des satellites. Ce travail a bénéficié d'une collaboration avec le CNES (Centre National d'Etudes Spatiales) sur le développement d'outils de vision stéréoscopique pour Pléiades, le premier satellite d'observation de la Terre capable de produire des paires ou triplets d'images quasi-simultanées.

Le premier chapitre de la thèse décrit un modèle simplifié de caméra *pushbroom* destiné aux satellites d'observation de la Terre, et aborde le problème de la correction des données de calibration en faisant intervenir des mesures externes. Ce chapitre propose un nouvel algorithme pour affiner les paramètres d'orientation du satellite à partir d'un jeu de points de contrôle. Il est utilisable pour tous les satellites munis de caméras *pushbroom*.

Dans le but d'appliquer aux images satellitaires les nombreux algorithmes de mise en correspondance stéréoscopique développés en traitement d'images et en vision par ordinateur, le deuxième chapitre explore l'adaptation de la théorie de la rectification épipolaire aux images prises par des caméras *pushbroom*. La rectification épipolaire est utilisée habituellement pour réduire la complexité du problème de mise en correspondence stéréoscopique, et permet d'appliquer les algorithmes les plus récents à des images satellitaires. Le chapitre suivant étudie les effets des erreurs de calibration géométrique sur la rectification et propose une méthode pour éliminer leur impact sur la mise en correspondance.

Le quatrième chapitre décrit et analyse en détails une implémentation de l'algorithme *Semi-Global Matching* (SGM), classé actuellement parmi les meilleurs algorithmes de mise en correspondance stéréoscopique. En se fondant sur une réinterprétation récente de SGM, ce chapitre en propose une variante qui permet de réduire d'un facteur cinq son écart en énergie par rapport aux algorithmes de référence pour la minimisation de champs aléatoires de Markov.

En assemblant les blocs algorithmiques décrits dans les chapitres précédents, le cinquième chapitre décrit `S2P`, une chaîne stéréoscopique complète qui produit des modèles numériques d'élévation à partir d'images satellitaires. Un modèle d'évolution de paysage est présenté dans le sixième chapitre comme exemple d'application. Le modèle est utilisé pour simuler numériquement la structure fine du réseau hydrographique sur des modèles numériques d'élévation obtenus à partir d'images prises par Pléiades.

Le code source de la chaîne `S2P`[2] est distribué en tant que logiciel *open source*. Afin d'assurer la reproductibilité des résultats obtenus, les algorithmes implémentés dans `S2P` sont en cours de publication dans le journal IPOL, accompagnés de descriptions et d'analyses détaillées, de codes sources documentés et de demonstrateurs en ligne.

[2] https://github.com/carlodef/s2p

Remerciements

Mes remerciements vont naturellement en premier lieu à mes deux directeurs de thèse, Jean-Michel Morel et Gabriele Facciolo. Les rendez-vous hebdomadaires avec Jean-Michel Morel ont rythmé et structuré les recherches de ces quatre années de thèse. Ses conseils toujours justes, sa confiance et son enthousiasme m'ont permis de travailler dans des conditions scientifiques idéales. La collaboration quotidienne avec Gabriele Facciolo a été un élément essentiel sans lequel la plupart des travaux présentés dans cette thèse n'auraient pas vu le jour. Je lui dois entre autres choses tout le savoir-faire informatique qu'il m'a inculqué patiemment avec une disponibilité à toute épreuve.

Tout ce que je n'ai pas appris de mes directeurs de thèse, je l'ai sans doute appris d'Enric Meinhardt-Llopis, dont l'art de transformer des questions mal formulées en des problèmes clairs et simples m'a souvent été d'un grand secours.

Je tiens également à remercier les membres du jury : mes rapporteurs Pascal Monasse et Marc Pierrot Deseilligny dont les remarques enrichissantes m'ont permis d'améliorer sensiblement le manuscrit ; Pablo d'Angelo, Hélène de Boissezon, Jean-Marc Delvit et Julien Michel, experts des agences spatiales allemande (DLR) et française (CNES) dont la présence valide à mes yeux le travail de cette thèse ; enfin François Goulette qui m'a fait l'honneur d'accepter de présider le jury.

Les travaux de cette thèse ont été menés dans le cadre du projet MISS (Mathématiques pour l'Imagerie Stéréoscopique et Spatiale) dont je remercie les infatigables organisateurs, Julien Michel et Myriam Cournet, du CNES, pour les réunions trimestrielles qui ont permis de faire des bilans réguliers de notre travail.

Merci aux équipes du SONC (Science Operation & Navigation Center) au CNES, notamment à Eric Jurado et Cédric Delmas, pour nous avoir impliqués dans l'aventure fascinante de la mission Rosetta en nous demandant de calculer un modèle 3D du site d'atterissage final de Philae.

J'ai eu la chance d'effectuer mon service d'enseignement au sein du département de mathématiques de l'Université Pierre et Marie Curie (UPMC), qui propose un très grand choix de cours. C'est ainsi que j'ai pu revisiter les fondements de l'algèbre linéaire dans le cours de Claire David et Andreas Hoering, redécouvrir les probabilités discrètes et la statistique dans le cours de Laurent Mazliak, et prendre goût à la théorie de la mesure dans le célèbre cours d'intégration de Thierry Lévy et Omer Adelman. Le souci constant de clarté de leurs explications, l'intérêt des questions, des exercices, des exemples sans cesse renouvelés et souvent originaux qu'ils proposent aux étudiants, et le style unique de leurs cours magistraux "à deux profs" resteront toujours pour moi un modèle de pédagogie.

Une partie du travail présenté dans le chapitre 5 sur l'algorithme SGM a été effectuée avec la participation substantielle de Félicien Comtat et Léo Tible dont j'ai encadré le stage de licence. Je les remercie pour leur enthousiasme et leur motivation.

La vie quotidienne au laboratoire n'aurait pas été aussi facile sans l'aide précieuse et efficace de Véronique, Micheline, Virginie, Sandra, Christophe, Sébastien et Atman. L'atmosphère détendue qui y règne est le fruit de la bonne humeur polyglotte installéé entre autres par Ives, Samy, Nicola, Martin, Rafael, Lara, José, Marc, Mauricio, Miguel, Morgan, Yohan, Nicolas, Loïc, Irène, Barbara, Rachel, Charles, Tristan, Hayato, Marie, Zhongwei et Javier. Et si je suis un peu (très peu) moins ignare en cinéma qu'au début de ma thèse, c'est sans doute gâce à Ives.

Enfin, merci à mes amis, à ma famille, et tout particulièrement à Lisa, pour leur soutien constant et indéfectible.

à Lisa

Contents

1 Introduction

1.1 Motivation

The number of optical Earth observation satellites has increased drastically over the past decade. According to the World Meteorological Organization[1], 65 civilian observation satellites are currently in operation, while at the end of the Nineties they were around 30. Their proliferation will certainly accelerate as new companies such as Skybox plan to launch dozens of low-cost satellites in the next few years. Starting during the Fifties, the development of the first Earth observation satellites was mostly driven by military needs related to the Cold War, but civilian applications followed very quickly in fields such as meteorology [Gordon 1962], cartography [Michel et al. 2013], geology [Rosu et al. 2014], hydrology [Ruelland et al. 2008], forestry [Gumbricht 2012] or glaciology [Marti et al. 2014]. Nowadays, images from Earth observation satellites are used to solve problems ranging from geographic mapping to meteorology predictions [Joseph 2015], including applications as diverse as the measurement of elevation changes for glaciers [Berthier et al. 2014] or rescue assistance for natural disasters [Yésou et al. 2015].

Earth observation satellites produce two-dimensional images. However, mapping and monitoring of the Earth's surface is essentially a three-dimensional problem. Even to produce 2D maps, accurate 3D models of the ground are needed in order to correct the distortions induced by the Earth's relief on the 2D images acquired by remote cameras. In most of the Earth observation use cases, both civilian and military, the crucial information is generally not contained in a particular image but in the changes that occur between images acquired at different times or dates. Geologists who study earthquakes, landslides, or ice flows are more interested in measuring the ground deformations over time than in assessing the land aspect at any given moment. In all the applications, as the displacements take place in a three-dimensional world, 3D measurements are needed to describe them. Even if one is not interested in measuring the changes but just in detecting them, working with 3D models is more reliable than working with images. Indeed many factors such as changes in sun exposure, weather or season can cause strong radiometric differences between images that do not correspond to a real change in the scene.

Modeling the 3D geometry of real scenes is a challenging problem. The approaches that have been developed to solve it are usually grouped in two categories: active methods [Jarvis 1983, Levoy et al. 2000] used in devices such as Kinect and Lidar, and passive image-based methods [Scharstein and Szeliski 2002, Seitz et al. 2006, Strecha et al. 2008]. Passive image-based methods are the cheapest. They use multiple photographs of the scene and compute the locations of 3D points that are visible in several images. This process is commonly known

[1] Observing Systems Capability Analysis and Review Tool, http://www.wmo-sat.info/oscar/satellites (accessed on 8 july 2015)

as *structure from motion*, or *multi-view stereo*, or simply *stereo matching* when used with only two views [Szeliski 2011, Furukawa and Hernández 2015]. The earliest stereo algorithms were developed in the field of *photogrammetry* to automatically construct topographic elevation maps from overlapping aerial images. In the field of *computer vision*, the topic of stereo matching has been one of the most studied problems [Marr and Poggio 1976, Barnard and Fischler 1982, Dhond and Aggarwal 1989, Scharstein and Szeliski 2002, Brown et al. 2003, Seitz et al. 2006], and continues to be one of the most active research areas.

Back to 1986 the French space agency, *Centre National d'Etudes Spatiales* (CNES), launched SPOT (Système Probatoire d'Observation de la Terre). It was the first imaging satellite to use a CCD (Charge Coupled Device) pushbroom sensor, which captures the images line by line as the satellite moves forward by using a linear array of sensors nearly perpendicular to the motion direction (pushbroom cameras will be presented in detail in chapter 2). It was also the first civilian satellite to provide off-nadir viewing capabilities, enabling the acquisition of stereo images. Taking images with a resolution of 10 m per pixel, SPOT was soon followed by other satellites with stereo capabilities and increasing spatial resolutions, such as Ikonos, WorldView, QuickBird and Pléiades to name just a few. While the spatial resolution of satellite images reached 30 cm per pixel on the recently launched WorldView-3 and will continue increasing, it will always be lower than that of aerial images, leading to less detailed 3D models. But satellite imaging presents several advantages over airplane-based imaging:

- for a large-scale usage over a long time range the acquisition cost is considerably lower,
- there are no distance limitations as all regions of the Earth's surface are quickly attainable by observation satellites at no cost,
- the delay between ordering and acquisition is usually shorter as observation satellites orbit the Earth in about 90 min and have a revisit period of a few days,
- the accuracy of the camera orientation parameters is greater as the path of a satellite does not suffer from atmospheric perturbations, hence is much more smooth.

Launched in 2011 and 2012, the Pléiades constellation is composed of two twin Earth observation satellites, Pléiades 1A and 1B, shown on figure 1.1. They fly on the same orbit, at about 694 km from the ground, and are phased 180° apart. Both are high resolution satellites, able to deliver images with a ground sampling distance of 70 cm (upsampled to 50 cm before delivery to end users) and a swath width of 20 km. The images have a pixel depth of 12 bits with a signal-to-noise ratio greater than 90 dB. The typical size of the images is about $40\,000 \times 40\,000$ pixels, i.e. 1.6 gigapixels, and the image height can be larger. The unique agility of the Pléiades satellites allows them to capture multiple views of the same area during a single orbit. This permits the nearly simultaneous acquisition of stereo datasets made of two or three images with a small base to height ratio, ranging from 0.15 to 0.8. As many other Earth observation satellites since SPOT, Pléiades acquire images with a pushbroom sensor. All the Pléiades images used in the experiments presented in this thesis were provided by CNES through the *Recette Thématique Utilisateurs* (RTU) program. Some of them can be viewed online[2].

The goal of this thesis is to build an algorithm to produce accurate 3D models from satellite images in a fully automatic way. Ideally the produced 3D models would be perfectly

[2] http://dev.ipol.im/~carlo/ipol_demo/catalogue_of_pleiades_images

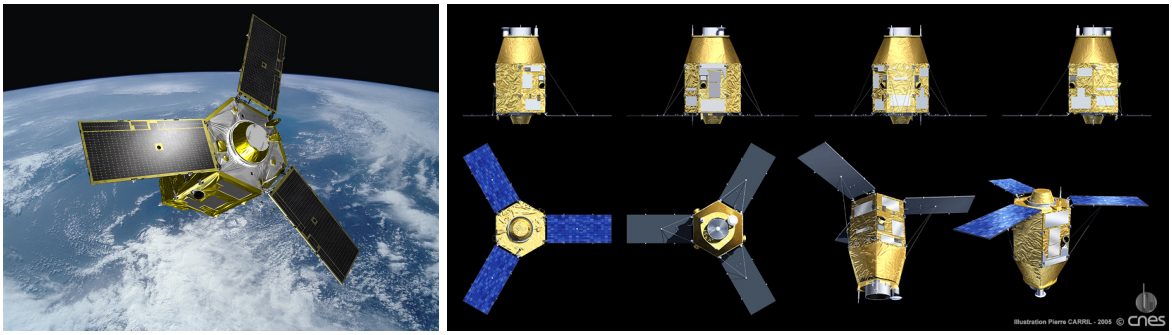


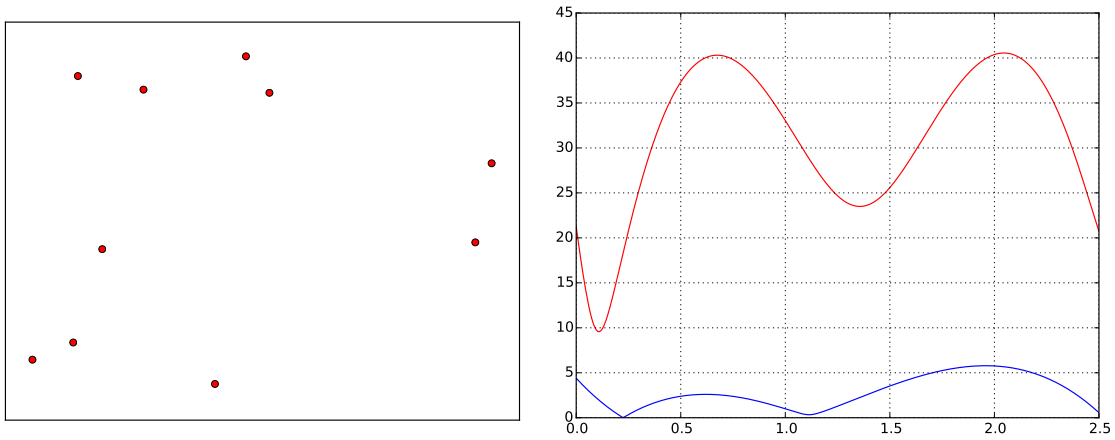
Figure 1.1 – Artist views of the satellite Pléiades-1A. Copyright ©Airbus DS/Master Image Films, 2007

georegistered and thus could be easily fused over time to obtain dense models. This would enable the automatic production of a high resolution digital model of the whole Earth's surface, which could be updated permanently and automatically. The `S2P` pipeline, whose development is still in progress, is a tentative answer. Many aspects of the 3D reconstruction problem have received a lot of attention in the past 30 years. The development of 3D reconstruction tools [Wohlfeil et al. 2012, Moratto et al. 2014, Pierrot Deseilligny 2015] was boosted by the increasing availability of high resolution stereo images from satellites. These tools have been shaped by the constraints of satellite imaging, like the large size of the images or the pushbroom camera geometry which imposes to use ad hoc stereo matching tools. These specializations have created a gap with respect to the fields of computer vision and image processing, where research focuses on small images taken with pinhole cameras. This thesis tries to fill this gap by adapting the tools of the computer vision and image processing communities to the specifics of satellite images.

We will start by describing a camera model for the pushbroom sensors used on Earth observation satellites, based on the model proposed by Westin [1990] in the early Nineties. This simplified model allows to identify the main causes for localization inaccuracies, and leads to an efficient correction algorithm that image vendors could use to improve the accuracy of the geometric calibration data that they sell with images. We will then change point of view and think as image users, trying to get the most of the stereo products proposed by image vendors. To that aim we will adapt the epipolar rectification technique to pushbroom images, and based on that we will propose a very simple algorithm to cancel the effects of calibration inaccuracies on the stereo matching performances. We will then study one of the most popular stereo matching algorithms, Semi-Global matching (SGM) [Hirschmüller 2008], that was developed at the German space agency (DLR), and present our own variant. Finally we will describe `S2P`, a complete stereo pipeline for pushbroom images made with the building blocks described in the previous chapters, and illustrate its usage with an application to geomorphology. The next sections give an overview of the thesis, chapter by chapter.

1.2 Camera Modeling for Orbiting Pushbrooms

This chapter describes a simple pushbroom camera model for Earth observation satellites and proposes a new algorithm to refine the orientation parameters of a camera from a set of ground control points. The relative importance of the various orientation parameters is an-



(a) Input image points

(b) Localization error in meters

Figure 1.2 – Example of results produced with the online demo of the attitude refinement algorithm. Red curves show the error before correction and blue curves after correction. The user clicked on the image domain to select the input image points shown in (a). The localization error (in meters) as a function of time (in seconds) is shown in (b). The red curve gives the error made using the measured attitudes, while the blue curve gives the error made using the estimated attitudes.

alyzed. On the last generation of high resolution satellites such as Pléiades and WorldView, the attitude angles are shown to be the main contributors to localization errors. Thus the proposed algorithm focuses on refining the attitude angles. It is based on a simple polynomial fitting method. Numerous experiments, which can be reproduced through the online demo associated to this chapter, show that the proposed algorithm is able to reduce the localization error by one order of magnitude with only a few ground control points. Figure 1.2 gives an example of the algorithm result. A geometric simulator for the proposed model is implemented, as well as the attitude refinement algorithm.

1.3 Epipolar Rectification of Pushbroom Images

Epipolar rectification is a common technique used in 3D reconstruction algorithms. It permits to simplify the search of corresponding points between the images of a stereo pair. However, it applies only to images taken with pinhole cameras. Pushbroom cameras produce images for which any epipolar rectification is theoretically not possible. In this chapter we study to what extent it is possible to apply epipolar rectification to pushbroom images anyway, in order to use standard matching algorithms from the image processing and computer vision communities. The approach presented here considers the epipolar rectification as an auxiliary step for the computation of stereo correspondences, not as a final product. Images are thus processed in small tiles by locally approximating the pushbroom camera with an affine camera model. The explicit modeling of the approximation allows to quantify and control the epipolar errors without needing ground control points. Experiments on Pléiades and WorldView-1 images of many kinds of scenes (urban, mountainous, flat) demonstrate that epipolar errors can be reduced to one tenth of pixel, as shown on figure 1.3.

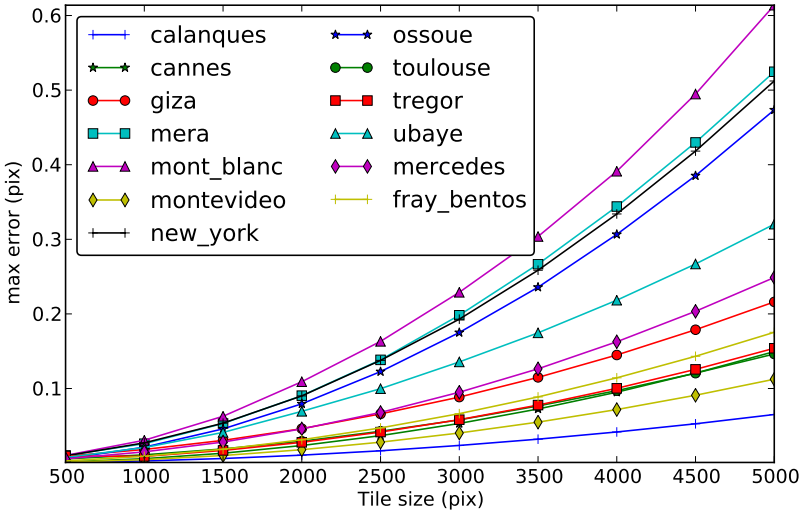


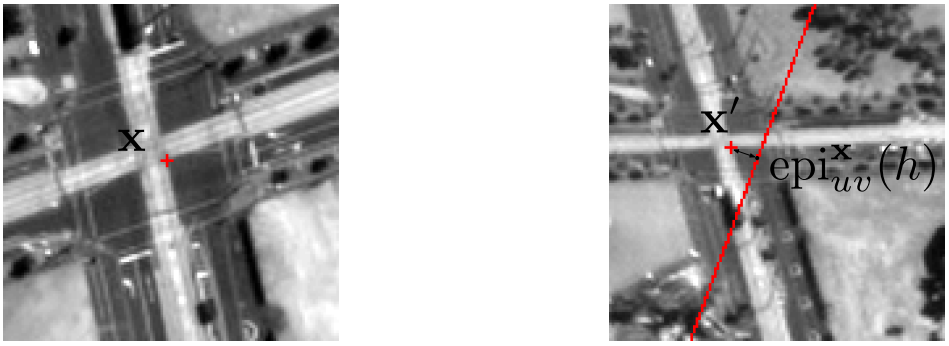
Figure 1.3 – Dependence of the epipolar error with the tile size. Each curve corresponds to a stereo pair. A tile of size ranging from 500×500 to 5000×5000 pixels was selected in the middle of a Pléiades reference image. Virtual matches were computed using the refined altitude range given by the SRTM data. The error reported by each plot is the maximal distance between a point and the epipolar line of its match.

1.4 Local Correction of the Relative Pointing Error

Modern Earth observation satellites are calibrated in such a way that a point on the ground can be located in the image domain with an error of just a few pixels. For many applications this error can be ignored, but this is not the case for stereo reconstruction which requires sub-pixel accuracy. In this chapter we propose a method to correct this error. The method works by estimating local corrections that compensate the error relative to a reference image. An example of this *relative pointing error* is shown on Figure 1.4. The proposed method does not rely on ground control points, but only on the relative consistency of the image contents. We validate our method with Pléiades and WorldView-1 images on a representative set of geographic sites.

1.5 Stereo Processing by Semi-Global Matching and Beyond

Semi-Global Matching (SGM) is among the top-ranked stereo vision algorithms. SGM is an efficient strategy to compute an approximate global minimum for an energy that comprises a pixel-wise matching cost and pairwise smoothness terms. In SGM the two-dimensional smoothness constraint is approximated as the average of one-dimensional line optimization problems. The accuracy and speed of SGM are the main reasons for its widespread adoption, even when applied to generic problems beyond stereo vision. However, this approximate minimization produces characteristic low amplitude streaks in the final disparity image, and is clearly suboptimal with respect to more comprehensive minimization strategies. Based on a recently proposed interpretation of SGM as a min-sum Belief Propagation algorithm, we propose a new algorithm that injects information from the 2D problem in the processing of SGM's 1D paths, as illustrated by figure 1.5. It allows to reduce by a factor five the energy



(a) First view u. (b) Second view v.

Figure 1.4 – Effect of the satellite relative pointing error on a pair of Pléiades views of a road intersection in Haiti. Two corresponding points $\mathbf{x}$ and $\mathbf{x}'$ are shown, and the epipolar curve of $\mathbf{x}$ as traced by the RPC does not pass through the corresponding point $\mathbf{x}'$. The *relative pointing error* is the distance from the point $\mathbf{x}'$ to the epipolar curve $\text{epi}^{\mathbf{x}}_{uv}(\mathbf{R})$.

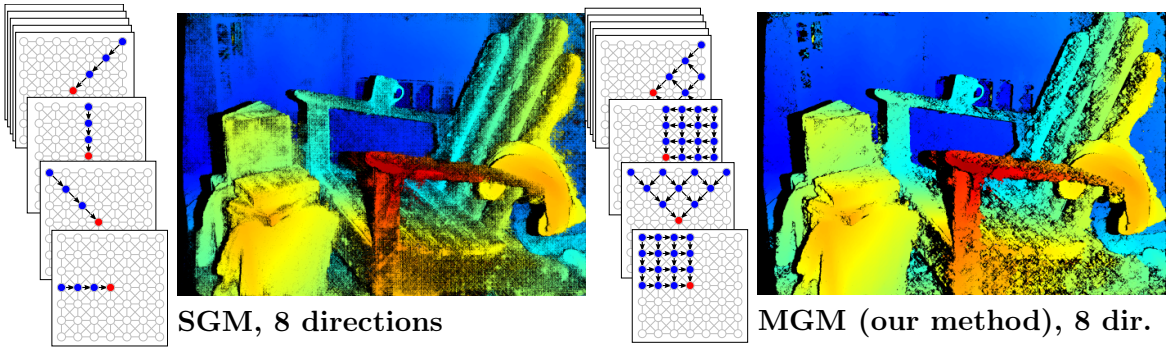


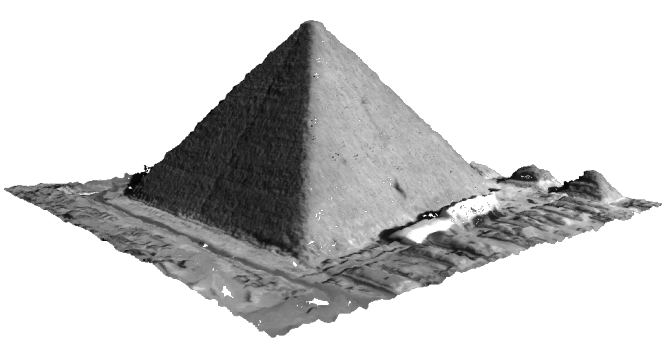
Figure 1.5 – Results of the baseline implementation of SGM and MGM on a stereo pair from the Middlebury evaluation benchmark dataset (version 3). The diagrams on the left depict for each method the information used by the recursive update of the red pixel in each of the 8 scans of the algorithms.

gap of SGM with respect to reference algorithms for Markov Random Fields (MRF) with truncated smoothness terms. The proposed method comes with no compromises with respect to the baseline SGM, no parameters and virtually no computational overhead. At the same time it attains higher quality results by removing the characteristic streaking artifacts of SGM.

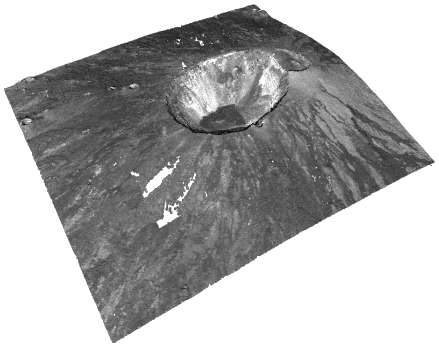
1.6 A Stereo Pipeline for Pushbroom Images: S2P

By assembling the algorithmic blocks described in the previous chapters, this chapter proposes a fully automatic and modular stereo pipeline to produce digital elevation models from satellite images. The aim of this new pipeline is to use and test off-the-shelf computer vision tools, while abstracting from the complexity associated to satellite imaging. To this aim, images are cut in small tiles for which chapter 3 proves that the pushbroom geometry is very accurately approximated an affine camera model. These tiles are then processed with standard rectification and stereo matching tools. The specifics of satellite imaging such

as pointing accuracy refinement, estimation of the initial elevation from SRTM data, and geodetic coordinate systems are handled transparently by the pipeline. We demonstrate the robustness of our approach on a large database of satellite images and by providing an online demo of the pipeline. As a teaser, Figure 1.6 shows views of 3D models obtained with S2P from two Pléiades pair of images of the Piton the la Fournaise and the Great Pyramid of Giza.



(a) The Great Pyramid of Giza, also known as the Pyramid of Cheops.

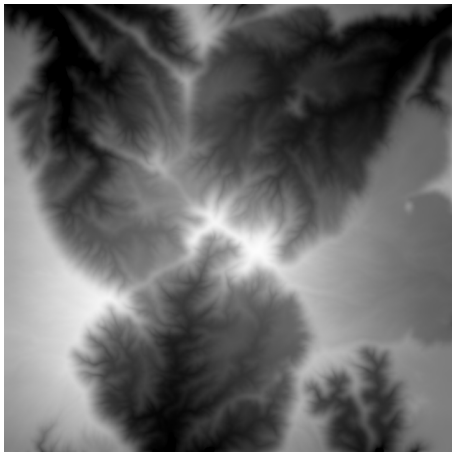


(b) Crater of the *Piton de la Fournaise*, La Réunion.

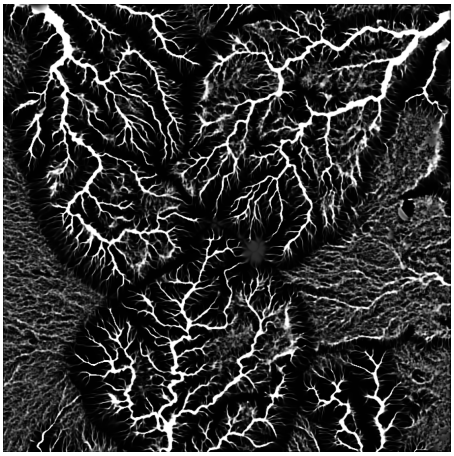
Figure 1.6 – Examples of 3D models computed with S2P from Pléiades stereo images, without any manual intervention.

1.7 Simulation of Landscape Evolution

This chapter illustrates how the main physical laws proposed in landscape evolution models (LEMs) can be modeled by a system of three partial differential equations governing water runoff, stream incision, hill slope evolution and sedimentation. Several numerical experiments on high resolution digital elevation models (DEMs) obtained from stereo image pairs of the satellite Pléiades thanks to the algorithm S2P presented in chapter 6 illustrate its potential to simulate the fine structure of the river network, and to understand the landscape morphology and its causes. They also permit to simulate plausible evolutions, as shown on Figure 1.7. The experiments on digital elevation models of the same site (La Réunion) are made at three different resolutions: the SRTM resolution, 90 m, and then 12 m and 4 m on digital elevation models derived from several Pléiades pairs of the same site. The experiments show that the water runoff simulation critically depends on the DEM resolution. This justifies the use of high resolution DEMs like those that can now be produced from Pléiades pairs or triplets.



(a) A plausible evolution of the digital elevation model where 10% of the terrain has been removed by erosion.



(b) Water height at the end of the evolution. Bright pixels represent high values, as in terrain elevation models, showing the structure of the water network in white.

Figure 1.7 – Example of results obtained by simulating the landscape evolution of the Piton des Neiges, La Réunion, starting from a digital elevation model computed by S2P from a Pléiades stereo pair.

List of Publications

The work in this thesis has led to the following publications:

- de Franchis, C., Meinhardt, E., Michel, J., Morel, J.-M., and Facciolo, G. *Automatic sensor orientation refinement of Pléiades stereo images.* In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2014.
- de Franchis, C., Meinhardt, E., Michel, J., Morel, J.-M., and Facciolo, G. *On stereo-rectification of pushbroom images.* In Proceedings of the IEEE International Conference on Image Processing (ICIP), 2014.
- de Franchis, C., Meinhardt, E., Michel, J., Morel, J.-M., and Facciolo, G. *An automatic and modular stereo pipeline for pushbroom images.* In ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. II, 2014.
- de Franchis, C., Meinhardt, E., Michel, J., Morel, J.-M., and Facciolo, G. *Automatic digital surface model generation from Pléiades stereo images.* Revue Française de Photogrammétrie et de Télédétection, vol. 208, 2014. Awarded the SFPT prize for the *Best student paper of the year*.
- Chen, A., Darbon, J., de Franchis, C., Facciolo, G., Meinhardt, E., Michel, J., and Morel, J.-M. *Numerical simulation of landscape evolution and water run-off on digital elevation models obtained from Pléiades.* Revue Française de Photogrammétrie et de Télédétection, vol. 209, 2015.
- Michel, J., de Franchis, C. Meinhardt, E., Facciolo, G. *S2P a new open-source stereo pipeline for satellite images.* In Geomatics Workbooks n°12 – FOSS4G Europe, 2015.

- Facciolo, G., de Franchis, C., and Meinhardt, E. *MGM: A Significantly More Global Matching for Stereovision*. In Proceedings of the British Machine Vision Conference, 2015.
- de Franchis, C., Meinhardt, E., Greslou, D., and Facciolo, G. *Attitude Refinement for Orbiting Pushbroom Cameras: a Simple Polynomial Fitting Method*. Image Processing On Line (IPOL), submitted.

List of Online Demos

The algorithms presented in this thesis can be tested in the following online demos:

- Attitude refinement for pushbroom cameras:
  http://dev.ipol.im/~carlo/ipol_demo/pushbroom_ba/
- Stereo-rectification and pointing correction of pushbroom image tiles
  http://dev.ipol.im/~carlo/ipol_demo/s2p_rectification/
- Semi-Global Matching (SGM) and More Global Matching (MGM)
  http://dev.ipol.im/~facciolo/ipol_demo/mgm/
- Satellite Stereo Pipeline (`S2P`):
  http://dev.ipol.im/~carlo/ipol_demo/workshop_s2p/
- Simulation of landscape evolution:
  http://dev.ipol.im/~carlo/ipol_demo/landscape_evolution/

The Pléiades stereo datasets used in most of the experiments can be viewed online at http://dev.ipol.im/~carlo/ipol_demo/catalogue_of_pleiades_images/.

2 Camera Modeling for Orbiting Pushbrooms

This chapter describes a simple pushbroom camera model for Earth observation satellites and proposes a new algorithm to refine the orientation parameters of a camera from a set of ground control points. The relative importance of the various orientation parameters is analyzed. On the last generation of high resolution satellites such as Pléiades and WorldView, the attitude angles are shown to be the main contributors to localization errors. Thus the proposed algorithm focuses on refining the attitude angles. It is based on a simple polynomial fitting method. Numerous experiments, which can be reproduced through the online demo associated to this chapter, show that the proposed algorithm is able to reduce the localization error by one order of magnitude with only a few ground control points. A geometric simulator for the proposed model is implemented, as well as the attitude refinement algorithm.

The attitude refinement algorithm presented in this chapter can be tested online on any input at http://dev.ipol.im/~carlo/ipol_demo/pushbroom_ba.

2.1 Introduction

Most of the problems relying on remote sensing images require accurately georeferenced images. For instance, when using multi-temporal images of the same area to monitor changes in vegetation, water bodies, or geological landmarks, the images have to be accurately registered in order to provide meaningful measurements. Obviously for this use case a relative registration of the images may suffice to the end users, but it implies a preprocessing specific to the application. In an ideal world, all satellite images would be natively georeferenced to sub-pixel precision.

To georeference an image, two things are necessary. First an accurate geometric model of the camera with all the numerical values of its parameters, and second a digital surface model of the ground area corresponding to the viewed scene. This chapter deals with the first thing. Observation satellites use pushbroom cameras which record the scene line by line as the satellite moves forward on its orbit. The resulting image is a stack of lines taken at slightly different times, from different positions and viewing angles. The camera position and viewing direction in space (usually referred to as *attitude*) are recorded on board during the image acquisition. The frequency and precision of these measurements depend on the satellite, but even in the best cases the resulting accuracy is limited and needs to be refined using the image content. A common solution is to use known correspondences between *ground control points*, i.e. 3D points located in the viewed scene, and their projection on the image. The camera parameters are then estimated as those that best fit the correspondences according to the geometric camera model.

2.1.1 Previous Work

Many papers describing pushbroom camera models have been published since the early Fifties and the advent of aerial photography. A thorough analysis of these works was made recently by Poli and Toutin [2012] in their last review paper, and also by Konecny [2003] and Joseph [2015] in their respective textbooks. An exhaustive review was already made by Toutin [2004b]. We give here a quick historical overview of the developments of pushbroom camera models and estimation algorithms, but the interested reader should refer to these publications for a comprehensive coverage.

Papers dealing with pushbroom cameras were published even before the launch of the first observation satellites, as pushbroom cameras were used on aircrafts [Case 1967]. In these papers pushbroom cameras are generally referred to as *strip cameras*.

The first camera model for orbiting pushbroom cameras was published by Guichard [1983] and Toutin [1985]. The model was used for SPOT-1 which was the first civilian Earth observation satellite to be launched after the American Landsat series. Later this model became known as the *Toutin Model* and was embedded in the commercial software PCI Geomatica. This model assumes that the satellite flies on a perturbed elliptical orbit and that its nominal constant attitude is perturbed by second degree polynomial functions of time. Later the model was extended and adapted to newer observation satellites [Toutin 2003, 2004a, 2006a, b].

In 1987, Konecny proposed to model the movement of orbiting pushbroom cameras with a straight path and a constant attitude, which leads to a model with six parameters [Konecny et al. 1987]. The differences between that model and reality were handled by eight perturbation parameters. This model was extended later by Jacobsen [1998, 2003, 2004].

In 1987, Gugan proposed an orbiting pushbroom camera model for SPOT-1 in which the satellite flies an elliptical orbit whose longitude of the ascending node changes linearly with time, as well as the satellite true anomaly [Gugan 1987]. The attitudes are assumed to drift linearly with time, and their initial value is unknown. This model was extended by Michalis and Dowman [2008].

In 1988, Kratky proposed to model the attitude variations of SPOT-1 by quadratic functions of time [Kratky 1988, 1989b]. This model was implemented in the software SPOT-CHECK+ and was later extended by Fritsch and Stallmann [2000].

In 1990, Westin proposed to simplify the usual models by approximating the orbit of SPOT-1 by a plane circular orbit [Westin 1990]. This approximation has to be valid only during the timespan of one scene. He reported experiments on SPOT-1 images showing that the ground error due to this approximation is less than 10 cm. To further simplify the model, the attitude derivatives measured on board are assumed to be accurate enough, leaving only a constant term (i.e. the initial attitude) to be estimated. An iterative least-squares minimization was proposed to estimate the seven parameters of the model (four for the circular orbit and three for the constant attitude term). Sub-pixel accuracies were obtained on SPOT-1 images using a limited number of control points.

In 1995, Gupta and Hartley described a model similar to Westin's model, in a paper aimed at describing a new algorithm for mapping 3D points to the image plane [Gupta and Hartley 1995]. The only difference of their model with respect to Westin's is that the satellite path is assumed to be elliptic instead of circular. But Gupta and Hartley do not explain how

the orbit parameters and the attitude offset are estimated from ground control points. In 1997 they introduced and gave a rigorous analysis of the *linear pushbroom camera* [Gupta and Hartley 1997]. This model is considerably simpler than the usual models because it assumes that the satellite flies at a uniform speed on a straight line and that its attitude is constant. As a consequence, the relationship between image and ground coordinates is linear in terms of homogeneous coordinates, almost as with standard pinhole cameras. The authors reported a maximal error of half a pixel on a whole SPOT-1 scene when comparing the linear pushbroom model to the more realistic model they published two years earlier [Gupta and Hartley 1995].

In 2005, Poli proposed to model both the position and attitude errors by second order piecewise polynomials depending on time [Poli 2005, 2007, Poli et al. 2014]. For pushbroom cameras carried by satellites, two segments were reported to be sufficient in general [Poli 2007]. The position and attitude errors are used to correct the cubic splines interpolated from the position and attitude samples measured on board and provided by the image vendors.

In 2007, Crespi et al. proposed to model the satellite path by an elliptical orbit and the attitude errors by second order polynomials depending on time [Crespi et al. 2007]. The measured attitude functions are obtained by interpolation from samples provided by vendors. This model was implemented in the SISAR software, but the implementation is not distributed. The whole model was detailed and analyzed later in a book chapter [Crespi et al. 2010].

2.1.2 Our Contribution

Poli and Toutin [2012] grouped the major Earth observation satellites in several classes and subclasses. This chapter focuses on the most recent subclass of satellites, which scan the Earth in *asynchronous* mode and use a so-called *agile configuration*. These satellites fly along sun-synchronous and quasi-polar orbits and provide very high resolution images using a flexible single-lens optical system. They have the ability to rotate about the axes of their camera in order to point at off-nadir targets up to 30°, or to view the target from different directions during a single orbit. Some well-known commercial satellites of this category are Pléiades 1A and 1B, QuickBird, WorldView 1, 2, and 3, GeoEye 1, Eros A and B, Orbview 3 and TopSat.

This chapter proposes a simple model for this kind of orbiting pushbroom cameras and analyzes the main causes for localization inaccuracies. A simple algorithm is proposed to improve the localization accuracy. To the best of our knowledge, the work of this chapter is the first that provides an open source implementation of an orbiting pushbroom camera simulator as well as an online demo allowing to test the proposed attitude refinement algorithm.

The next section exposes the camera model used in the chapter. Section 2.3 analyzes the main causes for localization inaccuracies and proposes an algorithm to reduce them. Section 2.4 presents numerous synthetic experiments to validate the proposed approach.

2.2 Pushbroom Camera Model

This section describes the pushbroom camera model used in this chapter. This model is an extension of the model that was described by Westin [1990]. The main difference is that

our model is built for satellites whose attitude vary continuously during the timespan of one scene, while the models made for SPOT-1 [Westin 1990, Guichard 1983, Toutin 1985, Konecny et al. 1987, Gugan 1987, Kratky 1988, Gupta and Hartley 1995] were based on the assumption that the attitude was unknown but almost constant. In our model the attitude evolution is thus described by polynomial functions of degree 3.

2.2.1 The Linear Pushbroom Camera

The goal of a camera model is to define the geometric relationship between image and space coordinates. We will first recall how to derive this relationship with the specialized case of the *linear pushbroom camera* that was introduced by Gupta and Hartley [1997]. The linear pushbroom camera model relies on two simplifying assumptions:

- the camera is traveling in a straight line at a constant velocity with respect to the world,
- the attitude of the camera is constant over the whole image acquisition duration.

This camera can be thought of as a standard pinhole camera [Hartley and Zisserman 2004] moving at constant speed along a straight line in 3-space, with fixed attitude. This pinhole camera would have the additional property of containing a single line of pixel sensors in its focal plane instead of the usual bi-dimensional array of pixel sensors. Thus, at any moment, only the points lying in the plane defined by the optical center and the line of pixel sensors are imaged by the camera. This plane is called the *view plane*, and is shown on Figure 2.1. The orientation of the view plane depends only on the camera attitude, and is assumed to be fixed. It is also assumed that the motion of the camera does not lie in the view plane. Consequently, the view plane sweeps out the whole 3-space as time varies. The image provided by such a camera is made of lines scanned independently at different instants of time and stored one next to the other.

All the coordinates on Figure 2.1 are expressed in the instantaneous camera frame, defined below.

Definition 1. *The* camera frame *is an orthonormal coordinate frame attached to the moving camera. The origin of the coordinate frame is the instantaneous camera center. The* Z*-axis lies in the instantaneous view plane and is perpendicular to the linear sensor array. It is directed towards the visible half-space. The* X*-axis is orthogonal to the view plane and its direction is determined by requiring that the motion of the camera has a positive* X *component. The* Y*-axis lies in the view-plane, is parallel to the sensor array, and its direction is determined by requiring that the* X, Y*, and* Z *axes form a right-handed coordinate frame.*

The image of a 3-space point $\mathbf{X}$ is described by its row and column coordinates x and y. The first coordinate x gives the time instant at which the point $\mathbf{X}$ was imaged, i.e. lay in the view plane. If δ_t denotes the time spent by the sensor to measure one sample, then the instant of time at which the row x was imaged is $x\delta_t$ (assuming that the first row was imaged at $t = 0$). The duration δ_t is called the *dwell time*. It is part of the internal parameters of the sensor.

The second coordinate y gives the position of the projection of $\mathbf{X}$ on the sensor. It is a 2D projection carried out in the view plane. The coordinates of the projected point can be

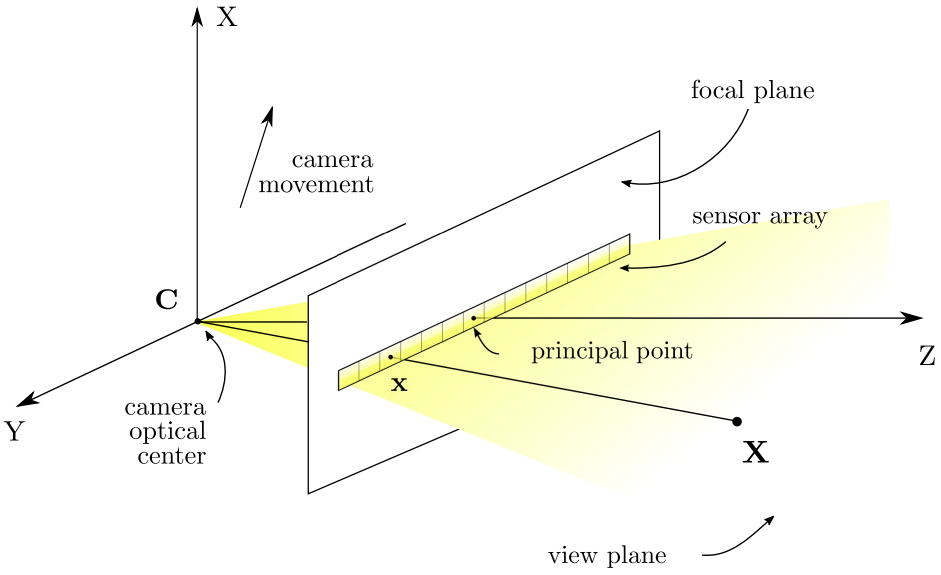


Figure 2.1 – Linear pushbroom camera model. The instantaneous camera optical center $\mathbf{C}$ moves at a constant speed. The view plane sweeps out the whole 3-space as the camera moves forward. The 3D point $\mathbf{X}$ is imaged by the camera to the focal plane location $\mathbf{x}$.

expressed in the camera frame. In that frame the view plane is defined by $\mathrm{X} = 0$. If f denotes the *focal length* of the camera, it follows from similar triangles that the point $\mathbf{X} = (0, \mathrm{Y}, \mathrm{Z})^\top$ is mapped to the point $(0, f\mathrm{Y}/\mathrm{Z}, f)^\top$, as shown on Figure 2.2.

The coordinate y is a pixel index expressed in the sensor frame, which differs from the camera frame by a translation. The magnitude of this translation is given by the *principal point* coordinate y_0. It is the pixel index of the projection of the camera center on the sensor array, as shown on Figure 2.3. If w denotes the *pixel size* measured with the camera frame units, then $w(y - y_0) = f\mathrm{Y}/\mathrm{Z}$.

This permits to describe the locus of 3-space points that may have been imaged on a given image point.

Result 1. *Let $\mathbf{x} = (x, y)^\top$ be an image point, where x is the row index and y the column index. Take as coordinate frame the instantaneous camera frame at time $t = x\delta_t$. Then the inverse image of $\{\mathbf{x}\}$ by the linear pushbroom camera mapping is the 3-space line passing through the origin and directed by $(0, w(y - y_0), f)^\top$.*

2.2.2 Camera Rotation

Most modern high resolution commercial satellites such as those of the Pléiades and WorldView series scan the Earth in *asynchronous mode* [Poli and Toutin 2012]. The word *asynchronous* refers to the fact that the satellite speed and ground scanning speed are not equal: the satellite moves faster than the projection of its sensor array on the ground. On the one hand this ensures a longer exposure time, which allows more photons to reach the CCD (Charge Coupled Device) sensor and thus improves the signal to noise ratio [Poli 2005]. On

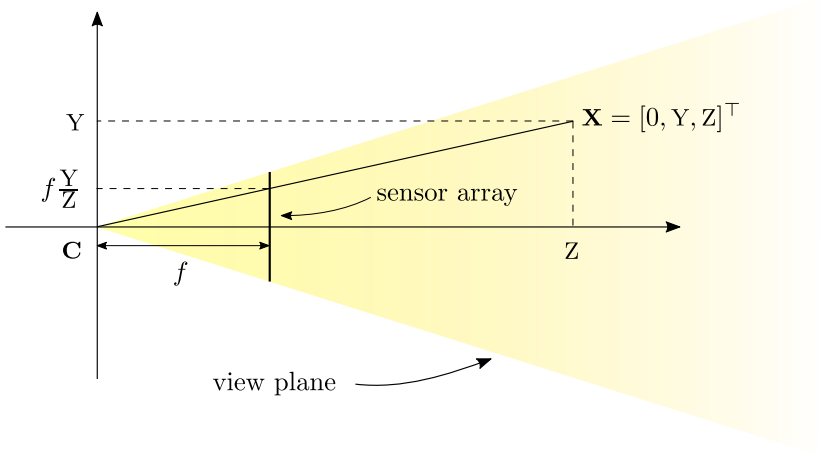


Figure 2.2 – For 3-space points in the view plane, the image formation works in the same way as with a pinhole camera. From similar triangles one knows that the ratio between the Y coordinate of the image point and f is equal to $\frac{\mathrm{Y}}{\mathrm{Z}}$.

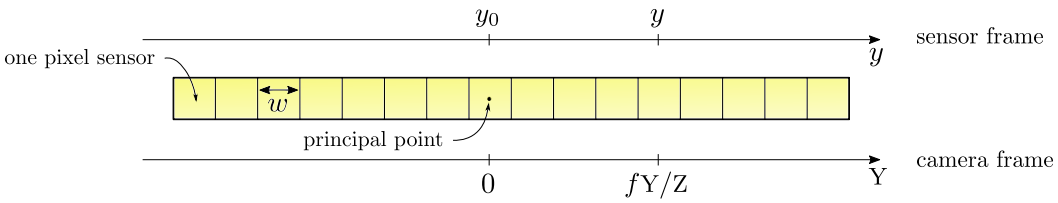


Figure 2.3 – The sensor frame is used to express the coordinate of a pixel in a given image row. Its relationship with the camera frame depends on internal parameters such as the pixel size w and the principal point coordinate y_0.

the other hand, it implies that the camera attitude is not constant during one image acquisition, i.e. that the second assumption made by the linear pushbroom camera model is not valid, as illustrated in Figure 2.4. This makes the image formation model slightly more complicated and increases the number of parameters needed to describe the image acquisition system.

The first consequence of the variation of attitude angles over time on our camera model is that the camera frame has no longer a fixed orientation. To convert coordinates between the camera frame and a fixed world frame, an intermediate coordinate frame is thus needed. This frame is the *local orbital frame*.

Definition 2. *The* local orbital frame *is an orthonormal coordinate frame attached to the moving satellite. The origin of the coordinate frame is the satellite center of mass. The* Z*-axis points towards the center of the Earth. The* X*-axis is defined by the motion of the satellite. The* Y*-axis is determined by requiring that the* X, Y*, and* Z *axes form an orthonormal right-handed coordinate frame.*

For the sake of simplicity the camera center is supposed to coincide with the satellite center of mass. The camera frame and the local orbital frame have thus the same origin, and differ only by a rotation. This rotation is described by its roll, pitch and yaw components

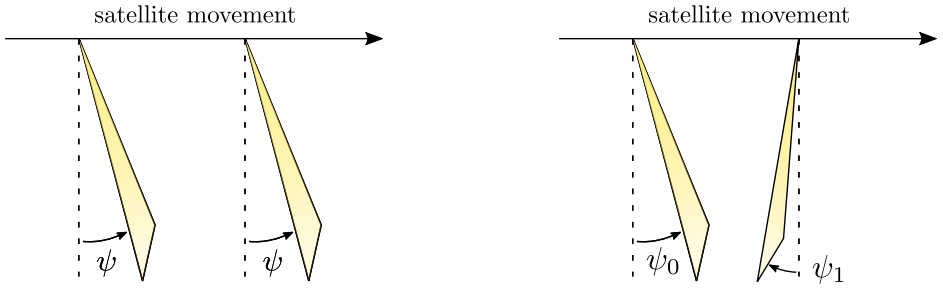


Figure 2.4 – *Synchronous* observation satellites, shown on the left, have fixed attitudes: the orientation of the camera with respect to the satellite does not change over time. On *asynchronous* observation satellites, shown on the right, the attitude angles may vary during the image acquisition, such as the pitch angle ψ shown here.

(φ, ψ, ω). To obtain the camera frame, the local orbital frame is first rotated about its X axis by the roll angle φ, then rotated about its new axis Y′ by the pitch angle ψ, and finally rotated about the twice new axis Z″ by the yaw angle ω. These three rotations are shown on Figure 2.5. As the three elemental rotations occur about axes of the rotating coordinate system (they are called *intrinsic rotations*), the resulting transformation matrix is obtained by composing the associated *Givens* rotation matrices [Hartley and Zisserman 2004] from left to right:

$$\mathtt{R}(\varphi,\psi,\omega) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\varphi & -s_\varphi \\ 0 & s_\varphi & c_\varphi \end{bmatrix} \begin{bmatrix} c_\psi & 0 & s_\psi \\ 0 & 1 & 0 \\ -s_\psi & 0 & c_\psi \end{bmatrix} \begin{bmatrix} c_\omega & -s_\omega & 0 \\ s_\omega & c_\omega & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.1}$$

where c_a and s_a denote $\cos(a)$ and $\sin(a)$. Note that the values of the attitude angles φ, ψ, ω depend on t. The change of basis matrix from the local orbital frame to the camera frame will usually be written as $\mathtt{R}(\varphi(t), \psi(t), \omega(t))$.

Result 2. *Let $\mathbf{x} = (x, y)^\top$ be an image point, where x is the row index and y the column index. In the local orbital frame at time $t = x\delta_t$, the inverse image of $\{\mathbf{x}\}$ by the pushbroom camera mapping is the 3-space line passing through the origin and directed by*

$$\mathtt{R}(\varphi(t), \psi(t), \omega(t)) \begin{pmatrix} 0 \\ w(y - y_0) \\ f \end{pmatrix}. \tag{2.2}$$

Our model makes the extra assumption that the roll, pitch and yaw are polynomial functions of time of degree 3:

$$\begin{aligned} \varphi(t) &= \varphi_0 + \varphi_1 t + \varphi_2 t^2 + \varphi_3 t^3 \\ \psi(t) &= \psi_0 + \psi_1 t + \psi_2 t^2 + \psi_3 t^3 \\ \omega(t) &= \omega_0 + \omega_1 t + \omega_2 t^2 + \omega_3 t^3 \end{aligned}$$

The attitude of the camera is thus entirely described by the set of coefficients of these 3 polynomial functions, which is denoted by $\Phi = \{(\varphi_k, \psi_k, \omega_k)\}_{k=0,\dots,3}$. Additional terms may also be considered to model certain oscillations of the satellite attitude [Perrier et al. 2015], but this is beyond the scope of this thesis.

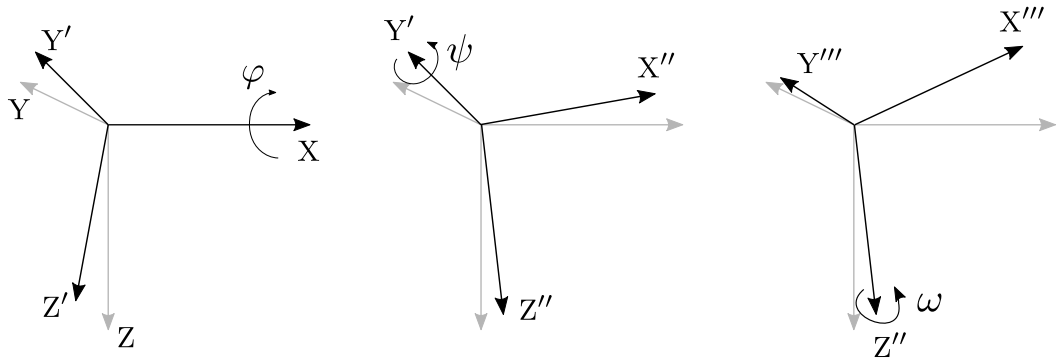


Figure 2.5 – The camera frame (represented in black) is obtained from the local orbital frame (in gray) by composing three *Givens* rotations [Hartley and Zisserman 2004] about the three axes of the rotating frame. The first rotation is about the X axis, thus the X axis does not change and $\mathrm{X}' = \mathrm{X}$. In the same way $\mathrm{Y}'' = \mathrm{Y}'$ and $\mathrm{Z}''' = \mathrm{Z}''$. The roll angle φ shown on the left figure is negative, while the pitch and yaw angles ψ and ω shown on the middle and right figures are positive.

2.2.3 Satellite Movement

Real satellites have elliptical orbits around the Earth as described by Kepler's laws. Gupta and Hartley [1997] reported a discrepancy of about 5 m when approximating an orbiting pushbroom with a linear pushbroom camera model on images from the first generation of SPOT satellites. As the first SPOT satellites had constant attitudes, the main factor responsible for this discrepancy might be the first assumption of the linear pushbroom camera model, i.e. the uniform and linear camera movement hypothesis.

Orbit description. In this chapter the orbit of the satellite is assumed to be a circle centered on the Earth's center of mass, as in [Westin 1990]. It is described by three parameters: the radius of the circle, plus two angles giving the orientation of the plane in which the circle lies.

The radius is $R_E + a$, where R_E denotes the Earth radius and a the flying altitude of the satellite, usually around 700 km. The orientation of the orbital plane is given by its *inclination* $i \in [0, \pi]$ and its *reference node longitude* $\lambda_0 \in [-\pi, \pi]$. The inclination is the angle formed with the equatorial plane, as shown on Figure 2.6. The reference node is the intersection between the orbit and the equatorial plane at which the satellite is ascending (i.e. going towards north). The longitude λ_0 is measured with respect to the *inertial frame*.

Definition 3. *The* Earth-centered inertial frame *is an orthonormal coordinate frame attached to the center of mass of the Earth, whose axes have fixed directions. Its* Z *axis is the Earth rotation axis, oriented towards the North pole. Its* X *axis lies in the equatorial plane and is permanently fixed in a direction relative to the celestial sphere. The* Y*-axis is determined by requiring that the* X, Y*, and* Z *axes form an orthonormal right-handed coordinate frame.*

The inertial frame does not rotate with the Earth. This means that any particular location on Earth, except the two poles, has coordinates that change over time. Coordinates of points on Earth are usually expressed in a frame that rotates with the Earth, namely the *Earth centered rotational frame*, defined below.

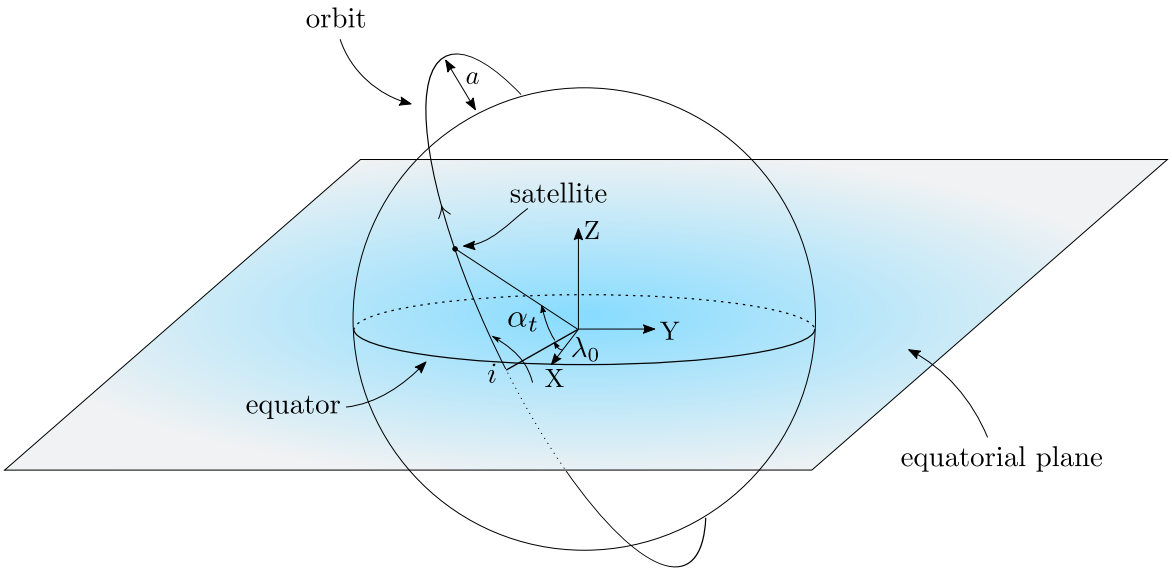


Figure 2.6 – A circular orbit around the Earth is described by three parameters: its radius $R_E + a$, the longitude λ_0 of its reference node, and its inclination i. The longitude λ_0 is defined with respect to the **X** axis of the inertial frame. The inclination i is the angle between the equatorial plane and the orbital plane. The satellite is localized by its angular position α_t. Note that the longitude represented here is negative.

Definition 4. *The* Earth centered rotational frame*, or* rotational frame *for short, is an orthonormal coordinate frame attached to the rotating Earth. The origin of the coordinate frame is the center of mass of the Earth. The* Z*-axis is pointing towards the north. The* X*-axis intersects the surface of the Earth at* 0° *latitude (equator) and* 0° *longitude (Greenwich). The* Y*-axis is determined by requiring that the* X, Y*, and* Z *axes form an orthonormal right-handed coordinate frame. The coordinates of a point in that frame are called its* geographic coordinates*. They are usually given in spherical form by the* longitude λ*, the* latitude θ *and the* altitude h *above the surface of the sphere.*

For the sake of simplicity, we assume that the rotational frame coincides with the inertial frame at the beginning of the acquisition ($t = 0$). The frame rotates with the Earth, therefore coordinates of a fixed point on the Earth surface do not change over time.

Position of the satellite. The position of the satellite on the orbit is given by an angle $\alpha_t \in [0, 2\pi]$. As the orbit is supposed to be circular and uniform, α_t depends only on its initial value α_0 and on the time t elapsed since the beginning of the acquisition:

$$\alpha_t = \alpha_0 + 2\pi \frac{t}{T_S}, \tag{2.3}$$

where T_S denotes the orbital period of the satellite. It is known from Kepler's third law that

$$\frac{2\pi}{T_S} = \sqrt{\frac{\mu}{(R_E + a)^3}}, \tag{2.4}$$

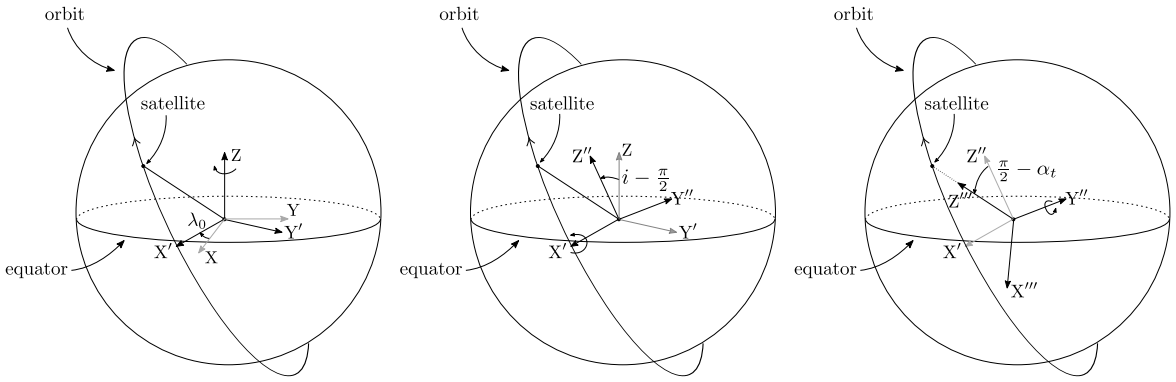


Figure 2.7 – Decomposition of the rotation that maps the axes of the Earth-centered inertial frame to the axes of the local orbital frame. The rotation is obtained by composing three *Givens* rotations about the three axes of the rotating frame. On each figure, the old axes are plotted in gray while the new axes are plotted in black. On the first figure, the inertial frame is rotated about its Z axis by λ_0. On the second figure, the new axes are rotated about X′ by $i - \frac{\pi}{2}$. On the last figure, the new axes are rotated about Y″ by $\frac{\pi}{2} - \alpha_t$.

where μ is the standard gravitational parameter of the Earth, thus T_S depends only on a. For typical Earth observation satellites with a roughly equal to 500 km, T_S is about one hour and a half.

Change of basis. The change of basis matrix from the rotational frame to the local orbital frame depends both on the orbit parameters and the satellite position. As the rotational frame and the satellite position depend on the time t, this change of basis matrix depends on t. It is a simple rotation composed with a translation. Its expression is derived in the next two paragraphs.

From inertial to orbital frame. Given a satellite position α_t and a triplet of orbit parameters (λ_0, i, a), the change of basis is obtained by first rotating the inertial frame about its Z axis by λ_0. This makes the new X axis, X′, point to the reference node, as shown on Figure 2.7. The new frame is then rotated about X′ to send the Z axis in the orbital plane. As the Z axis is perpendicular to the equatorial plane, the angle between the X′Z plane and the orbital plane is $i - \frac{\pi}{2}$. A third rotation by $\frac{\pi}{2} - \alpha_t$ about the new Y axis, Y″, makes the Z axis point towards the satellite. An additional rotation of π about the same axis makes the Z axis point towards the Earth center. The change of basis matrix is obtained by composing these three rotations:

$$\mathsf{P}_{\text{inertial, orbital}}(\lambda_0, i, \alpha_t) = \begin{bmatrix} c_{\lambda_0} & -s_{\lambda_0} & 0 \\ s_{\lambda_0} & c_{\lambda_0} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{i-\frac{\pi}{2}} & -s_{i-\frac{\pi}{2}} \\ 0 & s_{i-\frac{\pi}{2}} & c_{i-\frac{\pi}{2}} \end{bmatrix} \begin{bmatrix} c_{-\alpha_t-\frac{\pi}{2}} & 0 & s_{-\alpha_t-\frac{\pi}{2}} \\ 0 & 1 & 0 \\ -s_{-\alpha_t-\frac{\pi}{2}} & 0 & c_{-\alpha_t-\frac{\pi}{2}} \end{bmatrix} \tag{2.5}$$

From rotational to inertial frame. The change of basis at time t is a rotation of angle $\tau = 2\pi\frac{t}{T_E}$, where T_E denotes the duration of a stellar day, which is the Earth rotation

period relative to the fixed stars. It is approximately 86 164.10 s. The axis of the rotation is Z, thus the change of basis matrix is

$$\mathtt{P}_{\text{rotational, inertial}}(t) = \begin{bmatrix} c_\tau & -s_\tau & 0 \\ s_\tau & c_\tau & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.6}$$

The change of basis matrix from rotational to orbital frame is obtained by composing the previous matrices:

$$\begin{aligned} \mathtt{P}_{\text{rotational, orbital}}(t, \lambda_0, i, \alpha_0) &= \mathtt{P}_{\text{rotational, inertial}}(t)\mathtt{P}_{\text{inertial, orbital}}(\lambda_0, i, \alpha_0 + 2\pi\frac{t}{T_S}) \\ &= \begin{bmatrix} c_{\tau+\lambda_0} & -s_{\tau+\lambda_0} & 0 \\ s_{\tau+\lambda_0} & c_{\tau+\lambda_0} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{i-\frac{\pi}{2}} & -s_{i-\frac{\pi}{2}} \\ 0 & s_{i-\frac{\pi}{2}} & c_{i-\frac{\pi}{2}} \end{bmatrix} \begin{bmatrix} c_{-\alpha_t-\frac{\pi}{2}} & 0 & s_{-\alpha_t-\frac{\pi}{2}} \\ 0 & 1 & 0 \\ -s_{-\alpha_t-\frac{\pi}{2}} & 0 & c_{-\alpha_t-\frac{\pi}{2}} \end{bmatrix}, \end{aligned} \tag{2.7}$$

where for readability α_t is used to denote $\alpha_0 + 2\pi\frac{t}{T_S}$ in the last matrix (see formula 2.3).

Result 3. *Let $\mathbf{x} = (x, y)^\top$ be an image point. In the Earth centered rotational frame at time $t = x\delta_t$, the inverse image of $\{\mathbf{x}\}$ by the pushbroom camera mapping is the 3-space line passing through the point*

$$\mathbf{S} = -\mathtt{P}_{\text{rotational, orbital}}(t, \lambda_0, i, \alpha_0) \begin{pmatrix} 0 \\ 0 \\ R_E + a \end{pmatrix} \tag{2.8}$$

and directed by the vector

$$\mathbf{u} = \mathtt{P}_{\text{rotational, orbital}}(t, \lambda_0, i, \alpha_0)\mathtt{R}(\varphi(t), \psi(t), \omega(t)) \begin{pmatrix} 0 \\ w(y - y_0) \\ f \end{pmatrix}. \tag{2.9}$$

Proof. Let us denote by $\mathbf{S}$ the center of mass of the satellite. From Result 2 we know that the inverse image of $\{\mathbf{x}\}$ is the 3-space line passing through $\mathbf{S}$ and directed by a vector whose coordinates in the orbital frame are

$$\mathtt{R}(\varphi(t), \psi(t), \omega(t))(0, w(y - y_0), f)^\top. \tag{2.10}$$

The coordinates of that vector in the rotational frame are obtained by composing with the change of basis matrix $\mathtt{P}_{\text{rotational, orbital}}(t, \lambda_0, i, \alpha_0)$, which leads to the expression of $\mathbf{u}$ given in (2.9). If we denote by $\mathbf{E}$ the Earth's center of mass, then $\mathbf{S}-\mathbf{E}$ is a vector whose coordinates in the orbital frame are $-(0, 0, R_E + a)^\top$. Thus its coordinates in the rotational frame are given by

$$-\mathtt{P}_{\text{rotational, orbital}}(t, \lambda_0, i, \alpha_0)(0, 0, R_E + a)^\top. \tag{2.11}$$

As $\mathbf{E}$ is the origin of the rotational frame, the coordinates of $\mathbf{S}$ in that frame are the same. □

2.2.4 Image Formation Model

Localization function. Given an image point $\mathbf{x}$ and the altitude h above the sea level of the associated 3-space point $\mathbf{X}$, it is now possible to compute the geographic coordinates of $\mathbf{X}$ (see Definition 4). This is done by computing the first intersection point of the line defined in Result 3 with the sphere of radius $R_E + h$ centered at the Earth center.

The line defined in Result 3 passes through the point $\mathbf{S}$ and is directed by the vector $\mathbf{u}$ given by expression (2.9). The first intersection point with the sphere is computed by finding a scalar r such that

$$\|\mathbf{S} + r\mathbf{u}\|^2 = (R_E + h)^2. \tag{2.12}$$

This is a polynomial equation on r of degree 2. When the camera points towards the Earth, this equation has two positive roots, and the smallest root

$$r = \frac{-\mathbf{S}^\top\mathbf{u} - \sqrt{(\mathbf{S}^\top\mathbf{u})^2 - \mathbf{u}^\top\mathbf{u}(\mathbf{S}^\top\mathbf{S} - (R_E + h)^2)}}{\mathbf{u}^\top\mathbf{u}} \tag{2.13}$$

gives the position of the desired 3-space point $\mathbf{X} = \mathbf{S} + r\mathbf{u}$. This defines the *localization* function

$$\begin{aligned} L : \mathbf{R}^2 \times \mathbf{R} &\to [-\pi, \pi] \times [-\frac{\pi}{2}, \frac{\pi}{2}] \\ (\mathbf{x}, h) &\mapsto (\lambda, \theta), \end{aligned} \tag{2.14}$$

which associates a longitude λ and a latitude θ to each image point $\mathbf{x}$, provided that the altitude h is known. Here λ and θ are the geographic coordinates of the point $\mathbf{X} = \mathbf{S} + r\mathbf{u}$.

Projection function. Conversely, given a 3-space point $\mathbf{X}$ it is possible to compute the coordinates of its image $\mathbf{x}$. As there is no direct way of computing the time t at which the view plane will sweep through $\mathbf{X}$, this computation is not as straightforward as the previous one. This is a notable difference with the pinhole camera model for which the computation of image coordinates is a linear operation on homogeneous coordinates of the 3-space point. The *projection function*

$$\begin{aligned} P : [-\pi, \pi] \times [-\frac{\pi}{2}, \frac{\pi}{2}] \times \mathbf{R} &\to \mathbf{R}^2 \\ (\lambda, \theta, h) &\mapsto \mathbf{x} \end{aligned} \tag{2.15}$$

is the inverse of L with respect to the image coordinates. It is estimated iteratively from L. An initial estimate is computed by assuming that the dependence with h is linear and the dependence with (λ, θ) is quadratic. This initial estimate is then refined using the Newton algorithm.

A first estimation P_0 of the projection function P is made for a fixed altitude h. The estimation is done by assuming that the function $P(\cdot, \cdot, h)$ is quadratic. A 2×6 matrix $\mathtt{P}$ is estimated such that for all λ, θ

$$P_0(\lambda, \theta, h) = \mathtt{P}(1, \lambda, \theta, \lambda^2, \theta^2, \lambda\theta)^\top. \tag{2.16}$$

The matrix $\mathtt{P}$ is obtained through least squares minimization from a list of correspondences

$$(x_i, y_i) \leftrightarrow (\lambda_i, \theta_i) = L(x_i, y_i, h) \tag{2.17}$$

obtained by evaluating the localization function L on the vertices of a regular sampling grid $(x_i, y_i)_{i=1,\dots,n^2}$ of size $n \times n$ in the image domain (usually $n = 5$ is enough). The two rows $\mathbf{P^1}^\top$ and $\mathbf{P^2}^\top$ of $\mathtt{P}$ are computed by least-squares minimization of respectively $\|\mathtt{A}\mathbf{p} - \mathbf{b}_1\|$ and $\|\mathtt{A}\mathbf{p} - \mathbf{b}_2\|$, where

$$\mathtt{A} = \begin{bmatrix} 1 & \lambda_1 & \theta_1 & \lambda_1^2 & \theta_1^2 & \lambda_1\theta_1 \\ & & & \vdots & & \\ 1 & \lambda_{n^2} & \theta_{n^2} & \lambda_{n^2}^2 & \theta_{n^2}^2 & \lambda_{n^2}\theta_{n^2} \end{bmatrix}, \qquad \mathbf{b}_1 = \begin{pmatrix} x_1 \\ \vdots \\ x_{n^2} \end{pmatrix}, \quad \text{and} \quad \mathbf{b}_2 = \begin{pmatrix} y_1 \\ \vdots \\ y_{n^2} \end{pmatrix}. \tag{2.18}$$

The previous least-squares estimation is carried out for two different altitudes h_1 and h_2, leading to two matrices $\mathtt{P}_1$ and $\mathtt{P}_2$. These two altitudes can be, for example, lower and upper bounds for the 3D points of the scene. The projection function estimate P_0 is finally obtained from linear interpolation on h:

$$P_0(\lambda, \theta, h) = \left(\frac{|h_2 - h|}{|h_2 - h_1|}\mathtt{P}_1 + \frac{|h - h_1|}{|h_2 - h_1|}\mathtt{P}_2 \right) (1, \lambda, \theta, \lambda^2, \theta^2, \lambda\theta)^\top \tag{2.19}$$

Note that the estimation of $\mathtt{P}_1$ and $\mathtt{P}_2$ is done only once, while the formula (2.19) has to be evaluated each time we want to evaluate the projection function. The initial estimate P_0 is then refined using the Newton method [Press et al. 2007], as detailed in Algorithm 1. At each iteration, the function $(x, y) \mapsto L(x, y, h)$ is approximated by its first order Taylor expansion. The inversion of its Jacobian matrix, denoted by $J_{x,y}L(x, y, h)$, allows to refine the estimation. Note that the partial derivatives needed to compute the Jacobian matrix are approximated by forward differences. For example:

$$\frac{\partial L}{\partial x}(x, y, h) \approx L(x + 1, y, h) - L(x, y, h). \tag{2.20}$$

The loop stops when the function $L(\cdot, \cdot, h)$ maps the refined image coordinates to a point on Earth which is less than 1 mm from the input. This distance is computed using the fact that the squared distance between two points with spherical coordinates (λ, θ) and $(\tilde{\lambda}, \tilde{\theta})$ on the unit sphere is approximated to first order by $(\tilde{\theta} - \theta)^2 + (\tilde{\lambda} - \lambda)^2 \cos^2\theta$ when $(\tilde{\lambda}, \tilde{\theta})$ is close to (λ, θ).

The projection and localization functions depend obviously on the values of the camera parameters. For the sake of simplicity, this dependence is not made explicit in formula (2.14) but this may be done if needed by renaming L to $L_{\delta_t, w, f, y_0, a, i, \lambda_0, \alpha_0, \Phi}$. The complete set of 20 parameters on which L depends (12 of them are grouped together into Φ), plus the duration of the acquisition Δ_t, is summarized in Table 2.1.

2.3 Pushbroom Camera Parameters Refinement

The computation of the camera parameters, both external and internal, from a set of world to image correspondences is a problem known as *resectioning* [Hartley and Zisserman 2004]. In the case of remote sensing images, the internal parameters are usually known with a good accuracy because the sensor has been carefully calibrated before the launch of the satellite. Moreover the sensor calibration may have been refined during the in-flight commissioning, as happened for Pléiades [de Lussy et al. 2012]. In this chapter the internal parameters are thus assumed to be perfectly known.

Algorithm 1: Iterative estimation of the projection function $P : (\lambda, \theta, h) \mapsto (x, y)$.

Input: λ, θ, h: geographic coordinates and altitude of a 3D point
n_{iter}: number of iterations
Output: x, y: image coordinates
begin
1 $x, y = P_0(\lambda, \theta, h)$ *// initialization*
2 **for** $k \in \{1 \dots n_{\text{iter}}\}$ **do**
3 $\tilde{\lambda}, \tilde{\theta} = L(x, y, h)$
4 **if** $R_E \sqrt{(\tilde{\theta} - \theta)^2 + (\tilde{\lambda} - \lambda)^2 \cos^2 \theta} < 1\,\text{mm}$ *// error check*
5 **then**
6 break
7 $\begin{pmatrix} x \\ y \end{pmatrix} \leftarrow \begin{pmatrix} x \\ y \end{pmatrix} - [J_{x,y} L(x, y, h)]^{-1} \begin{pmatrix} \tilde{\lambda} - \lambda \\ \tilde{\theta} - \theta \end{pmatrix}$ *// update*

2.3.1 Camera Position Parameters

On modern high resolution satellites, the external parameters are measured on board in real time and provided with the images. The attitude parameters are more sensitive to measurement inaccuracies than the position parameters: for the typical case of a nadir looking satellite orbiting at 700 km, an error of 1 µrad on the roll or pitch angles yields a displacement of about 70 cm on the ground, while an error of 1 cm on the camera position yields a displacement of at most 1 cm on the ground. The accuracy of the attitude is much more critical than that of the camera position. On Pléiades the orbit parameters, i.e. the satellite positions, are computed on board with centimeter accuracy by two DORIS (Doppler Orbitography and Radiopositioning Integrated by Satellite) positioning instruments [de Lussy et al. 2012]. Note that satellite trajectories are subject to much less perturbations than airplanes trajectories.

Moreover, in terms of ground coordinates, attitude and position errors have similar effects and are therefore difficult to distinguish from each other. A roll or pitch error has locally the same effect as a translation of the focal plane, which can be obtained through a translation of the camera center. This observation was already reported by Gugan [1987], Konecny et al. [1987], and Kratky [1989b] in the late Eighties about SPOT-1 orientation parameters.

Therefore we make the extra assumption that the camera positions (i.e. the path followed in 3-space by the optical center of the camera) are perfectly known, and that all the measurement inaccuracies lie in the attitude angles.

2.3.2 Camera Attitude Parameters

The problem tackled in this section is to estimate the values of the 12 attitude parameters $\Phi = (\varphi_k, \psi_k, \omega_k)_{k=0,\dots,3}$, given a set of correspondences between image and world points. To emphasize the fact that only the attitude parameters are being estimated, the localization and projection functions L and P introduced in Section 2.2.4 are denoted by L_Φ and P_Φ.

	Parameter	Description	Pléiades	WorldView-2
internals	δ_t	dwell time	0.07 ms	?
	w	pixel width	13 µm	8 µm
	f	focal length	12.9 m	13.3 m
	y_0	principal point coordinate	15000 px	17500 px
externals	a	orbit altitude	694 km	770 km
	i	orbit inclination	98.2°	98.5°
	λ_0	reference node longitude	−180° - 180°	−180° - 180°
	α_0	initial angular position on the orbit	0° - 360°	0° - 360°
	Δ_t	duration of the acquisition	≈ 3 s	≈ 3 s
	$\Phi = (\varphi_k, \psi_k, \omega_k)_{k=0,\dots,3}$	coefficients of the attitude functions		

Table 2.1 – Parameters of the pushbroom camera model. The last two columns give typical numerical values for Pléiades and WorldView-2 satellites. The parameters may be divided into two classes: *internal parameters*, sometimes called *intrinsics*, which characterize the camera, and *external parameters*, or *extrinsics*, which characterize the position and orientation of the camera with respect to the inertial frame (see Definition 3).

Yaw versus Roll and Pitch

If we assume that the satellite is at nadir, i.e. with zero roll and pitch angles, the effect of attitude errors on the localization function L_Φ can be computed simply, as shown on Figure 2.8. To first order, a small error ε on the roll or pitch angles induces a displacement of $a\varepsilon$ on the ground, where a is the distance between the satellite and the ground. Things are different for the yaw, as it induces a rotation of the imaging sensor around its center. A small error ε results in a displacement of at most $\frac{D}{2}\varepsilon$ on the ground, where D is the swath width of the satellite. The relative importance of yaw errors with respect to roll and pitch errors is thus $\frac{D}{2a}$, which is generally less than 2%. For example with Pléiades $D = 20\,\text{km}$ and $a = 694\,\text{km}$, the ratio is approximately 1.4%.

It is thus reasonable to neglect the yaw error and assume that the yaw coefficients are perfectly known. The problem is thus narrowed down to estimating the value of the roll and pitch coefficients given a set of world to image correspondences. For simplicity of notation, the reduced set of 8 attitude parameters $(\varphi_k, \psi_k)_{k=0,\dots,3}$ is still denoted by Φ.

Global Energy Minimization

Assume $n \geq 1$ point correspondences $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ between 3D points $\mathbf{X}_i$ and 2D image points $\mathbf{x}_i = (x_i, y_i)$ are given. x_i and y_i are the row and column indices of the image points, meaning that $x_i\delta_t$ is the acquisition time and y_i is the column index, i.e. the position on the pushbroom array. When the attitude coefficients Φ are exact, the line $L_\Phi(\mathbf{x}_i, \cdot)$ passes through the 3-space point $\mathbf{X}_i$ and the energy E defined by

$$E(\Phi) = \frac{1}{n}\sum_{i=1}^{n} d(\mathbf{X}_i, L_\Phi(\mathbf{x}_i, \cdot))^2, \tag{2.21}$$

is zero. Here $d(\mathbf{X}, \ell)$ denotes the Euclidean distance between the 3-space point $\mathbf{X}$ and the line ℓ. Note that even if the line $L_\Phi(\mathbf{x}_i, \cdot)$ can be thought of as the set of points $L_\Phi(\mathbf{x}_i, h)$

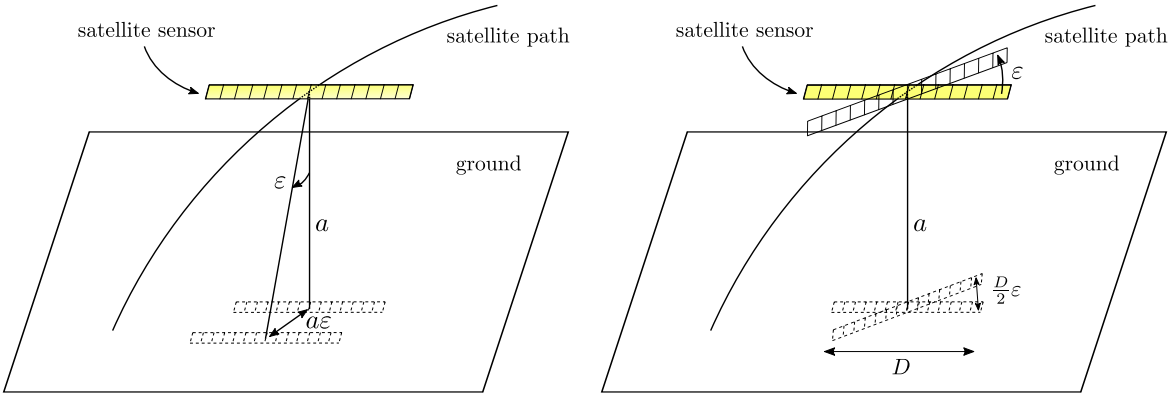


Figure 2.8 – Effect of attitude errors on the localization function. The figure on the left shows a pitch error of ε. To first order, this induces a ground displacement of the sensor projection of $a\varepsilon$, where a is the flying altitude of the sensor. The figure on the right shows a yaw error of ε. It induces a maximal ground displacement of $\frac{D}{2}\varepsilon$, obtained for the sensor endpoints, where D denotes the swath width.

obtained by evaluating L_Φ on $(\mathbf{x}_i, h)$ for h varying in $\mathbf{R}$, the line is entirely defined by a point and a direction (given in Result 3). Hence the evaluation of $L_\Phi(\mathbf{x}_i, h)$ for any particular h is not needed. The attitude parameters can then be estimated by

$$\hat{\Phi} = \underset{B(\Phi,\eta)}{\arg\min}\, E, \tag{2.22}$$

where Φ is the initial estimate measured on board, $B(\Phi, \eta)$ is the ball centered in Φ with radius η, and η is given by the nominal precision of the satellite attitudes measurement system (typically 50 µrad, i.e. around 30 m on the ground).

The cost expressed in formula (2.21) involves the distance between 3-space points $\mathbf{X}_i$ and rays back-projected from image points $\mathbf{x}_i$. Another option could be to minimize the distance, in the image plane, between image points $\mathbf{x}_i$ and projected points $P_\Phi(\mathbf{X}_i)$:

$$E'(\Phi) = \frac{1}{n}\sum_{i=1}^{n} d(\mathbf{x}_i, P_\Phi(\mathbf{X}_i))^2, \tag{2.23}$$

This would require to evaluate the projection function P_Φ instead of the localization function L_Φ, which, as reported by Gugan [1987] and Kratky [1989a], is more complex computationally since each evaluation of P_Φ requires several evaluations of L_Φ and its derivatives (see Algorithm 1). For solving the general camera resectioning problem, energy (2.23) is usually preferred to energy (2.21) because in energy (2.21) the points located far away from the camera have more importance than the points located close to the camera. But in the case of remote sensing images all the distances between points are small compared to the distance separating the camera from the scene. The two energies can thus be considered as equivalent.

Pointwise Minimization

An alternative approach to global minimization is to compute, for any correspondence $\mathbf{X} \leftrightarrow \mathbf{x}$, the instantaneous roll and pitch angles $\varphi(t)$ and $\psi(t)$ needed to make the back-projected ray

from $\mathbf{x}$, namely $L(\mathbf{x}, \cdot)$, pass through $\mathbf{X}$. The direction of this particular back-projected ray depends only on the values of the roll and pitch angles at time $t = x\delta_t$, not on the whole set of coefficients Φ. The coefficients in Φ are then estimated by polynomial fitting from the n computed roll and pitch values.

Each world to image correspondence imposes instantaneous roll and pitch values. Let us consider a single correspondence between the image point $\mathbf{x} = (x, y)$ and the world point $\mathbf{X}$. Knowing the instantaneous yaw $\omega(t)$ at time $t = x\delta_t$, we look for roll and pitch values φ and ψ such that the ray back-projected from $\mathbf{x}$ passes through $\mathbf{X}$. Assume that the coordinates of $\mathbf{X}$ are expressed in the local orbital frame at time t (the conversion from geographic coordinates to local orbital coordinates is done thanks to formula 2.7). Using the direction of the ray back-projected from $\mathbf{x}$ given in Result 2, our problem amounts to find φ and ψ such that $\mathtt{R}(\varphi, \psi, \omega(t))(0, w(y-y_0), f)^\top$ has the same direction as $\mathbf{X}$. The factorization of $\mathtt{R}(\varphi, \psi, \omega)$ given in equation (2.1) allows to separate the known yaw term from the unknown roll and pitch: $\mathtt{R}(\varphi, \psi, \omega) = \mathtt{R}(\varphi, \psi, 0)\mathtt{R}(0, 0, \omega)$. By denoting

$$\mathbf{u} = \mathtt{R}(0, 0, \omega(t)) \begin{pmatrix} 0 \\ w(y - y_0) \\ f \end{pmatrix} = \begin{bmatrix} c_{\omega(t)} & -s_{\omega(t)} & 0 \\ s_{\omega(t)} & c_{\omega(t)} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} 0 \\ w(y - y_0) \\ f \end{pmatrix}, \tag{2.24}$$

the problem becomes finding φ and ψ such that

$$\mathtt{R}(\varphi, \psi, 0)\mathbf{u} \propto \mathbf{X}. \tag{2.25}$$

With normalized coordinates for $\mathbf{u}$ and $\mathbf{X}$ the proportionality constraint is replaced by an equality. The angles φ and ψ are obviously bounded to (at least) $[-\frac{\pi}{2}, \frac{\pi}{2}]$, otherwise the camera would not be oriented toward the Earth. Another reasonable assumption is that the Z coordinate of $\mathbf{u}$ and $\mathbf{X}$ is much bigger than both their X and Y coordinates. Even the most agile satellites such as Pléiades and WorldView-3 do not usually point more than 30° away from nadir. With that assumption our problem is then completely solved by the next result.

Result 4. *Let* $\mathbf{u} = (u_1, u_2, u_3)^\top$ *and* $\mathbf{v} = (v_1, v_2, v_3)^\top$ *be two points on the unit sphere* S^2*, such that* $u_3 > |u_1| + |v_1|\sqrt{2}$ *and* $v_3 > |v_2| + |u_2|\sqrt{2}$*. Then there exists a unique pair of angles* $(\varphi, \psi) \in [-\frac{\pi}{4}, \frac{\pi}{4}]^2$ *such that*

$$\mathtt{R}(\varphi, \psi, 0)\mathbf{u} = \mathbf{v}, \tag{2.26}$$

where $\mathtt{R}(\varphi, \psi, 0)$ *is the rotation obtained by composing two* Givens *rotations about the* X *and* Y *axis as defined in equation* (2.1)*.*

Proof. Using the factorization of equation (2.1), equation (2.26) becomes

$$\begin{bmatrix} c_\psi & 0 & s_\psi \\ 0 & 1 & 0 \\ -s_\psi & 0 & c_\psi \end{bmatrix} \mathbf{u} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\varphi & s_\varphi \\ 0 & -s_\varphi & c_\varphi \end{bmatrix} \mathbf{v}, \tag{2.27}$$

which is equivalent to the following system of equations:

$$\begin{aligned} u_1 \cos\psi + u_3 \sin\psi &= v_1, \\ v_2 \cos\varphi + v_3 \sin\varphi &= u_2, \\ -u_1 \sin\psi + u_3 \cos\psi &= -v_2 \sin\varphi + v_3 \cos\varphi. \end{aligned} \tag{2.28}$$

Lemma 1, stated below, shows that each of the first two independent equations has a unique solution on $[-\frac{\pi}{4}, \frac{\pi}{4}]$. Let us show that these solutions ψ and φ satisfy the third equation. As $\mathbf{u}$ and $\mathbf{v}$ lie on the unit sphere, one can find (α, β) and (γ, δ) in $[-\pi, \pi] \times [-\frac{\pi}{2}, \frac{\pi}{2}]$ such that

$$\begin{aligned} u_1 &= \cos\beta\cos\alpha & v_2 &= \cos\delta\cos\gamma, \\ u_3 &= \cos\beta\sin\alpha & v_3 &= \cos\delta\sin\gamma, \\ u_2 &= \sin\beta & v_1 &= \sin\delta. \end{aligned} \tag{2.29}$$

α and β (resp. γ and δ) are spherical coordinates for $\mathbf{u}$ (resp. $\mathbf{v}$). As $u_3 > |u_1|$ and $v_3 > |v_2|$, one can require additionally that α and γ lie in $]\frac{\pi}{4}, \frac{3\pi}{4}[$. With these notations the system (2.28) is rewritten as

$$\begin{aligned} \cos\beta\cos(\alpha - \psi) &= \sin\delta, \\ \cos\delta\cos(\gamma - \varphi) &= \sin\beta, \\ \cos\beta\sin(\alpha - \psi) &= \cos\delta\sin(\gamma - \varphi). \end{aligned} \tag{2.30}$$

The square of the left term of the third equation is

$$\cos^2\beta\sin^2(\alpha-\psi) = \cos^2\beta(1-\cos^2(\alpha-\psi)) = \cos^2\beta - \cos^2\beta\cos^2(\alpha-\psi) = \cos^2\beta - \sin^2\delta, \tag{2.31}$$

where the last equality comes from the first equation of (2.30). Similarly, using the second equation, the square of the right term is

$$\cos^2\delta\sin^2(\gamma-\varphi) = \cos^2\delta(1-\cos^2(\gamma-\varphi)) = \cos^2\delta - \cos^2\delta\cos^2(\gamma-\varphi) = \cos^2\delta - \sin^2\beta, \tag{2.32}$$

hence the two terms have the same squares. As β is in $[-\frac{\pi}{2}, \frac{\pi}{2}]$, $\cos\beta$ is non-negative. Moreover, as $\psi \in [-\frac{\pi}{4}, \frac{\pi}{4}]$ and $\alpha \in]\frac{\pi}{4}, \frac{3\pi}{4}[$, it is clear that $0 < \alpha - \psi < \pi$ hence $\sin(\alpha - \psi) > 0$. Similarly, $\cos\delta$ and $\sin(\gamma - \varphi)$ are non-negative thus the equality between squares suffices to prove the desired result. □

Lemma 1. *Let a, b, c be three real numbers such that $|a| + |c|\sqrt{2} < b$. Then the equation*

$$a\cos x + b\sin x + c = 0 \tag{2.33}$$

has a unique solution on $[-\frac{\pi}{4}, \frac{\pi}{4}]$.

Proof. Let $f : [-\frac{\pi}{4}, \frac{\pi}{4}] \to \mathbf{R}$ be the mapping $x \mapsto a\cos x + b\sin x + c$. This function is continuous and has values of opposite sign at the endpoints of its domain interval, since

$$f\left(-\frac{\pi}{4}\right) f\left(\frac{\pi}{4}\right) = \left(\frac{a-b}{\sqrt{2}} + c\right)\left(\frac{a+b}{\sqrt{2}} + c\right) = \left(\frac{a}{\sqrt{2}} + c\right)^2 - \frac{b^2}{2} < 0. \tag{2.34}$$

Thus from the intermediate value theorem [Rudin 1976] f has a root in $[-\frac{\pi}{4}, \frac{\pi}{4}]$. To prove the uniqueness, the case $a = 0$ being trivial, one can assume that $a > 0$ (or otherwise use $-f$). The derivative of f is $x \mapsto -a\sin x + b\cos x$. It has a unique zero and changes sign from positive to negative in $\arctan\frac{b}{a}$, which is outside the domain $[-\frac{\pi}{4}, \frac{\pi}{4}]$ since $\frac{b}{a} > 1$. Thus f is strictly increasing and the root is unique. □

In practice the solution to equation (2.33), and hence to equations (2.28) and (2.26), is computed thanks to the change of variable $t = \sin x$. If x is a solution then t satisfies the quadratic equation

$$(a^2 + b^2)t^2 + 2bct + c^2 - a^2 = 0 \tag{2.35}$$

which has two distinct solutions since its discriminant has the sign of $a^2 + b^2 - c^2$ and we assumed that $c^2 < b^2$. The unique solution to equation (2.33) is found by testing these two candidates. The procedure to compute φ and ψ from a world to image correspondence is summarized in Algorithm 2.

Algorithm 2: Computation of roll and pitch values from a world to image correspondence.

Input: $\mathbf{X}$: Cartesian coordinates of a control point in the Earth-centered rotational frame
x, y: image coordinates
δ_t, w, f, y_0: camera internal parameters
a, i, λ_0: orbit parameters
α_0: satellite initial position on the orbit
ω: yaw function

Output: $\varphi, \psi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$: roll and pitch values

begin

$\mathbf{v} \leftarrow \mathtt{P}_{\text{rotational, orbital}}(x\delta_t, \lambda_0, i, \alpha_0)^{-1}\mathbf{X} + (0, 0, R_E + a)$ *// conversion to orbital coordinates*

$\mathbf{v} \leftarrow \frac{\mathbf{v}}{\|\mathbf{v}\|}$

$\mathbf{u} \leftarrow \mathtt{R}(0, 0, \omega(x\delta_t))(0, w(y - y_0), f)^\top$ *// yaw application*

$\mathbf{u} \leftarrow \frac{\mathbf{u}}{\|\mathbf{u}\|}$

if $u_3 < |u_1|+|v_1|\sqrt{2}$ or $v_3 < |v_2|+|u_2|\sqrt{2}$ **then**

  break *// the ground control point is unusable*

else

  solve $\mathtt{R}(\varphi, \psi, 0)\mathbf{u} = \mathbf{v}$ for φ and ψ *// using result 4*

return φ, ψ

Polynomial fitting of roll and pitch error evolution. Once a list of roll and pitch samples has been computed from the set of world to image correspondences thanks to Algorithm 2, the polynomial coefficients of the roll and pitch functions are estimated by least-squares minimization.

More precisely, assume that a list of roll samples $\hat{\varphi}^1, \ldots, \hat{\varphi}^n$ has been computed at times $t_1, \ldots, t_n$ from the world to image correspondences. If we denote by φ the initial roll function (the one derived from the onboard attitude measurements), then the roll correction function is estimated by fitting a polynomial of degree 3 to the points $t_i, \hat{\varphi}^i - \varphi(t_i)$ in $\mathbf{R}^2$. As the initial roll function has a known accuracy η, given by the satellite specifications, any true roll value cannot differ from $\varphi(t)$ more than η. The samples $t_i, \hat{\varphi}^i$ such that $|\hat{\varphi}^i - \varphi(t_i)| > \eta$ are thus discarded. Moreover the fitted correction polynomial must be bounded by η. Its

coefficients $\mathbf{p} = (p_0, p_1, p_2, p_3)^\top$ are estimated by solving a constrained least squares problem:

$$\begin{aligned} \underset{\mathbf{p}}{\text{minimize}} \quad & \|\mathtt{A}\mathbf{p} - \mathbf{b}\|^2 \\ \text{subject to} \quad & -\eta \leq \sum_i p_i t^i \leq \eta, \ \forall t \in [0, T]. \end{aligned} \tag{2.36}$$

with

$$\mathtt{A} = \begin{bmatrix} 1 & t_1 & t_1^2 & t_1^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & t_n & t_n^2 & t_n^3 \end{bmatrix} \text{ and } \mathbf{b} = \begin{pmatrix} \hat{\varphi}^1 - \varphi(t_1) \\ \vdots \\ \hat{\varphi}^n - \varphi(t_n) \end{pmatrix}. \tag{2.37}$$

The constraint $-\eta \leq \sum_i p_i t^i \leq \eta$, $\forall t \in [0, T]$ is approximated thanks to a finite sampling $0, \frac{T}{m}, \frac{2T}{m}, \ldots, \frac{mT}{m}$ of the interval $[0, T]$, by

$$-\eta \leq \sum_i p_i \left(\frac{kT}{m}\right)^i \leq \eta \quad \forall k \in \{0, \ldots, m\}. \tag{2.38}$$

This linear constraint is expressed as

$$-\mathbf{h} \preceq \mathtt{G}\mathbf{p} \preceq \mathbf{h}, \tag{2.39}$$

where $\mathbf{x} \preceq \mathbf{y}$ denotes componentwise inequality, i.e. $x_i \leq y_i$ for all i, while $\mathtt{G}$ is an $(m+1) \times 4$ matrix and $\mathbf{h}$ an $(m+1)$-vector defined by

$$\mathtt{G} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & \frac{T}{m} & \frac{T}{m}^2 & \frac{T}{m}^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & T & T^2 & T^3 \end{bmatrix} \text{ and } \mathbf{h} = \begin{pmatrix} \eta \\ \vdots \\ \eta \end{pmatrix}. \tag{2.40}$$

The optimization problem 2.36 is thus replaced by a quadratic program problem [Boyd and Vandenberghe 2004]:

$$\begin{aligned} \underset{\mathbf{p}}{\text{minimize}} \quad & \|\mathtt{A}\mathbf{p} - \mathbf{b}\|^2 \\ \text{subject to} \quad & -\mathbf{h} \preceq \mathtt{G}\mathbf{p} \preceq \mathbf{h}. \end{aligned} \tag{2.41}$$

This kind of convex optimization problem that includes inequality constraints can be solved by interior-point methods. These methods solve the problem by applying Newton's method to a sequence of equality constrained problems [Boyd and Vandenberghe 2004]. A quadratic program solver is available in the `cvxopt` package, which is a free software package for convex optimization based on the Python programming language. It is developed by Martin Andersen, Joachim Dahl and Lieven Vandenberghe. This solver implements a primal-dual path-following method based on the Nesterov-Todd scaling [Vandenberghe 2010, Andersen et al. 2003].

Note that for simplicity we detailed only the roll function estimation, but the method applies to pitch as well. The complete procedure to refine the roll and pitch functions is summarized in Algorithm 3.

Algorithm 3: Refinement of roll and pitch coefficients from world to image correspondences.

Input: $\{\mathbf{X}_i \leftrightarrow \mathbf{x}_i\}$: set of $n \geq 1$ world to image correspondences
δ_t, w, f, y_0: camera internal parameters
a, i, λ_0: orbit parameters
α_0: satellite initial position on the orbit
φ, ψ, ω: attitude functions measured on board
η: nominal attitude accuracy

Output: $\hat{\varphi}, \hat{\psi}$: roll and pitch refined functions

begin

1 **for** $k \in \{1 \dots n\}$ **do**
2 $t_i \leftarrow x_i \delta_t$
3 $\hat{\varphi}^i, \hat{\psi}^i \leftarrow \texttt{roll\_and\_pitch\_from\_gcp}(\mathbf{X}_i, \mathbf{x}_i, \delta_t, w, f, y_0, a, i, \lambda_0, \alpha_0, \omega)$ *// algorithm 2*
4 **if** $|\hat{\varphi}^i - \varphi(t_i)| > \eta$ or $|\hat{\psi}^i - \psi(t_i)| > \eta$ **then**
5 discard $\hat{\varphi}^i, \hat{\psi}^i, t_i, \mathbf{x}_i, \mathbf{X}_i$
6 $n \leftarrow n - 1$
7 $\mathtt{A} \leftarrow \text{Vandermonde}(t_1, \dots t_n)$ *// formula 2.37*
8 **for** $\theta \in \{\varphi, \psi\}$ **do**
9 $\mathbf{b} \leftarrow (\hat{\theta}^i - \theta(t_i))^\top$ *// formula 2.37*
10 $\underset{\mathbf{p}}{\text{minimize}} \ \|\mathtt{A}\mathbf{p} - \mathbf{b}\|^2$ subject to $-\mathbf{h} \preceq \mathtt{G}\mathbf{p} \preceq \mathbf{h}$ *// $\mathtt{G}$ and $\mathbf{h}$ are precomputed*
11 $\hat{\theta} \leftarrow \theta + \sum_i p_i t^i$
12 return $\hat{\varphi}, \hat{\psi}$

2.4 Empirical Validation

This section presents an empirical validation procedure for the attitude correction Algorithm 3. All the experiments are carried out on synthetic data, using the pushbroom camera model described in Section 2.2. The implementation of that model is a *geometric*, and not *radiometric*, simulator.

On the one hand, synthetic data permits to test any particular configuration, enabling a fine understanding of the effects of each parameter. Moreover the synthetic observations, i.e. the 3D and 2D coordinates of world to image correspondences, are generated from the image formation model using the camera parameters, hence are noiseless. Adding artificial noise permits to analyze and quantify the impact of noise on the refinement algorithm.

On the other hand, the camera model used in the present chapter is simplified: to mention just a few approximations, the orbit is supposed perfectly circular while real observation satellites fly on elliptical sun-synchronous orbits whose orbital plane precesses once a year, the Earth is supposed to be a sphere while a closer approximation would be an oblate spheroid, the light travel time from the Earth surface to the sensor is neglected, as well as the atmospheric refraction. Experiments on real data are obviously needed for a definitive validation of the proposed algorithm. This delicate task is left to a forthcoming article.

2.4.1 Experimental Setup

We use the same notational convention as Hartley and Zisserman [2004] in chapter 5 of their reference book:

- measured quantities are denoted by $\mathbf{x}$ (image point coordinates), $\mathbf{X}$ (ground point coordinates) or Φ (attitude coefficients),
- estimated quantities are represented by a hat, such as $\hat{\mathbf{x}}$, $\hat{\mathbf{X}}$ or $\hat{\Phi}$,
- true values are represented by a bar, such as $\bar{\mathbf{x}}$, $\bar{\mathbf{X}}$ or $\bar{\Phi}$.

The testing procedure begins by choosing a fixed set of values for the parameters describing the pushbroom camera (see Table 2.1), among which the true attitude coefficients $\bar{\Phi}$ lie. Next a set of image locations $\bar{\mathbf{x}}_i$ and associated altitudes $\bar{h}_i$ are chosen. The number of such locations will vary. Synthetic world to image correspondences $\bar{\mathbf{X}}_i \leftrightarrow \bar{\mathbf{x}}_i$ are generated thanks to the localization function $L_{\bar{\Phi}}$. The correspondence is exact, in the sense that $\bar{\mathbf{X}}_i = L_{\bar{\Phi}}(\bar{\mathbf{x}}_i, \bar{h}_i)$ precisely, up to machine accuracy.

Next, artificial Gaussian noise is added to the image and ground locations by perturbing both the x and y coordinates of the image points and the X, Y and Z coordinates of the ground control points by a zero-mean Gaussian random variable with known variance. The resulting noisy points are denoted by $\mathbf{X}_i$ and $\mathbf{x}_i$. A suitable Gaussian random number generator is given in [Press et al. 2007]. As the number of points is usually rather small, the experiment should be run many times with different realizations of the noise, and the results averaged. In the case of only one experiment, it makes more sense to impose the magnitude of the noise by choosing the perturbations uniformly on a sphere of known radius. This is what is implemented in Algorithm 4.

Next, the roll and pitch coefficients are perturbed. On real satellites, the attitude functions are interpolated from a list of regularly sampled attitude values. These attitude values are measured with star trackers and gyroscopes at fixed intervals. For examples, on WorldView-2, a state vector containing the time, position, velocity and attitude is recorded every 0.02 s [Poli and Toutin 2012]. On Pléiades, a state vector containing the time, position and velocity is recorded every 30 s during a timespan longer than the acquisition in order to provide around 10 samples. The attitudes are given directly as polynomial functions, but we can reasonably assume that they have been fitted from measured samples. To simulate this behavior, $d+1$ real numbers are generated from a zero-mean uniform random variable with known amplitude. A polynomial perturbation of degree at most d is interpolated from these $d+1$ values, regularly spaced at times $t = 0, \frac{\Delta_t}{d}, \frac{2\Delta_t}{d}, \ldots, \Delta_t$. This perturbation is added to the roll *true* function $\bar{\varphi}$ to produce the *measured* roll function φ. The same is done with the pitch. The resulting noisy attitude coefficients are denoted by Φ.

Then the roll and pitch coefficients are estimated from the noisy ground control points $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ and the noisy attitude coefficients Φ using the proposed refinement algorithm (see Algorithm 3). This results in the *estimated* functions $\hat{\varphi}$ and $\hat{\psi}$.

To assess the algorithm performance, the estimation errors $\hat{\varphi} - \bar{\varphi}$ and $\hat{\psi} - \bar{\psi}$ are compared to the measurement errors $\varphi - \bar{\varphi}$ and $\psi - \bar{\psi}$. In order to facilitate the comparison, the L^2 and L^∞ norms of the error functions are computed. These norms give the root mean square and the maximal errors. The same quantities are computed for the *localization error function* defined by

$$\begin{aligned} [0, \Delta_t] &\to \mathbf{R}_+ \\ t &\mapsto d(L_\Phi(t\delta_t, y_0, h_0), L_{\bar{\Phi}}(t\delta_t, y_0, h_0)), \end{aligned} \tag{2.42}$$

where y_0 is the principal point coordinate, $h_0 = \frac{1}{n}\sum_i \bar{h}_i$ is the mean altitude and $d(\mathbf{X}, \mathbf{X}')$ is the geodetic distance between two points on the surface of the Earth. As for φ and ψ, the estimation error (computed with $\hat{\Phi}$) is compared to the measurement error (computed with Φ as in (2.42)).

2.4.2 Online Demo

The experimental setup presented in Section 2.4.1 is made available to the reader through the online demo associated to this chapter[1].

To help the user with the fastidious task of setting the numerous parameters of the camera, a few predefined sets of parameters are proposed. They correspond to the internal and orbit parameters of well-known commercial satellites such as Pléiades and WorldView and are given in Table 2.1. The reference node longitude λ_0 and the initial angular position α_0 on the orbit change over time for any satellite, thus they should be chosen by the user of the demo. As these two parameters shown no influence on the results of the refinement algorithm, they are not exposed in the demo interface and have fixed values $\lambda_0 = 30°$ and $\alpha_0 = 180°$ for all the experiments.

As it would be difficult to choose manually a set of realistic values for the 12 attitude coefficients, the user only selects an initial pointing direction $\Psi_x, \Psi_y \in [-\frac{\pi}{4}, \frac{\pi}{4}]^2$ and a heading

[1] http://boucantrin.ovh.hw.ipol.im:8082/pushbroom_ba

$\gamma \in [0, 2\pi]$. The *heading* defines the desired direction of the pushbroom track on the ground, with respect to the local North direction. The attitude coefficients $\bar{\Phi}$ are then computed from Ψ_x, Ψ_y and γ thanks to the *guidance algorithm*, given in appendix A.

Next the user selects a number of image points $\bar{\mathbf{x}}_i$ by clicking on a white square representing the image domain. A random altitude $\bar{h}_i$ is associated to each image point.

Finally the user selects the standard deviations σ_{image} and σ_{world} of the Gaussian perturbations added to the coordinates of the image and world points, and selects the degree and amplitude of the polynomial perturbation added to the roll and pitch functions.

As a result of the simulation, the online demo shows the plots of the localization error function (see (2.42)) as well as the roll and pitch error functions, each computed with the measured data and with the estimated data. An example is given in Figure 2.9. The complete testing procedure implemented in the online demo is summarized in Algorithm 4.

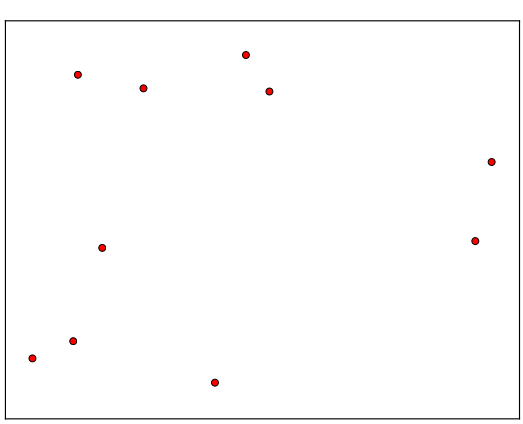
Algorithm 4: Testing procedure for the roll and pitch refinement algorithm.

Input: $\bar{\delta}_t, \bar{w}, \bar{f}, \bar{y}_0, \bar{a}, \bar{i}$: set of true internal and orbit parameters
$\bar{\lambda}_0, \bar{\alpha}_0$: true reference node longitude and initial angular position on the orbit
Ψ_x, Ψ_y, γ: initial pointing direction and heading
$\{\bar{\mathbf{x}}_k\}$: set of $n \geq 1$ true image locations
$\sigma_{\text{image}}, \sigma_{\text{world}}$: std dev of the Gaussian perturbations added to the correspondences
d, η: degree and amplitude of the perturbation added to the roll and pitch functions

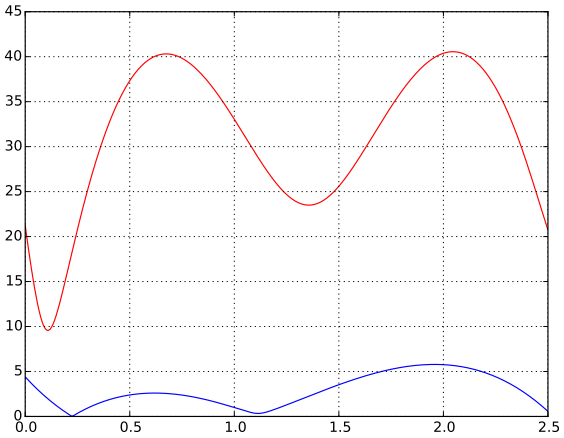
Output: statistics (rmse, max) of the roll, pitch and localization error functions, with measured and estimated attitudes

begin

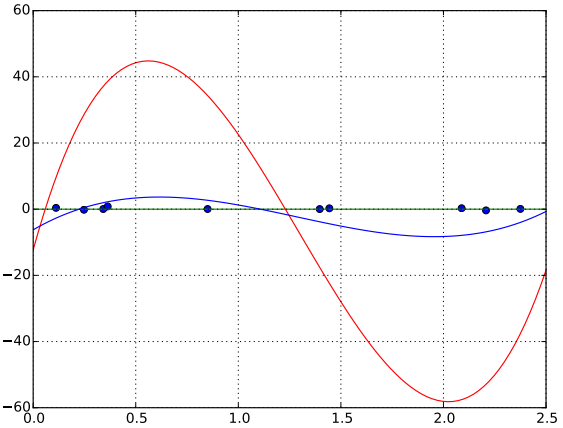
$\bar{\Phi} \leftarrow$ `guidance`$(\Psi_x, \Psi_y, \gamma, \bar{\delta}_t, \bar{w}, \bar{f}, \bar{y}_0, \bar{a}, \bar{i}, \bar{\lambda}_0, \bar{\alpha}_0)$
for $k \in \{1 \dots n\}$ **do**
  $\bar{h}_k \leftarrow$ random value between 0 and 1000
  $\bar{\mathbf{X}}_k \leftarrow L_{\bar{\Phi}}(\bar{\mathbf{x}}_k, \bar{h}_k)$
for $k \in \{1 \dots n\}$ **do**
  $\mathbf{u} \leftarrow$ random vector on the unit sphere S^2
  $\mathbf{v} \leftarrow$ random vector on the unit circle S^1
  $\mathbf{X}_k \leftarrow \bar{\mathbf{X}}_k + \sigma_{\text{world}}\mathbf{u}$
  $\mathbf{x}_k \leftarrow \bar{\mathbf{x}}_k + \sigma_{\text{image}}\mathbf{v}$
for $\theta \in \{\varphi, \psi\}$ **do**
  $\theta^0, \theta^1, \dots, \theta^d \leftarrow$ uniform random samples in $[-\eta, \eta]$
  $p \leftarrow$ Lagrange polynomial interpolating $(0, \theta^0), (\frac{\Delta_t}{d}, \theta^1), \dots, (\Delta_t, \theta^d)$
  $\theta \leftarrow \bar{\theta} + p$
$\hat{\varphi}, \hat{\psi} \leftarrow$ `roll_and_pitch_refinement`$(\{\mathbf{X}_i \leftrightarrow \mathbf{x}_i\}, \bar{\delta}_t, \bar{w}, \bar{f}, \bar{y}_0, \bar{a}, \bar{i}, \bar{\lambda}_0, \bar{\alpha}_0, \varphi, \psi, \bar{\omega}, \eta)$
for $\theta \in \{\varphi, \psi\}$ **do**
  return statistics of $\theta - \bar{\theta}$ and $\hat{\theta} - \bar{\theta}$



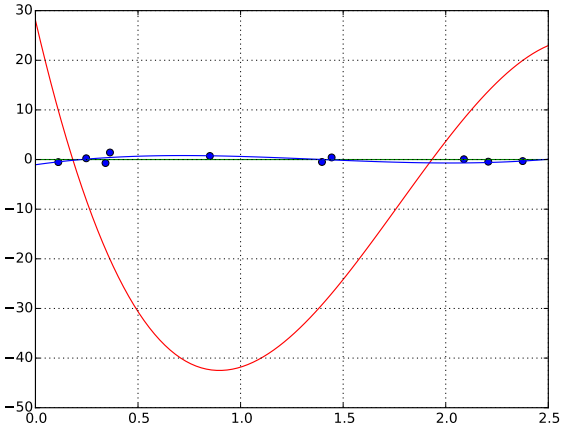
(a) Input image points



(b) Localization error in meters



(c) Roll error in micro radians



(d) Pitch error in micro radians

Figure 2.9 – Example of results produced with the online demo associated to this chapter. Red curves show the error before correction and blue curves after correction. The user clicked on the image domain to select the input image points shown in (a). The localization error (in meters) as a function of time (in seconds) is shown in (b). The red curve gives the error made using the measured attitudes, while the blue curve gives the error made using the estimated attitudes. The roll and pitch errors (in micro radians) as functions of time (in seconds) are shown on (c) and (d). The blue dots give the errors of the instantaneous roll and pitch samples computed from each world to image correspondence.

2.4.3 Numerical Results

The testing procedure described in Algorithm 4 was run with typical sets of camera parameters on various control points configurations and various roll and pitch errors. Results presented in Figure 2.10, page 53, show that for a given degree d of the roll and pitch polynomial errors, $d+1$ control points are sufficient to reduce the localization error by one order of magnitude.

However this conclusion holds only if the row indices of the control points are roughly allocated regularly over the whole row range. Two control points located on neighboring rows correspond to very close times t and t', thus the two constraints they provide for the roll and pitch fitting behave as only one constraint. This is illustrated by the results in Table 2.3, page 54.

Experiments with different amounts of noise added to the control points show that, as expected, the estimation is perfect in the absence of noise. The accuracy of the estimation reduces as soon as the noise amount increases, but this can be compensated by increasing the quantity of control points. These observations are illustrated by results presented in Table 2.4, page 55.

2.5 The Rational Polynomial Camera Model

Image vendors, such as Airbus Defence and Space [Coeurdevey and Gabriel-Robez 2012], DigitalGlobe and Skybox, provide usually the orientation parameters of the cameras along with the images. To save their customers the tedious task of understanding and implementing a specific geometric camera model, they provide directly the *localization* and *projection* functions L and P that we introduced in section 2.2.4.

These functions allow to convert from image coordinates to coordinates on the globe and back. The projection function $P : \mathbf{R}^3 \to \mathbf{R}^2$, $(\lambda, \theta, h) \mapsto \mathbf{x}$ returns the image coordinates, in pixels, of a given 3-space point represented by its spheroidal coordinates in the World Geodetic System (WGS 84). In that system a point of 3-space is identified by its longitude $\lambda \in [-180, 180]$, latitude $\theta \in [-90, 90]$ and altitude h, in meters, above the reference ellipsoid. The localization function $L : \mathbf{R}^3 \to \mathbf{R}^2$, $(\mathbf{x}, h) \mapsto (\lambda, \theta)$ is its inverse with respect to the first two components. It takes a point $\mathbf{x} = (x, y)^\top$ in the image domain together with an altitude h, and returns the geographic coordinates of the unique 3-space point $\mathbf{X} = (\lambda, \theta, h)$ whose altitude is h and whose image is $\mathbf{x}$.

The *Rational Polynomial Coefficient* (RPC) camera model is an analytic description of the projection and localization functions. It was introduced in the late eighties by Baltsavias and Stallmann [1992], and studied in depth by Tao and Hu [2001]. In the RPC model, the projection and localization functions are expressed as ratio of multivariate cubic polynomials. For example, the latitude component of the localization function for the image point (x, y) at altitude h is

$$\theta = \frac{\sum_{i=1}^{20} C_i^{\theta,\mathrm{NUM}} \rho_i(x, y, h)}{\sum_{i=1}^{20} C_i^{\theta,\mathrm{DEN}} \rho_i(x, y, h)}, \tag{2.43}$$

where $C_i^{\theta,\mathrm{NUM}}$ (resp. $C_i^{\theta,\mathrm{DEN}}$) is the i^{th} coefficient of the numerator (resp. denominator)

polynomial and ρ_i produces the i^{th} factor of the three variables cubic polynomial. The functions are defined between normalized coordinate systems.

A cubic polynomial in three variables has 20 coefficients, thus each component of the localization and projection functions requires 40 coefficients. With the additional 10 parameters specifying the scale and offset for the five variables x, y, λ, θ and h, the RPC camera model for an image is described by a total of 170 coefficients.

In the remainder of this thesis, we will use the notations L and P to denote the RPC localization and projection functions associated to the images. If needed, we may indicate the underlying image with a subscript: L_u and P_u are the RPC functions associated to image u. Ideally, these functions should verify

$$L_u(P_u(\lambda, \theta, h), h) = (\lambda, \theta) \tag{2.44}$$

and

$$P_u(L_u(x, y, h), h) = (x, y), \tag{2.45}$$

but as any model the rational polynomial projection has a limited precision. In particular the two RPC functions are not exact inverses of each other. The errors due to concatenating the projection and inverse functions are negligible, being of the order of 10^{-7} degrees in WGS 84 coordinates, and $\frac{1}{100}$ of pixel in the image i.e. about 1 cm on the ground.

Note that Pléiades images are provided by Airbus Defence and Space with both the localization L and projection P functions, while DigitalGlobe and Skybox provide only the projection function P (i.e. only 90 coefficients). The inverse L has to be estimated from P. This is done iteratively using an algorithm similar to the algorithm 1.

2.6 Conclusion and Future Work

We gave a detailed description of a simplified orbiting pushbroom camera model and showed that, under reasonable assumptions, the main factor responsible for the localization error are the roll and pitch functions. We presented a simple algorithm to refine the roll and pitch functions using ground control points. It exploits the fact that each ground control point imposes two independent constraints on the instantaneous values of the roll and pitch, thus permits to compute these values. Experimental results show that as soon as one has $n + 1$ control points on roughly well spaced rows, a polynomial error of degree no greater than n is substantially corrected by the proposed algorithm. In contrast to classical minimization approaches, this algorithm has the particularity that outliers are automatically detected and removed, preventing them to bias the final result.

To further improve the localization accuracy, once the roll and pitch have been refined, one could refine the yaw coefficients by minimizing the mean quadratic distance between world points and back-projected rays (see energy (2.21)). Other parameters such as the orbit parameters and the focal length may be incorporated in that energy to be refined too. The Hessian of the energy could be used to evaluate the quality of the extrema.

Future work will also include careful experimentation of the proposed algorithm with real data, and study whether it is possible or not to reduce the need for ground control points by using stereo images, i.e. sequences of at least two views of the same area. Refining the orientation parameters of several cameras given a sequence of views of the same scene is a

problem known in computer vision and photogrammetry as *bundle adjustment*. Several open source tools have been developed to solve this problem, such as Apero [Pierrot Deseilligny and Clery 2012], Bundler [Snavely et al. 2006] and OpenMVG [Moulon et al. 2013]. It is likely that ground control points will always be needed because, as was shown by Jama et al. [2011], the bundle adjustment problem with pushbroom cameras has several degrees of freedom, even if the camera positions are known.

parameters					results					
	noise		perturbation		roll error (µrad)		pitch error (µrad)		loc. error (m)	
satellite type	σ_{image}	σ_{world}	d	η (µrad)	before	after	before	after	before	after
Pléiades	0.5 px	0.2 m	0	50	76.48	0.01	61.33	0.92	43.03	0.40
Pléiades	0.5 px	0.2 m	1	50	40.11	0.08	39.51	0.59	24.75	0.26
Pléiades	0.5 px	0.2 m	2	50	60.53	1.01	27.42	0.89	29.16	0.59
Pléiades	0.5 px	0.2 m	3	50	34.74	0.94	52.42	5.24	27.62	2.36

Table 2.2 – Numerical parameters and results of the experiments shown in Figure 2.10, just below. For each polynomial error degree d, only $d+1$ control points are used.

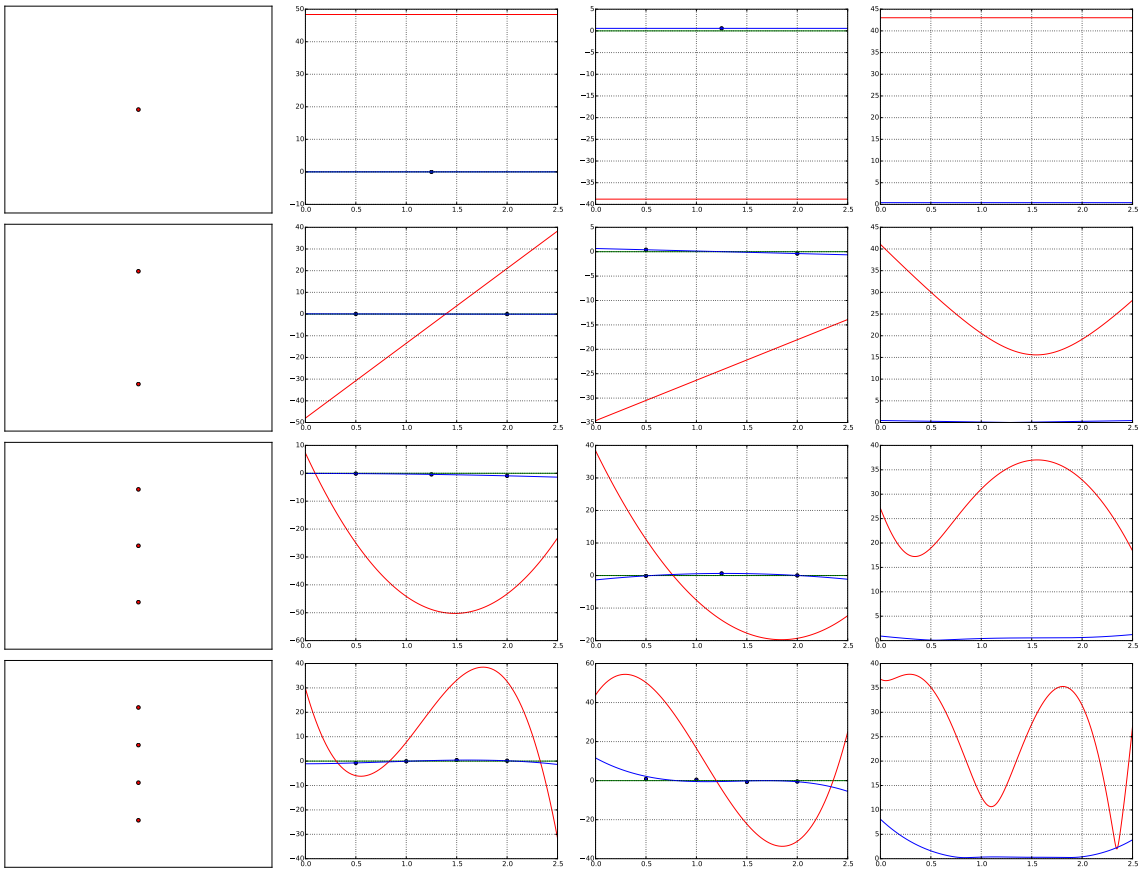


Figure 2.10 – Minimal number of ground control points needed to correct the roll and pitch error, depending on the degree of the polynomial error. Each of the four rows presents a different experiment. On the d^{th} row, a polynomial error of degree $d-1$ is corrected with d control points. The numerical parameters and results of these experiments are given in Table 2.2.

parameters					results					
	noise		perturbation		roll error (µrad)		pitch error (µrad)		loc. error (m)	
satellite type	σ_{image}	σ_{world}	d	η (µrad)	before	after	before	after	before	after
Pléiades	0.5 px	0.2 m	3	50	10.97	2.96	51.50	2.15	23.11	1.62
Pléiades	0.5 px	0.2 m	3	50	10.97	5.75	51.50	9.58	23.11	4.94
Pléiades	0.5 px	0.2 m	3	50	10.97	18.10	51.50	9.07	23.11	8.89
Pléiades	0.5 px	0.2 m	3	50	10.97	34.91	51.50	58.08	23.11	29.84

Table 2.3 – Numerical parameters and results of the experiments shown in Figure 2.11, just below. As soon as two control points are located barely on the same row, they count as only one point. This affects the accuracy of the refinement.

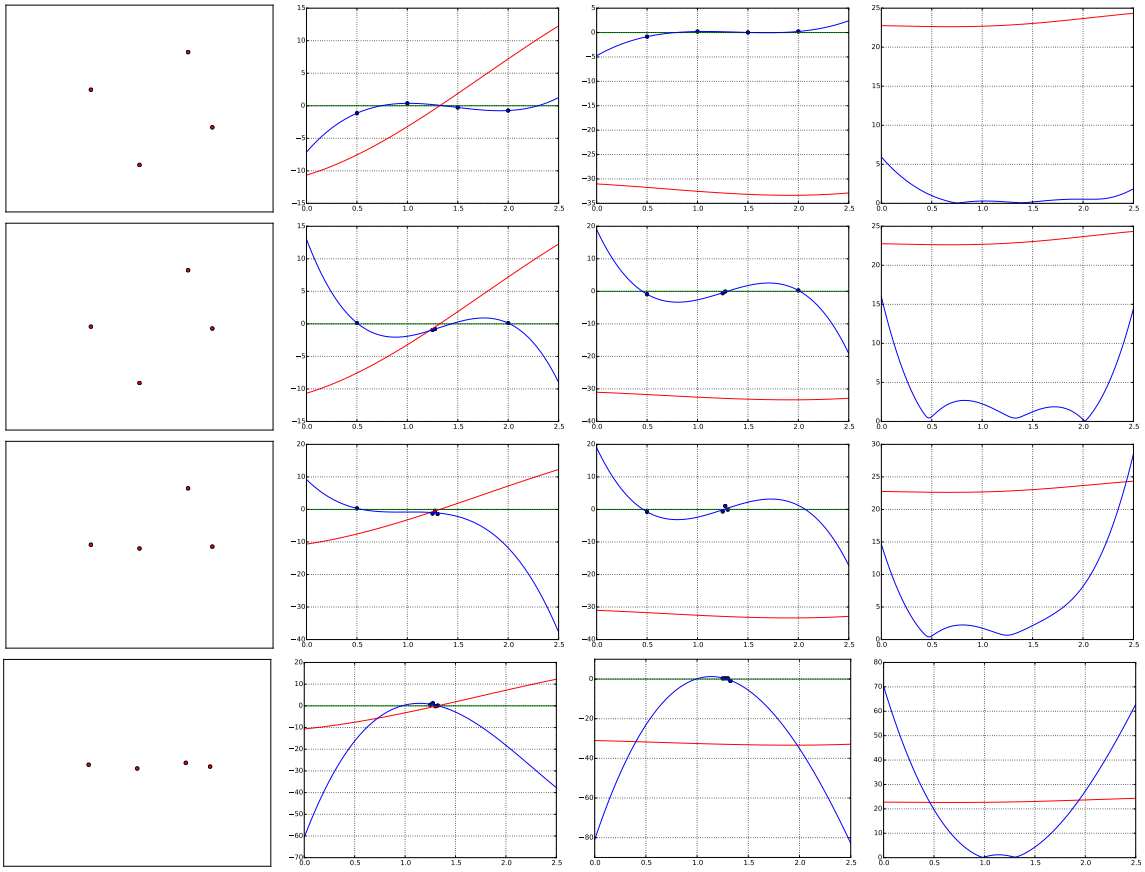


Figure 2.11 – Each of the four rows presents a different experiment. The polynomial error of degree 3 added to the attitudes is identical in the four experiments, as well as the number of control points, but the positions of the control points vary. When two control points are located on neighboring rows in the image plane, their effect on the estimated roll and pitch functions is that of a single control point. This is why on the second experiment shown here the polynomial fitting of the roll and pitch corrections is poorly conditioned. We are trying to fit a polynomial of degree up to 3 to three points. Hence the refinement is less accurate. The situation gets worse on the third and fourth experiments, as confirmed by the numerical results given in Table 2.3.

parameters					results					
	noise		perturbation		roll error (µrad)		pitch error (µrad)		loc. error (m)	
satellite type	σ_{image}	σ_{world}	d	η (µrad)	before	after	before	after	before	after
Pléiades	0.5 px	0.2 m	3	50	45.87	0.93	31.18	1.01	24.33	0.60
Pléiades	1.0 px	1.0 m	3	50	45.87	8.84	31.18	7.41	24.33	5.07
Pléiades	1.0 px	1.0 m	3	50	45.87	2.72	31.18	2.12	24.33	1.52
Pléiades	2.0 px	2.0 m	3	50	45.87	3.30	31.18	7.50	24.33	3.60
Pléiades	2.0 px	2.0 m	3	50	45.87	1.43	31.18	2.41	24.33	1.23

Table 2.4 – Numerical parameters and results of the experiments shown in Figure 2.11, just below. As soon as the accuracy of the ground control points decreases, the overall accuracy of the refined attitudes decreases. This effect can be compensated by adding more ground control points.

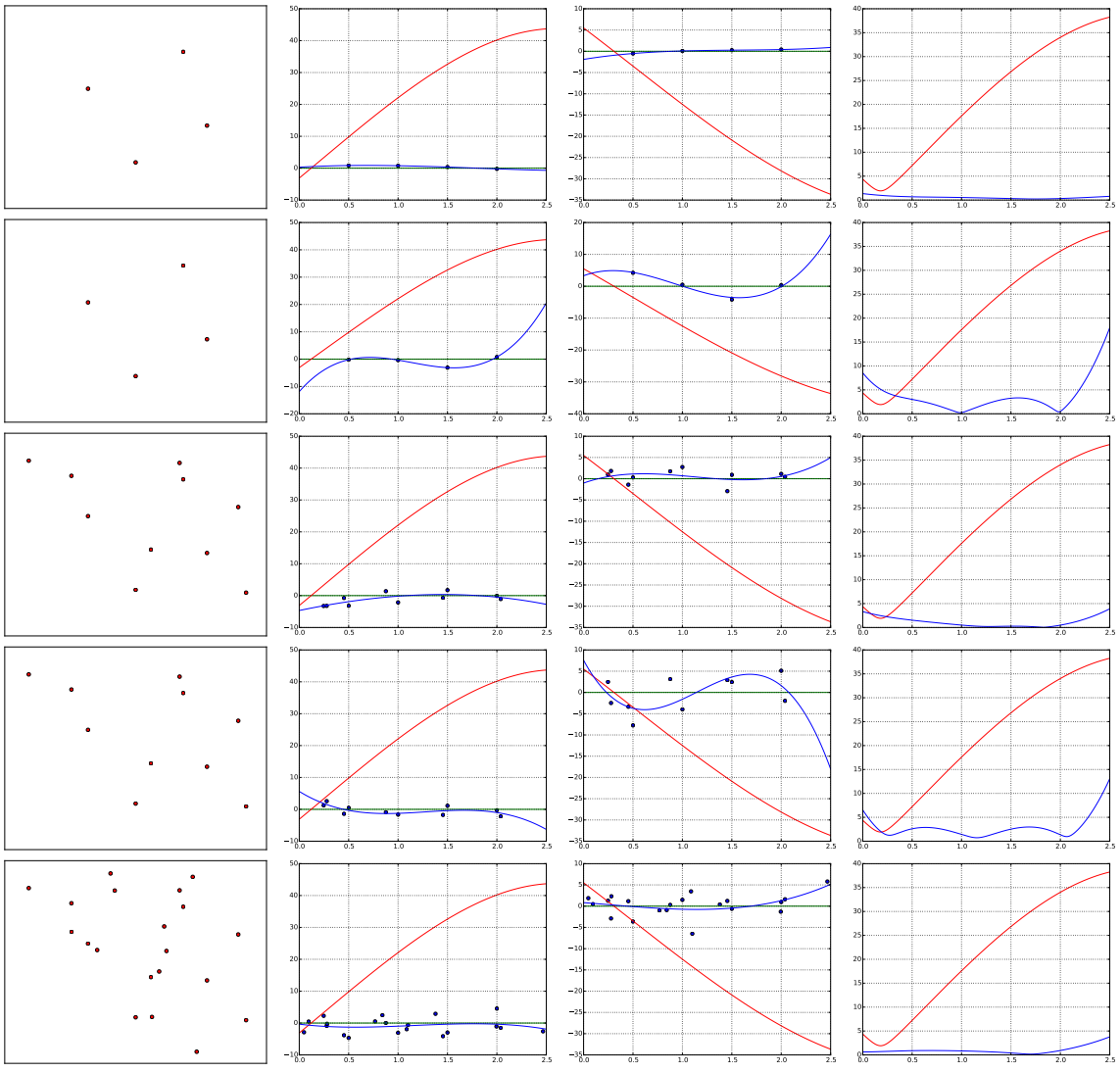


Figure 2.12 – In the first experiment (first row) the roll and pitch are refined from four control points. In the second row, the same experiment is done with a higher level of noise. The accuracy of the output roll and pitch functions is lower. In the third row the same experiment is done with six additional control points. The output is more accurate. Similarly, on the fourth row the noise level is increased and in the fifth row 10 points are added.

3 Epipolar Rectification of Pushbroom Images

Epipolar rectification is a common technique used in 3D reconstruction algorithms. It permits to simplify the search of corresponding points between the images of a stereo pair. However, it applies only to images taken with pinhole cameras. Pushbroom cameras produce images for which any epipolar rectification is theoretically not possible. In this chapter we study to what extent it is possible to apply epipolar rectification to pushbroom images anyway, in order to use standard matching algorithms from the image processing and computer vision communities for processing satellite stereo pairs. The approach presented here considers the epipolar rectification as an auxiliary step for the computation of stereo correspondences, not as a final product. Images are thus processed in small tiles by locally approximating the pushbroom camera with an affine camera model. The explicit modeling of the approximation allows to quantify and control the epipolar errors without needing ground control points. Experiments on Pléiades and WorldView-1 images of many kinds of scenes (urban, mountainous, flat) demonstrate that epipolar errors can be reduced to one tenth of pixel.

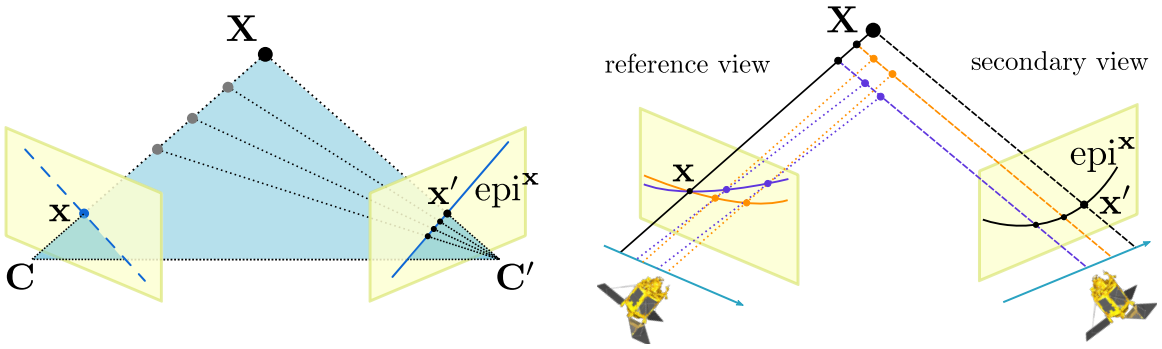
The stereo-rectification algorithm presented in this chapter can be tested online on any crop of a satellite image from our test database at
http://dev.ipol.im/~carlo/ipol_demo/s2p_rectification.

3.1 Introduction

Epipolar rectification is a technique that permits to simplify and accelerate the computation of image point correspondences between the views of a stereo pair. It restricts the search for corresponding image points from the entire image plane to a single line. For any point $\mathbf{x}$ of the first view, the corresponding point $\mathbf{x}'$ in the other view, if it exists, lies on the *epipolar line* of $\mathbf{x}$ denoted by $\text{epi}^{\mathbf{x}}$. Conversely $\mathbf{x}$ lies on $\text{epi}^{\mathbf{x}'}$. The epipolar rectification, also known as *stereo-rectification* or *epipolar resampling*, aims to resample the images in such a way that corresponding points are located on the same row, thus simplifying the matching task and permitting to use all classic stereo matching algorithms, such as those listed in the Middlebury Stereo Evaluation[1] benchmark.

For images taken with pinhole cameras there is a correspondence between the epipolar lines of the two views. All the points $\mathbf{x}'$ of the second view lying on the epipolar line $\text{epi}^{\mathbf{x}}$ share the same epipolar line in the first view. Epipolar lines $\text{epi}^{\mathbf{x}}$ and $\text{epi}^{\mathbf{x}'}$ are said to be *conjugate*. Figure 3.1 (a) illustrates the conjugacy of epipolar lines. It is well-known [Hartley and Zisserman 2004] that images can be resampled according to a planar homography in order to produce a rectified pair in which the epipolar lines are horizontal and conjugate epipolar lines have the same row index. Several algorithms have been proposed to compute the resampling homographies [Loop and Zhang 1999, Monasse et al. 2010, Monasse 2011]. Matching rectified images is much simpler than matching the original images, because the search of correspondences is performed along horizontal lines only [Ohta and Kanade 1985a].

[1] http://vision.middlebury.edu/stereo/



(a) In the pinhole case the epipolar plane defines a one-to-one correspondence between epipolar lines. Corresponding epipolar lines are said to be *conjugate*.

(b) In the pushbroom case the projection of a 3-space ray on the secondary view generates a ruled quadric. The projection of this quadric on the reference view contains many epipolar curves: epipolar curves are not conjugate.

Figure 3.1 – Difference between the epipolar constraints of the pinhole camera model (a) and the pushbroom camera model (b).

Satellite images however cannot be rectified because they are taken with pushbroom sensors, for which the pinhole model is invalid. Using various pushbroom camera models [Orun and Natarajan 1994, Gupta and Hartley 1997], it has been shown that pushbroom image pairs have non-straight epipolar curves and that these curves are not conjugate, making stereo-rectification impossible [Kim 2000, Habib et al. 2005]. The case of linear pushbroom cameras (see section 2.2.1) is illustrated by Figure 3.1 (b). In this chapter we study to what extent it is possible to perform an epipolar rectification of parts of pushbroom images anyway, in order to use standard matching algorithms to process stereo images taken by Earth observation satellites. This study was motivated by the availability of high resolution images from new satellites with stereo capabilities such as Pléiades.

3.1.1 Previous Work

The non rectifiability of pushbroom images did not block the development of 3D reconstruction solutions for satellite images. Many solutions have been proposed to circumvent this issue. We may group them into three categories:

1. No rectification: many authors [Lee et al. 2003, Hirschmüller et al. 2005, Hirschmüller 2008] propose to keep the original images unchanged and to perform stereo matching by following the non-straight epipolar curves. This approach eliminates the need for epipolar rectification while keeping the benefits of one-dimensional exploration. However, non-straight epipolar curves may prevent from applying stereo matching optimizations and from using off-the-shelf correlators. Other methods [Paparoditis and Pierrot Deseilligny 2006] estimate directly a Digital Elevation Model (DEM) by minimizing an energy function. This function of the DEM is made of a data term measuring the images similarity at the projections of DEM points, plus a regularization term penalizing the variations of the DEM. The energy minimization involves graph cuts techniques (see

introduction of chapter 5), which are computationally intensive. A limitation of these methods is that they do not allow to compute a real 3D model. The vertical facades of a building, for example, will not be reconstructed even if they are visible in the images.

2. Affine camera approximation: other authors [Ono 1988, Fraser et al. 2004, Morgan et al. 2006, Wang et al. 2011] propose to approximate the pushbroom sensor with an affine camera model. This approach usually uses Ground Control Points (GCP) to estimate the affine model for each image, and the overall achieved precision is on the order of one pixel on images from SPOT and Ikonos satellites.

3. Polynomial epipolar rectification [Oh et al. 2010, Christophe et al. 2008]: Oh et al. show that even if pairs of epipolar curves do not exist in the pushbroom case, for small altitude ranges of the scene one may assume that curve pairs exist with small error. Thus they build whole epipolar curve pairs on Ikonos stereo images by putting together small pieces of corresponding curves. Then they resample the images to transform these curves into straight horizontal lines. They report a maximal error of one pixel. Since their rectification procedure is non-linear, i.e. it is not a homographic resampling, it cannot guarantee that straight lines are preserved.

It is important to note that errors in the rectification are critical as they may result in a vertical disparity between corresponding points in the rectified images, which may hurt the performance of the stereo matching. We refer to this vertical disparity as *epipolar error*. The epipolar error is the ultimate performance measure for the different methods. Current state of the art methods attain errors on the order of one pixel. The method presented in this chapter lowers this error by one order of magnitude.

3.1.2 Our Contribution

Our goal is to build an automatic 3D reconstruction pipeline for satellite images. We started by observing that a large-scale rectified pair is not needed for applying a stereo matching algorithm. The need of a rectified whole stereo pair may be motivated by the photointerpreter's use, which match points manually using the rectified images. However this does not apply for an automatic matching method. Thus we propose, like Morgan et al. [2006], to approximate the sensor by an affine camera model. But, unlike Morgan et al. [2006], this approximation is made only on small image tiles. This limits the discrepancy between epipolar curves, as was studied by Oh et al. [2010]. It leads in practice to an almost perfect rectification, with a very small epipolar error. By not imposing to build a single large rectified image we greatly reduce the residual vertical disparities and thus reduce their impact on the stereo matching results.

In our approach the rectification is seen as an intermediary step to efficiently solve the stereo matching problem, not as a product per se. For each locally rectified tile a standard off-the-shelf stereo matching algorithm can be applied to estimate a horizontal disparity map, with high chances of success thanks to the subpixel accuracy of the stereo-rectification. The computed correspondences are then transferred back to the coordinate system of the original images. This eliminates the need for rectifying the whole images all at once.

The proposed local rectification hinges on the external and internal camera calibration information, provided for satellite images as localization and projection functions, as explained

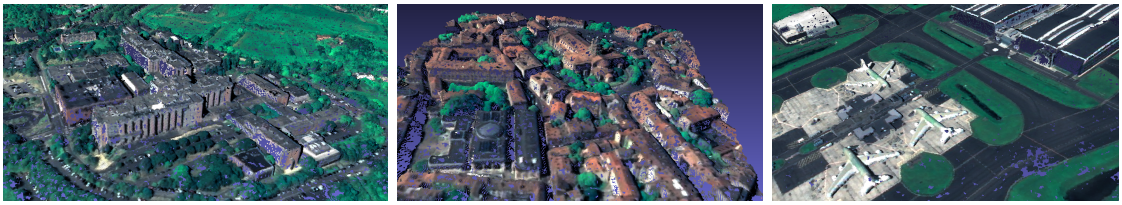


Figure 3.2 – 3D point clouds automatically generated from Pléiades tri-stereo datasets, without any manual intervention, with the S2P *Satellite Stereo Pipeline* which is available online on the IPOL website.

in section 2.5, page 50. This information permits to quantify a priori the epipolar error due to the affine approximation. Conducting our experimentation on several satellite stereo pairs of urban, flat and mountainous regions, we obtained errors on the order of $\frac{1}{20}$ of pixel.

The proposed method is currently being used by CNES on Pléiades images, as part of the S2P *3D reconstruction stereo pipeline*, which will be presented in chapter 6. It is an automatic pipeline for computing digital elevation models and 3D points clouds from stereo and tri-stereo datasets. Figure 3.2 shows some of its results. The method can be tested online[2] and provides rectified stereo pairs to be used by the image processing community.

We detail the proposed solution in the next section and in section 3.3 we validate our approach with extensive experimentation carried out using images from the Pléiades satellites.

3.2 Local Rectification without Control Points

While Morgan et al. [2006] uses ground control points to estimate the affine camera models, this work uses the standard computer vision approach for epipolar rectification [Hartley and Zisserman 2004]: first estimate the affine fundamental matrix that encapsulates completely the affine geometry of the pair of cameras, then compute a pair of transformations to rectify the images. The fundamental matrix estimation requires only image matches, eliminating the need for ground control points and manual intervention.

3.2.1 In Defense of the Affine Approximation

While epipolar rectification of images from pinhole cameras is generally done with planar homographies [Loop and Zhang 1999], for small pieces of satellite images one may use simpler transformations as the sensor can be modeled by an *affine* camera. The suitability of the affine camera model in approximating a satellite pushbroom sensor can be attributed to Okamoto et al. [1993]. Its main arguments are all applicable to Pléiades and WorldView images:

- Altitude differences in the photographed terrain are small in comparison with the flying altitude of the satellite, whose average is 694 km for Pléiades.
- The angular field of view of the sensor is narrow. For a full Pléiades image it is less than 2°, and it is much less if one considers only a small tile.

[2] http://dev.ipol.im/~carlo/s2p

- The acquisition time of such a tile is less than one tenth of a second, thus the sensor may be assumed to have constant attitude and speed.

3.2.2 Virtual Correspondences Generation

A natural way to compute correspondences between the two views is to extract feature points, compute descriptors and match them, as done by SIFT [Lowe 2004, Rey Otero and Delbracio 2014]. But this may lead to a set of keypoints all lying on the same plane, i.e. on the ground. This configuration is degenerate and $\mathtt{F}$ cannot be computed from it, even with robust methods such as [Espuny et al. 2014]. This problem occurs also if the keypoints do not exactly lie on the same plane: as relief reduces to zero, the covariance of the estimated $\mathtt{F}$ increases [Hartley and Zisserman 2004]. We found that a safer way to estimate $\mathtt{F}$ is to use the orientation parameters of the cameras, usually provided by image vendors as RPC coefficients. The RPC model and the associated notations are introduced in section 2.5 of chapter 2. A detailed presentation and in depth analysis of the RPC model are given in [Tao and Hu 2001].

As previously done by Oh et al. [2010], we use the RPC functions to generate virtual correspondences between the two views. Given a region Ω in the reference image and an estimated altitude range $[h_m, h_M]$ for the associated 3-space points (i.e. points that were imaged into Ω) Ω is back-projected on the Earth surface thanks to the RPC localization function L. Let denote by $\Gamma = L(\Omega \times [h_m, h_M]) \subset \mathbf{R}^3$ the back-projected domain[3], and by $(\mathbf{X}_i)_{i=1,\dots,n}$ a regular sampling of Γ. Each 3-space point $\mathbf{X}_i$ is projected on the two images using the associated RPC projection functions, leading to a virtual correspondence $(\mathbf{x}_i, \mathbf{x}'_i)$. The images contents at locations $\mathbf{x}_i$ and $\mathbf{x}'_i$ may not correspond, but $\mathbf{x}'_i$ is located on the epipolar curve of $\mathbf{x}_i$, and that is enough to estimate a fundamental matrix.

How to determine the altitude range. Starting with a rough altitude range estimation, given by the RPC validity domain which is usually provided with the coefficients [Coeurdevey and Gabriel-Robez 2012], Ω is back-projected on the WGS 84 ellipsoid. All the SRTM samples (see appendix B) in the back-projected region are collected. The extremal values of these samples give a better control of the needed altitude range.

3.2.3 Local Affine Rectification Algorithm

The stereo-rectification algorithm is summarized in Algorithm 6. Given a list of at least 4 correspondences $(\mathbf{x}_i, \mathbf{x}'_i)_{i=1,\dots,n}$ between the two views, the *affine fundamental matrix* $\mathtt{F}$ is estimated using the Gold Standard algorithm [Hartley and Zisserman 2004]. An affine fundamental matrix is an homogeneous 3×3 matrix of the form

$$\mathtt{F} = \begin{bmatrix} 0 & 0 & a \\ 0 & 0 & b \\ c & d & e \end{bmatrix}. \tag{3.1}$$

Note that the estimation problem is easier for affine fundamental matrices than for general fundamental matrices. In particular, the Gold Standard algorithm is reduced to performing

[3] in section 2.2.4 the localization function L was defined with values in $\mathbf{R}^2$. Here the input altitude h is included in the function outputs thus L has values in $\mathbf{R}^3$.

the Singular Value Decomposition (SVD) of a $n \times 4$ matrix, as can be seen in Algorithm 5 which we reproduce here from chapter 14 of [Hartley and Zisserman 2004].

Algorithm 5: The Gold Standard algorithm for estimating $\mathtt{F}$ from image correspondences.

Input: $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}_{i=1,\dots,n}$: set of $n \geq 4$ image point correspondences. A correspondence is represented as $\mathbf{X}_i = (\mathbf{x}'_i, \mathbf{z}'_i, \mathbf{x}_i, \mathbf{z}_i)^\top$.

Output: $\mathtt{F}$: affine fundamental matrix

begin

1. Compute the centroid $\bar{\mathbf{X}} = \frac{1}{n}\sum_i \mathbf{X}_i$ and center the vectors $\Delta\mathbf{X}_i = \mathbf{X}_i - \bar{\mathbf{X}}$.
2. Compute the $n \times 4$ matrix $\mathtt{A}$ with rows $\Delta\mathbf{X}_i^\top$.
3. Then $N = (a, b, c, d)^\top$ is the singular vector corresponding to the smallest singular value of $\mathtt{A}$, and $e = -N^\top \bar{\mathbf{X}}$. The matrix $\mathtt{F}$ has the form (3.1).

Once the affine fundamental matrix is known, one can compute two rectifying similarities using formulas (3.2) and (3.3).

Algorithm 6: Locally affine stereo-rectification of pushbroom images.

Input: L_1, P_1, L_2, P_2: RPC localization and projection functions of the input images
$x, y, w, h \in \mathbf{R}$: coordinates of a rectangular region of interest in the first view
SRTM data (when available)

Output: $\mathtt{S}_1, \mathtt{S}_2$: rectifying similarities

begin

1. estimate the altitude range $[h_m, h_M]$ *// from RPCs or SRTM*
2. compute n virtual matches $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ *// section 3.2.2*
3. estimate $\mathtt{F}$ from the set of matches $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}_{i=1,\dots,n}$ *// algorithm 5*
4. compute $\mathtt{S}_1$ and $\mathtt{S}_2$ from $\mathtt{F}$ *// formula 3.3*

$$\mathtt{R}_1 = \frac{1}{\sqrt{a^2+b^2}} \begin{bmatrix} b & -a \\ a & b \end{bmatrix} \qquad \mathtt{R}_2 = \frac{1}{\sqrt{c^2+d^2}} \begin{bmatrix} -d & c \\ -c & -d \end{bmatrix} \tag{3.2}$$

$$\mathtt{S}_1 = \left[\begin{array}{cc|c} & & 0 \\ & z\mathtt{R}_1 & t \\ \hline 0 & 0 & 1 \end{array}\right] \qquad \mathtt{S}_2 = \left[\begin{array}{cc|c} & & 0 \\ & \frac{1}{z}\mathtt{R}_2 & -t \\ \hline 0 & 0 & 1 \end{array}\right] \tag{3.3}$$

3.3 Experimental Validation

The proposed stereo-rectification method is evaluated by measuring the epipolar error, which is completely determined by the fundamental matrix $\mathtt{F}$. This error is measured by

$$\max_{i \in \{1,\dots,n\}} \max\{d(\mathbf{z}'_i, \mathtt{F}\mathbf{z}_i), d(\mathbf{z}_i, \mathtt{F}^\top \mathbf{z}'_i)\}, \tag{3.4}$$

where $d(\mathbf{z}, \mathbf{l})$ is the distance, in pixels, between a point $\mathbf{z}$ and a line $\mathbf{l}$. The matches $\{\mathbf{z}_i \leftrightarrow \mathbf{z}'_i\}$ are new virtual correspondences obtained from a sampling of volume Γ. This error is the

maximal distance between a point and the epipolar line of its match (computed for both points of the match). The distance $d(\mathbf{z}'_i, \mathtt{F}\mathbf{z}_i)$ between a point $\mathbf{z}'_i$ and the epipolar line it is supposed to lie on ($\mathtt{F}\mathbf{z}_i$) is computed as

$$d(\mathbf{z}'_i, \mathtt{F}\mathbf{z}_i) = \frac{|\mathbf{z}'^{\top}_i \mathtt{F}\mathbf{z}_i|}{\sqrt{(\mathtt{F}_1^{\top}\mathbf{z}_i)^2 + (\mathtt{F}_2^{\top}\mathbf{z}_i)^2}}, \tag{3.5}$$

where $\mathtt{F}_1^{\top}, \mathtt{F}_2^{\top}$ and $\mathtt{F}_3^{\top}$ denote the three rows of matrix $\mathtt{F}$.

3.3.1 Numerical Results

From a geometric viewpoint, the locally affine rectification method described in this chapter amounts to approximate the two pushbroom sensors with affine camera models. The validity of such an approximation relies on the dimensions of the 3-space domain on which it is used, and on its location. These dimensions depend on two parameters: the tile size and the width of the altitude range. To understand the influence of these two parameters on the epipolar error, experiments were performed on numerous Pléiades stereo datasets, listed in Table 3.1. Each dataset contains a stereo pair of images together with their geometrical calibration data given by the RPC functions. The validity domain of the RPC functions in the altitude variable gives a rough estimation of the altitude range of the scene.

Figures 3.3 and 3.4 show the error measured on each dataset by varying the tile size and the width of the altitude range separately. Figure 3.3 shows the errors obtained when varying the tile size up to 5000×5000 pixels, while keeping the width of the altitude range fixed to 2500 m (or less if needed by the RPC validity range). This is enough because at Pléiades resolution, i.e. 50 cm per pixel, 5000 pixels correspond to 2.5 km, and it is reasonable to assume that no place on Earth is steeper than 45° when averaged on a 2.5×2.5 km area, even in high mountain areas such as the Himalaya. It is clear from the plots of Figure 3.3 that for tiles of size 1000×1000 pixels the error is always lower than 0.1 pixel. The error increases with the tile size, and it increases faster for mountainous regions like Mont Blanc (Alps), Mera peak (Himalaya), Ossoue glacier (Pyrenees), Ubaye valley (Alps) and Spitsberg island. This may be due to the fact that the RPCs of the other datasets have an altitude validity range narrower than 2500 m.

These results are confirmed by the plots presented on Figure 3.4, where the tile size is fixed to 1000×1000 pixels while the width of the altitude range is varied up to 3000 m. Whenever the validity domain of RPCs for the altitude is narrower than 3000 m, the width of the altitude range is varied only up to that limit. This is why some plots do not span the entire figure. This figure confirms that on any Pléiades dataset, for any altitude variations in the scene, if one considers a tile of 1000×1000 pixels the epipolar error due to the affine stereo-rectification is smaller than 0.1 pixel. This is the main result of the chapter.

By using data from the SRTM database (see appendix B) the estimation of the altitude range of the scene can be improved. The accuracy gained on the altitude range allows to use bigger tiles, while keeping the epipolar error lower than 0.1 pixel. Figure 3.5 shows the influence of the tile size on the error when the altitude range is estimated from SRTM data. These results show that it is always possible to rectify tiles of size 1000×1000 pixels with an epipolar error lower than 0.05 pixels.

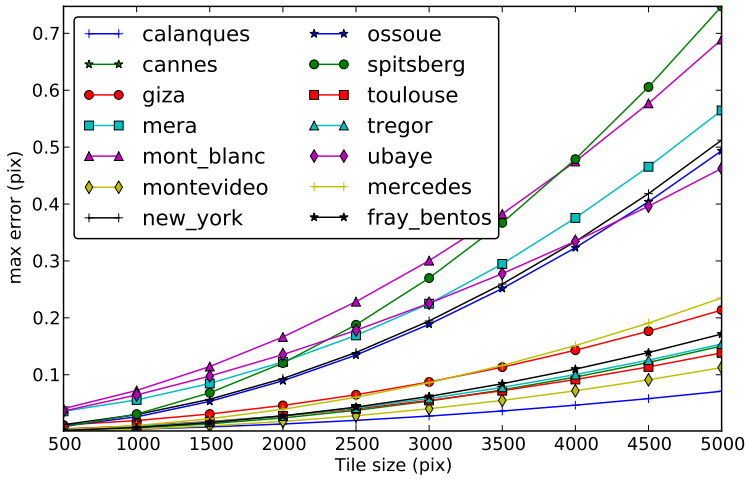


Figure 3.3 – Dependence of the epipolar error with the tile size. Each curve corresponds to a stereo pair. A square tile of size ranging from 500×500 to 5000×5000 pixels was selected in the middle of the reference image. The experiments presented in Figure 3.6 show indeed that the position of the tile in the image does not influence the result. Virtual matches between this central tile and the secondary image were computed using an altitude range of width fixed to $2500\,\mathrm{m}$. The altitude range was centered in the RPCs validity domain, and shrinked, if needed, to fit the validity domain. The error reported by each plot is the maximal distance between a point and the epipolar line of its match. The epipolar lines were computed thanks to the affine fundamental matrix estimated from the virtual matches, as explained in Algorithm 6.

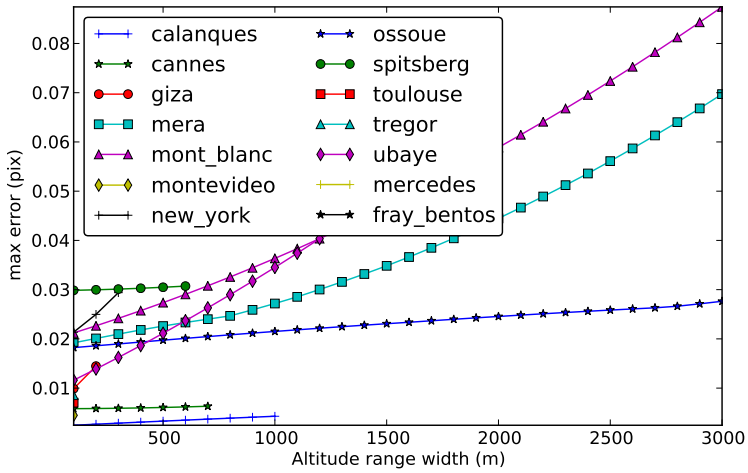


Figure 3.4 – Dependence of the epipolar error with the width of the altitude range. Each curve corresponds to a stereo pair. A tile of size fixed to 1000×1000 pixels was selected in the middle of the reference image. Virtual matches between this tile and the secondary image were computed using an altitude range of width ranging from $100\,\mathrm{m}$ to $3000\,\mathrm{m}$. The altitude range was centered in the RPCs validity domain. For some datasets, it was not possible to extend the altitude range up to $3000\,\mathrm{m}$ without getting out from the validity domain. This is why some plots stop at smaller range widths. The error reported by each plot is the maximal distance between a point and the epipolar line of its match. The epipolar lines were computed thanks to the affine fundamental matrix estimated from the virtual matches, as explained in Algorithm 6.

Dataset	Scene dimensions (km)	RPC altitude validity (m)
calanques	25×24	40 – 1090
cannes	21×20	50 – 830
giza	26×23	10 – 290
mera	25×42	-10 – 8610
mont_blanc	21×15	850 – 4730
montevideo	22×20	-10 – 150
new_york	48×37	-120 – 190
ossoue	22×22	-10 – 3320
spitsberg	21×20	-10 – 640
toulouse	25×21	150 – 340
tregor	26×24	50 – 160
ubaye	22×15	1100 – 3050
mercedes	25×23	10 – 90
fray_bentos	22×20	0 – 80

Table 3.1 – Pléiades datasets used for the experiments.

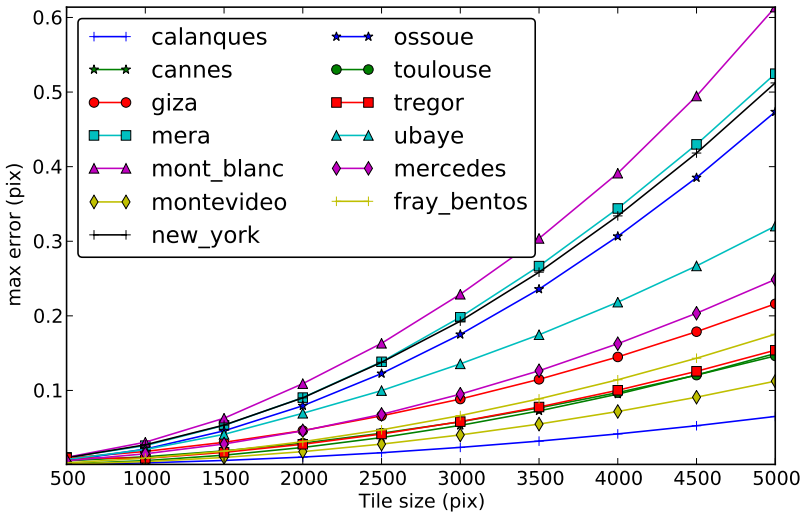
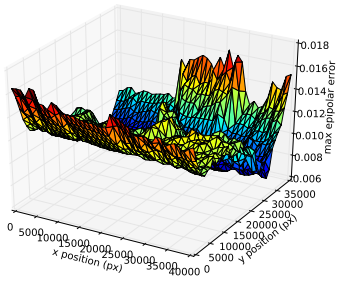
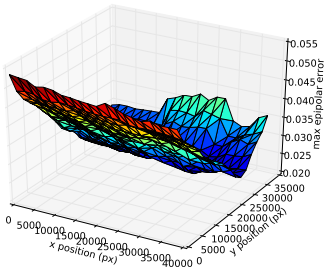


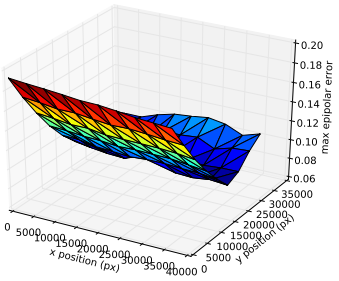
Figure 3.5 – Dependence of the epipolar error with the tile size. Each curve corresponds to a stereo pair. A tile of size ranging from 500×500 to 5000×5000 pixels was selected in the middle of a Pléiades reference image. Virtual matches were computed using the refined altitude range given by the SRTM data. The error reported by each plot is the maximal distance between a point and the epipolar line of its match. Spitsberg is located above latitude $60°$N, outside SRTM domain, thus omitted here.



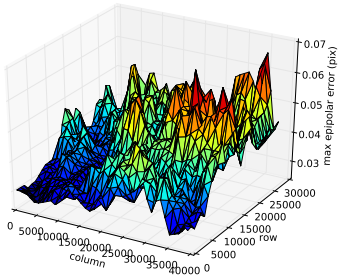
(a) Tiles of size 1000×1000



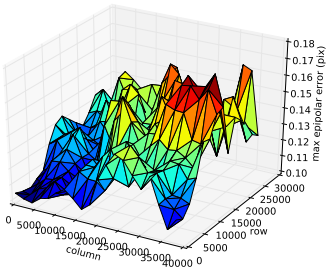
(b) Tiles of size 2000×2000



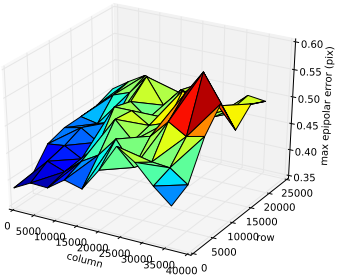
(c) Tiles of size 4000×4000



(d) Tiles of size 1000×1000



(e) Tiles of size 2000×2000



(f) Tiles of size 4000×4000

Figure 3.6 – Maximal epipolar error on a tile, using different tile sizes ranging from 1000×1000 to 4000×4000 pixels, depending on the position of the tile in the full image. Two different datasets are used: Toulouse in figures (a), (b) and (c), and Mont Blanc in figures (d), (e) and (f). The error distributions are quite different for the two datasets. On the one hand, this indicates that there is no preferred image location to measure the highest epipolar errors that works across all datasets. On the other hand, the ratio between maximum and minimum on these plots is lower than 3. This indicates that if we measure the epipolar error on a given tile, we are within a factor 3 of the maximal epipolar error that could be measured on the image.

3.4 Conclusion

In this chapter we showed that, for the purpose of stereo matching, pushbroom satellite images can be rectified and therefore be reduced to standard stereo pairs. This permits to produce and make available to the stereo community rectified pairs from real satellite images, that can be processed using standard computer vision tools. Thorough experimentation on numerous Pléiades datasets has shown that using tiles of size 1000×1000 pixels ensures a precision of 0.1 pixel, regardless of the altitude range of the scene. The stereo-rectification described here is implemented in the Satellite Stereo Pipeline `S2P` (see chapter 6). This method could be used on images from other Earth observation satellites such as WorldView-2 and GeoEye-1. However, since for these images only the direct RPC projection function is provided, the inverse RPC^{-1} must be estimated, as was mentioned in section 2.5.

4 Local Correction of the Relative Pointing Error

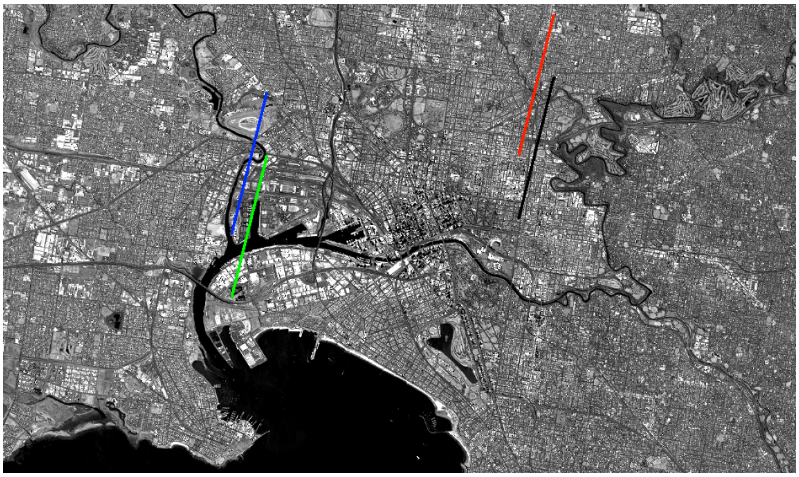
Modern Earth observation satellites are calibrated in such a way that a point on the ground can be located with an error of just a few pixels in the image domain. For many applications this error can be ignored, but this is not the case for stereo reconstruction, which requires sub-pixel accuracy. In this chapter we propose a method to correct this error. The method works by estimating local corrections that compensate the error relative to a reference image. The proposed method does not rely on ground control points, but only on the relative consistency of the image contents. We validate our method with Pléiades and WorldView-1 images on a representative set of geographic sites.

The relative pointing correction algorithm presented in this chapter is used in the stereo-rectification method described in chapter 3, which can be tested online at `http://dev.ipol.im/~carlo/ipol_demo/s2p_rectification`.

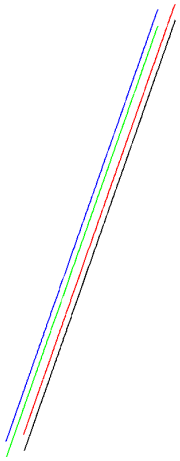
4.1 Introduction

As was pointed out in the general introduction (see chapter 1), the agility of modern Earth observation satellites such as Pléiades allows them to capture multiple views of the same target in a single orbit. This enables nearly simultaneous stereo acquisitions aimed at producing accurate 3D models. The internal and external parameters of the pushbroom camera system are known, including the effects of the atmospheric refraction. Thus, as explained in chapter 2, each image is accompanied by a pair of functions that describe the formation model. These functions permit a conversion of image coordinates to geographic coordinates and back. They are described using the *Rational Polynomial Coefficient* (RPC) camera model, that we introduced in section 2.5.

There is a noticeable error of a few pixels in the RPC functions [Hanley et al. 2002, Fraser and Hanley 2003, Grodecki and Dial 2003, Fraser and Hanley 2005]. This is inevitable due to the limited precision of the camera calibration, especially for the determination of the attitude, as was seen in chapter 2. For many purposes, this error can be ignored, since it typically results in a global offset of the results. However, for stereo matching, the epipolar constraints derived from the parameters of the cameras have to be as precise as possible. The local stereo image rectification algorithm proposed in chapter 3 relies entirely on the RPC functions. Thus the relative error between the RPC functions of the images of a stereo pair must be corrected before applying rectification. In this chapter we propose a method to correct this error relative to a given reference image. Our method does not rely on ground control points, but on the relative consistency of the image contents. Thus it can be implemented as an automatic pre-processing of the input images.



(a) Four epipolar curves plotted in the domain of the second image. They correspond to four points located near the edges of the first image.



(b) The same epipolar curves placed closer to facilitate the comparison.

Figure 4.1 – The RPC projection and localization functions allow to draw the epipolar curves for a pair of Pléiades images of Melbourne. The size of the images is approximately $40\,000 \times 16\,000$ pixels. The range of altitudes considered is $h \in [-200, 3000]$ meters.

4.2 The Relative Pointing Error

4.2.1 Epipolar curves

The knowledge of the localization and projection functions L and P (as defined in (2.14), (2.15) and in section 2.5) for two images u and v allows to give an analytical description of epipolar curves. If $\mathbf{x}$ is a point in image u, then the function

$$\mathrm{epi}_{uv}^{\mathbf{x}} : h \longmapsto P_v(L_u(\mathbf{x}, h), h) \tag{4.1}$$

defines a parametrized curve in the domain of image v containing all the possible correspondences of $\mathbf{x}$ for different altitudes h. This curve is called the *epipolar curve* of the point $\mathbf{x}$. In practice, we observe that these curves are locally straight line segments which are almost parallel (see Figure 4.1).

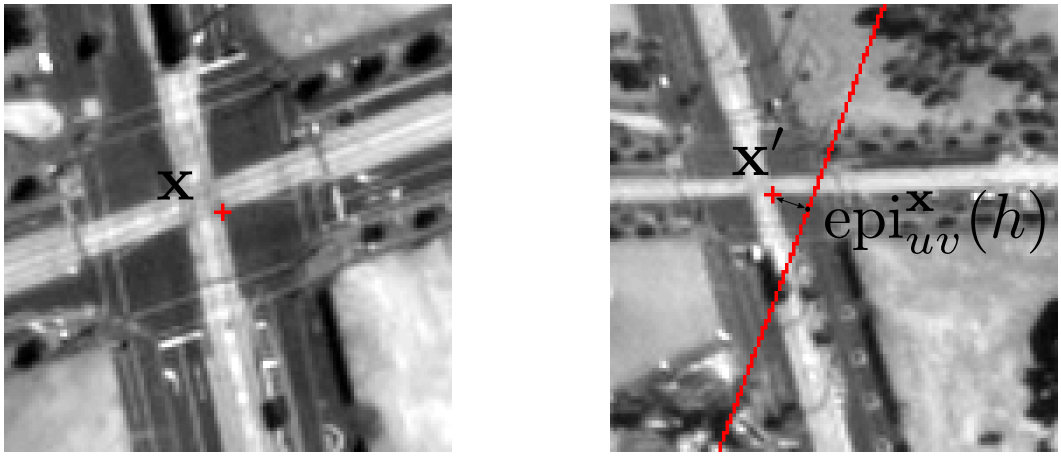
The epipolar curves are used to compute the altitudes of 3-space points which are visible in two images. Suppose that $\mathbf{x}$ is the projection of a point in image u, and $\mathbf{x}'$ is the projection of the same point in image v. Then the epipolar curve of $\mathbf{x}$ passes through $\mathbf{x}'$ and the value h for which

$$\mathbf{x}' = \mathrm{epi}_{uv}^{\mathbf{x}}(h) \tag{4.2}$$

is the altitude of the 3-space point. An algorithm to compute h will be presented in chapter 6.

4.2.2 Relative pointing error evidence

The RPC functions simplify the manipulation of the image formation model. However, due to inevitable errors in the estimation of the external parameters of the system, the formation



(a) First view u. (b) Second view v.

Figure 4.2 – Effect of the satellite relative pointing error on a pair of Pléiades views of a road intersection in Haiti. Two corresponding points $\mathbf{x}$ and $\mathbf{x}'$ are shown, and the epipolar curve of $\mathbf{x}$ as traced by the RPC does not pass through the corresponding point $\mathbf{x}'$. The *relative pointing error* is the distance from the point $\mathbf{x}'$ to the epipolar curve $\mathrm{epi}^{\mathbf{x}}_{uv}(\mathbf{R})$. The altitude of the 3-space point corresponding to $\mathbf{x}$ and $\mathbf{x}'$ is approximated by the parameter h for which $\mathrm{epi}^{\mathbf{x}}_{uv}(h)$ is the orthogonal projection of $\mathbf{x}'$ on the curve.

model itself may not agree with the images, as was seen in chapter 2. Given a pair of corresponding points $\mathbf{x}$ and $\mathbf{x}'$ in two images, the epipolar curve of $\mathbf{x}$ may not pass through the point $\mathbf{x}'$, as illustrated in Figure 4.2. We call this error the *relative pointing error*. It is not negligible at all, being often of the order of tens of pixels, as shown by the numerical measurements reported in Table 4.1.

4.2.3 How to measure the relative pointing error

Given two images u and v and a set of correspondences $(\mathbf{x}_i, \mathbf{x}'_i)_{i=1\dots n}$, the relative pointing error between u and v is formally defined by

$$\frac{1}{n}\sum_{i=1}^{n} d(\mathbf{x}'_i, \mathrm{epi}^{\mathbf{x}_i}_{uv}(\mathbf{R})). \tag{4.3}$$

Here $\mathrm{epi}^{\mathbf{x}_i}_{uv}(\mathbf{R})$ is the epipolar curve of point $\mathbf{x}_i$, and d is the Euclidean distance, measured in pixels, between a point and a subset of $\mathbf{R}^2$. The set of correspondences between two images can be determined using SIFT [Lowe 2004, Rey Otero and Delbracio 2014]. Table 4.1 gives values for the relative pointing error measured on several Pléiades and WorldView-1 stereo pairs.

Dataset	error on small tile (pix)	error on large tile (pix)	Altitude range width (m)
calanques	0.61	0.58	182
cannes	4.33	3.83	151
giza	0.63	0.39	57
lenclio	1.88	1.17	55
mera	8.47	8.13	1097
mercedes_1	2.01	2.07	19
mercedes_2	2.18	2.14	21
mont_blanc_1	2.23	2.13	466
mont_blanc_2	3.25	3.05	1159
montevideo	0.16	0.22	18
new_york	0.17	0.46	40
ossoue	1.02	0.80	540
reunion_1	1.13	3.34	79
reunion_3	0.99	1.07	72
reunion_4	0.80	0.77	28
reunion_5	0.72	0.67	349
spitsberg	1.16	1.12	610
toulouse	0.92	0.66	4
ubaye	0.27	0.31	220
ambedkar*	2.55	–	9
charbagh*	2.35	–	16
mean	1.80	1.57	–

Table 4.1 – Pointing error values measured on each stereo pair for two tile sizes: *small* is 1000×1000 pixels and *large* is $25\,000 \times 25\,000$ pixels. Datasets marked with an asterisk (*) are from WorldView-1. The others are from Pléiades 1A and 1B.

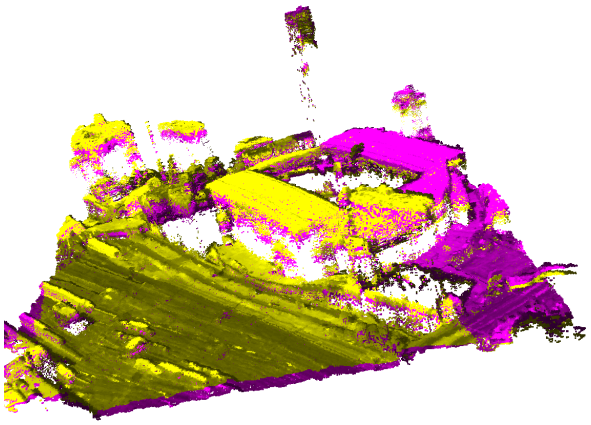


Figure 4.3 – Consequence of the satellite relative pointing error. Two point clouds computed from two image pairs of the Melbourne stadium while keeping the same reference image. One is colored in yellow and the other in pink. They do not fit exactly because of the residual pointing error in the direction of the epipolar lines. But these clouds are easy to register.

4.3 Correction of the Relative Pointing Error

4.3.1 Not absolute but automatic correction

The error affecting RPC functions is well known [Fraser and Hanley 2003]. It comes from inaccuracies in the sensor attitude estimation and also affects the rigorous model, as was discussed in chapter 2. Thus the RPC approximation is not to blame for it [Fraser and Hanley 2005]. This error is absolute. It can be evidenced with a single image u and a unique ground control point $\mathbf{X}$ by observing that $L_u(\mathbf{X})$ is not exactly located on the actual image of $\mathbf{X}$. Several authors have modeled this absolute error and proposed methods to compensate it [Hanley et al. 2002, Fraser and Hanley 2003, Grodecki and Dial 2003, Fraser and Hanley 2005]. All these methods need ground control points and manual interactions, thus are not suitable in a fully automatic 3D reconstruction pipeline.

The relative pointing error is measured using two images and can be corrected without any control points. This correction will not remove the absolute error affecting the RPC projection and localization functions, but will allow to perform efficient stereo matching between the views by following epipolar curves. Moreover, this correction procedure does not require any manual interaction and can be integrated in a fully automatic pipeline to compute digital elevation models. However, as the correction is relative to a given image pair, 3D point clouds obtained from different image pairs may not be correctly registered, as illustrated by Figure 4.3.

4.3.2 Relative pointing error model

As mentioned in chapter 2, and previously reported by Fraser and Hanley [2005], errors within the direct measurement of sensor orientation reside mainly in sensor attitude. For an image tile of size 1000×1000 pixels, covering a scene of size 500×500 m on the ground (with Pléiades resolution), we can assume that the scene is located at infinity with respect to the satellite. The error can then be modeled in image space as a translation composed with a rotation.

A simple way to correct the relative pointing error is thus to transform one of the two images in such a way that the corresponding points fall on the respective epipolar curves: given two images u, v and a set of correspondences $(\mathbf{x}_i, \mathbf{x}'_i)_{i=1\dots n}$, we search for a translation $\mathtt{T}$ and a rotation $\mathtt{R}$ centered in the center of the tile such that, for all i, the transformed point $\mathtt{RT}\mathbf{x}'_i$ lies on the epipolar curve $\text{epi}_{uv}^{\mathbf{x}_i}(\mathbf{R})$. The desired translation $\mathtt{T}^*$ and rotation $\mathtt{R}^*$ minimize the relative pointing error defined by equation 4.3:

$$\mathtt{R}^*, \mathtt{T}^* = \arg\min_{\mathtt{R},\mathtt{T}} \frac{1}{n} \sum_{i=1}^{n} d(\mathtt{RT}\mathbf{x}'_i, \text{epi}_{uv}^{\mathbf{x}_i}(\mathbf{R})). \tag{4.4}$$

The goal of the relative pointing error correction is to improve the precision of the epipolar constraints derived from the RPC functions. Epipolar constraints are used to rectify the images for efficient stereo matching. But as was shown in chapter 3, stereo-rectification of pushbroom images is possible only on small image tiles. For Pléiades images an appropriate tile size was found to be 1000×1000 pixels. On such small tiles the rotation component $\mathtt{R}$ of the pointing error is not visible. The correction can thus be performed with a translation

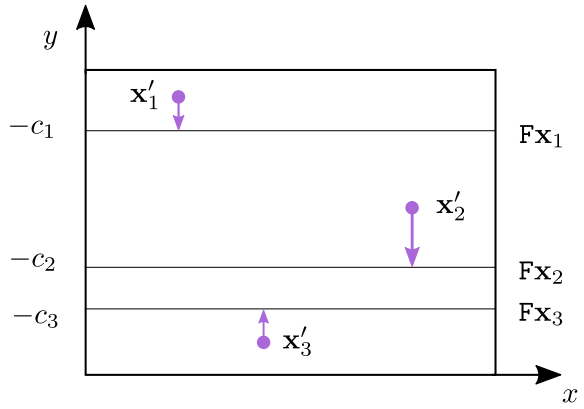


Figure 4.4 – For a tile of size 1000×1000, the epipolar curves are well approximated by parallel lines (see chapter 3). On this figure the lines are assumed to be horizontal. For each correspondence $(\mathbf{x}_i, \mathbf{x}'_i)$ there is a vertical shift between the point $\mathbf{x}'_i$ and the line $\mathtt{F}\mathbf{x}_i$. The median of all these shifts minimizes the relative pointing error defined by formula (4.3).

only. The minimization can hence be simplified and becomes:

$$\mathtt{T}^* = \arg\min_{\mathtt{T}} \frac{1}{n} \sum_{i=1}^{n} d(\mathtt{T}\mathbf{x}'_i, \mathrm{epi}_{uv}^{\mathbf{x}_i}(\mathbf{R})). \tag{4.5}$$

4.3.3 Correction algorithm

From chapter 3 we know that the epipolar curve $\mathrm{epi}_{uv}^{\mathbf{x}_i}(\mathbf{R})$ is approximated up to 0.05 pixels by the straight line $\mathtt{F}\mathbf{x}_i$, where $\mathtt{F}$ is the affine fundamental matrix of the two views for the considered tile. As this fundamental matrix is an *affine* fundamental matrix, all the lines $\mathtt{F}\mathbf{x}_i$ are parallel. Without any additional restriction, we may assume that these lines are horizontal (otherwise just do a change of coordinates). The horizontal line $\mathtt{F}\mathbf{x}_i$ can be written, in homogeneous coordinates, as

$$\mathtt{F}\mathbf{x}_i = \begin{bmatrix} 0 \\ 1 \\ c_i \end{bmatrix}. \tag{4.6}$$

With these notations, for each point correspondence $(\mathbf{x}_i, \mathbf{x}'_i)$ we have

$$d(\mathbf{x}'_i, \mathtt{F}\mathbf{x}_i) = |y'_i + c_i|, \tag{4.7}$$

where $\mathbf{x}'_i = (x'_i, y'_i, 1)^\top$. The situation is illustrated in figure 4.4. This error is invariant to horizontal translations, thus the search for a translation minimizing the relative pointing error defined in formula (4.3) can be restricted to vertical translations only. With a vertical translation of parameter t, the error becomes

$$\frac{1}{n} \sum_{i=1}^{n} d(\mathtt{T}\mathbf{x}'_i, \mathtt{F}\mathbf{x}_i) = \frac{1}{n} \sum_{i=1}^{n} |y'_i + t + c_i|. \tag{4.8}$$

The translation that minimizes this sum is given by the geometric median [Weiszfeld 1937] of the vectors $(-y'_i - c_i)_{i=1,\dots,n}$, as shown by the next lemma.

Lemma 2. *Let $a_1, a_2, \ldots, a_n$ be real numbers. The function $f : \mathbf{R} \to \mathbf{R}$ defined by*

$$f(t) = \sum_{i=1}^{n} |a_i - t| \tag{4.9}$$

is minimal when t is equal to the median of the a_i $(i = 1, \ldots, n)$.

The relative pointing error can thus be minimized in a tile by applying a translation to one of the images. The translation is estimated from a set of point correspondences by computing a median. Note that the median is robust against outliers, thus this correction procedure works well even in the presence of false matches.

4.4 Experimental Results

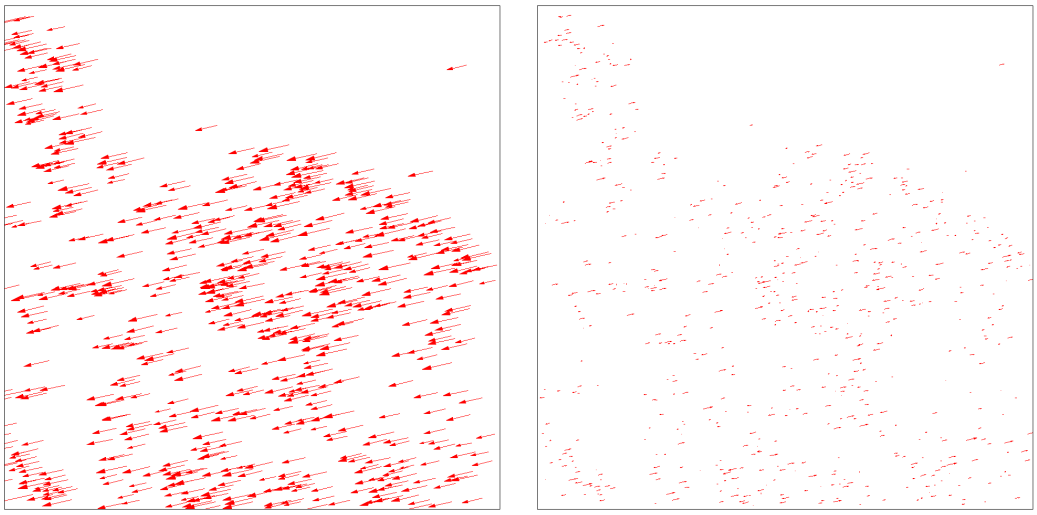
The relative pointing error, as defined by equation (4.3), was measured on several Pléiades and a few WorldView-1 datasets. For each dataset the measurement was made with tiles sizes of 1000×1000 pixels (small) and $25\,000 \times 25\,000$ pixels (large). The tiles were located in the center of the reference image. The matches used to compute the relative pointing errors were obtained using SIFT [Lowe 2004, Rey Otero and Delbracio 2014]. Figure 4.5 shows the displacements that should be applied to the matching points of the second image to make them fit on the corresponding epipolar curves, before and after correction. Table 4.2 gives the measured values before and after correction, for both the tile sizes (small and large). The correction was performed by computing the optimal translation in image space as described in section 4.3.3. The results show that, on average, for tiles of size 1000×1000, the relative pointing error is reduced by a factor 10 and the residual error is always smaller than half a pixel. On the contrary, for large tiles of size $25\,000 \times 25\,000$ pixels, the error is reduced on average by a factor 2, but the residual error may still be greater than 1 pixel, and in some cases the correction increases the error. There are two explanations for that:

- the pointing error is not well approximated by a translation in image space for such large tiles,
- the epipolar geometry is not well approximated by an affine fundamental matrix. Indeed the results of chapter 3 show that the approximation is not valid for such large tiles.

For these large tiles, we included the rotation term in the error model, as in equation 4.4. This led to an almost null rotation and almost the same residual errors. This indicates that for large tiles the error cannot be modeled by a rigid transformation in the image space. Nevertheless, this study shows that for stereo reconstruction purposes, which require only small tiles, the pointing error correction is satisfactory.

4.5 Conclusion

We presented a method to automatically correct the relative error that exists in the RPC coefficients associated to two views of any high resolution satellite stereo dataset. This method reduces the relative error by a factor 10, lowering it under 0.2 pixels, thus allowing



(a) Error vectors before correction.

(b) Error vectors after correcting the position of the second image by the optimal translation given by equation (4.5).

Figure 4.5 – Error vectors for some keypoints on a 1000×1000 tile of a Pléiades image.

a very precise stereo-rectification. This pointing error correction method is implemented in the *Satellite Stereo Pipeline* (`S2P`), which will be presented in chapter 6. Future work should investigate more complex error models for entire images. Independently of the error model, the correction could be further improved by refining the position of the matches using techniques such as *least square matching* [Gruen 1985, Liu et al. 2014].

Dataset	Small tile error (pix)		Large tile error (pix)		Altitude range width (m)
	before	after	before	after	
calanques	0.61	0.14	0.58	0.41	182
cannes	4.33	0.12	3.83	0.36	151
giza	0.63	0.09	0.39	0.40	57
lenclio	1.88	0.12	1.17	0.44	55
mera	8.47	0.29	8.13	0.38	1097
mercedes_1	2.01	0.15	2.07	0.31	19
mercedes_2	2.18	0.13	2.14	0.18	21
mont_blanc_1	2.23	0.15	2.13	1.35	466
mont_blanc_2	3.25	0.21	3.05	1.27	1159
montevideo	0.16	0.09	0.22	0.15	18
new_york	0.17	0.10	0.46	0.73	40
ossoue	1.02	0.36	0.80	0.43	540
reunion_1	1.13	0.10	3.34	2.49	79
reunion_3	0.99	0.10	1.07	0.17	72
reunion_4	0.80	0.12	0.77	0.21	28
reunion_5	0.72	0.13	0.67	0.19	349
spitsberg	1.16	0.27	1.12	4.58	610
toulouse	0.92	0.14	0.66	0.29	4
ubaye	0.27	0.17	0.31	0.36	220
ambedkar*	2.55	0.16	–	–	9
charbagh*	2.35	0.48	–	–	16
mean	1.80	0.17	1.57	0.7	–

Table 4.2 – Pointing error values before and after correction, for two tile sizes: *small* is 1000×1000 pixels and *large* is $25\,000 \times 25\,000$ pixels. On average, the proposed algorithm allows to reduce the error by a factor 10 on the small tiles, and a factor 2 on the large tiles. Thus the correction is used only for small tiles. Datasets marked with an asterisk (∗) are from WorldView-1. As the images from these datasets are smaller than the large tile, only the small tile was used. The other datasets are from Pléiades 1A and 1B.

5 Stereo Processing by Semi-Global Matching and Beyond

Semi-Global Matching (SGM) is among the top-ranked stereo vision algorithms. SGM is an efficient strategy for approximately minimizing a global energy that comprises a pixel-wise matching cost and pairwise smoothness terms. In SGM the two-dimensional smoothness constraint is approximated as the average of one-dimensional line optimization problems. The accuracy and speed of SGM are the main reasons for its widespread adoption, even when applied to generic problems beyond stereo vision. However this approximate minimization produces characteristic low amplitude streaks in the final disparity image, and is clearly suboptimal with respect to more comprehensive minimization strategies. This chapter gives a detailed description of the algorithm, and analyzes the approximations made by SGM with respect to the underlying Markov Random Field (MRF). Based on a recent interpretation of SGM as a min-sum Belief Propagation algorithm, we propose a new algorithm that allows to reduce by a factor five the energy gap of SGM with respect to reference algorithms for MRFs with truncated smoothness terms. The proposed method comes with no compromises with respect to the baseline SGM, no parameters and virtually no computational overhead. At the same time it attains higher quality results by removing the characteristic streaking artifacts of SGM.

Implementations of the SGM and MGM algorithms presented in this chapter can be tested online on any stereo pair at `http://dev.ipol.im/~facciolo/ipol_demo/mgm`.

5.1 Introduction

Stereo vision estimates the depth of a scene from two or more images taken from slightly different viewpoints. This is done by computing the apparent motion of the scene points between the views. In the case of *stereo-rectified* image pairs, such as those produced by the Algorithm 6 presented in chapter 3, this motion is referred to as *disparity*.

Stereo matching methods are traditionally divided into *local* and *global methods* [Scharstein and Szeliski 2002]. Local methods estimate the disparity independently for each pixel by comparing features (usually a window around the pixel) of the left and right image. Local methods are computationally cheap, however they suffer from several artifacts among which the well-known *foreground fattening effect* near depth discontinuities. Numerous methods have been proposed to overcome this effect by adapting the shape or the weight distribution of the window to the local image content, such as [Yoon and Kweon 2006] which has been analyzed and implemented by Julià and Monasse [2015]. But in any case if the comparison window falls on an ambiguous area (lacking texture or with repetitive patterns) the estimated disparity will likely be incorrect.

Global methods cope with these ambiguities by imposing the smoothness of the disparity map, which permits to derive reasonable estimates even in the ambiguous areas. Global methods model the correspondence estimation as a global minimization usually associated

with a Markov Random Field (MRF) formulation. The resulting energy functional is a sum of unary matching terms and smoothness terms that force adjacent pixels to have similar disparities. While this problem is NP-hard, strong local minima can be computed [Boykov et al. 2001] and many techniques have been developed to that effect, such as move-making approaches [Boykov et al. 2001, Kolmogorov and Zabih 2001, Kolmogorov et al. 2014] and message-passing methods [Felzenszwalb and Huttenlocher 2006, Kolmogorov 2006]. Their effectiveness has been reviewed by Szeliski et al. [2008]. Most of these techniques, however, are too slow when applied to images of a reasonable size such as those of the Middlebury stereo evaluation dataset (version 3) [Scharstein et al. 2014].

Because of the ever increasing size of the problems, fast approximate algorithms producing reasonable solutions are currently preferred to global techniques [Hirschmüller 2008, Sinha et al. 2014, Hosni et al. 2013, Geiger et al. 2011, Cech and Sara 2007, Besse et al. 2014, Chen and Koltun 2014, Komodakis et al. 2007]. Among these approximate algorithms, *Semi-Global Matching* (SGM) [Hirschmüller 2008] is nowadays one of the preferred choices for stereo vision applications [Hermann and Klette 2013, Zhu et al. 2012] because of its efficiency and good performance, and it has even been applied to problems beyond disparity estimation, for example, to approximately solve the Potts model [Sinha et al. 2014].

In SGM the two-dimensional smoothness constraint is efficiently approximated as the average of one-dimensional line optimization problems. This approximation reduces at each pixel to an optimization on a star-shaped graph (usually with 8 cardinal directions) centered at the pixel. This structure favors the occurrence of piecewise constant solutions along these directions. However, two adjacent scan lines share little information, so if the data term is weak, their solutions can differ, yielding the well known streaking artifacts of SGM.

The success of the SGM algorithm relies on several heuristics. In this chapter we clarify some of them by analyzing SGM in the light of its recently proposed interpretation as a min-sum Belief Propagation algorithm by Drory et al. [2014]. This leads us to propose a new version of the algorithm that overcomes some of its limitations. Our principled interpretation of SGM reduces by a factor five its energy gap with respect to reference optimization algorithms for MRFs with truncated smoothness terms.

Extensive evaluation shows that the proposed algorithm removes all streaking artifacts, improves the visual quality of the result, and computes efficiently approximate solutions to large MRF problems. All this is achieved while keeping the flexibility of SGM, without introducing new parameters and with a computing overhead of at most 20%.

Previous work. The strategies for finding a minimum of the global energy function differ. Many methods simplify the 2D graph by reducing it to simpler subproblems. But these simplifications are critical as they determine the flow of information. Dynamic programming approaches [Ohta and Kanade 1985b] perform the optimization in one dimension for each scan line individually, which commonly leads to streaking effects. This is avoided by tree-based dynamic programming approaches. Some methods derive, from the 2D graph, a single tree that spans the entire image [Veksler 2005]. Others construct trees that vary their grid structure with the position of the pixel [Bleyer and Gelautz 2008]. The Fast-PD algorithm [Komodakis et al. 2007] exploits information from the original MRF problem and its dual resulting in a remarkable speed-up with respect to alpha-expansions. In [Chen and Koltun 2014] a very efficient block coordinate descent is used to approximately solve MRFs

with truncated smoothness terms. In SGM [Hirschmüller 2008] the optimization is restricted to a star-shaped graph centered at the current pixel, which sometimes results in streaking artifacts over poorly textured areas. Increasing the number of paths (i.e. 16 directions) may suppress them but doubles the computational burden [Hirschmüller 2008].

The technique we propose aims at increasing the support of SGM's graph without extra overhead, by leveraging its connection with message passing algorithms [Drory et al. 2014].

5.2 The Semi-Global Matching algorithm

The Semi-Global Matching (SGM) algorithm [Hirschmüller 2008] for dense stereo matching is an efficient tool to perform approximate minimization of energies for 2D Markov Random Fields (MRF). Let us denote by $G = (\mathcal{V}, \mathcal{E})$ the image graph associated to the first view of the stereo pair, where $\mathcal{V}$ denotes the set of vertices, i.e. pixels, and $\mathcal{E}$ denotes the set of edges, i.e. neighborhood relations. The graph generally used is the 4-connected or the 8-connected image graph. The stereo matching problem is then formulated as finding the disparity map D that minimizes the energy E defined on G by

$$E(\mathrm{D}) = \sum_{\mathbf{p} \in \mathcal{V}} C(\mathbf{p}, \mathrm{D}_{\mathbf{p}}) + \sum_{(\mathbf{p},\mathbf{q}) \in \mathcal{E}} V(\mathrm{D}_{\mathbf{p}}, \mathrm{D}_{\mathbf{q}}). \tag{5.1}$$

Here $C(\mathbf{p}, d)$ is a unary data term that represents the pixel-wise cost of matching $\mathbf{p}$ with disparity $d \in \mathcal{D}$, where $\mathcal{D} = \{d_{\min}, \dots, d_{\max}\}$ is the search space, while $\mathrm{D}_{\mathbf{p}}$ denotes the value of the disparity map D at pixel $\mathbf{p}$. The pairwise terms $V(\mathrm{D}_{\mathbf{p}}, \mathrm{D}_{\mathbf{q}})$ enforce smoothness of the solution by penalizing changes of neighboring disparities on the edge set $\mathcal{E}$. SGM considers a particular type of pairwise terms of the form

$$V(d, d') = \begin{cases} 0 & \text{if } d = d' \\ P_1 & \text{if } |d - d'| = 1 \\ P_2 & \text{otherwise} \end{cases} . \tag{5.2}$$

This imposes a small penalty P_1 for small jumps in disparity (up to one pixel), which are common on slanted surfaces. The second penalty term P_2 (with $P_2 > P_1$) accounts for larger disparity jumps. The penalty P_2 can be further adapted [Hirschmüller 2008, Bleyer and Gelautz 2008, Sinha et al. 2014] depending on the image content to align the disparity discontinuities with the discontinuities in the image.

Remark. *In the original papers [Hirschmüller 2005, 2008] the energy (5.1) is written with pairwise terms $V(\mathrm{D}_{\mathbf{p}}, \mathrm{D}_{\mathbf{q}})$ for all neighbors of all pixels. Thus each edge of $\mathcal{E}$ is counted twice (one time for each endpoint of the edge). This does not have any consequences, except for the fact that the numerical values of P_1 and P_2 have to be doubled here to obtain the same energy.*

The exact minimization of energy (5.1) is NP-hard, as proved in section 5.2.1. The SGM algorithm extends single-line dynamic stereo matching [Ohta and Kanade 1985b] into a multi-line integration strategy along multiple directions which leads to an approximate solution of the NP-hard problem (5.1). The strategy adopted by SGM consists in dividing the grid-shaped problem into multiple one-dimensional problems defined on scanlines, which

are straight lines that run through the image in the 8 cardinal directions. For each cardinal direction $\mathbf{r}$, SGM computes a matrix of costs $L_{\mathbf{r}}$. The costs $L_{\mathbf{r}}(\mathbf{p}, d)$ are computed recursively, starting from the image borders, along a path in the direction $\mathbf{r}$:

$$L_{\mathbf{r}}(\mathbf{p}, d) = C(\mathbf{p}, d) + \min_{d' \in \mathcal{D}}(L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d') + V(d, d')). \tag{5.3}$$

This recursion is in fact a dynamic programming algorithm that solves the problem restricted to the directed graph induced by the scanline $\mathbf{p} - \mathbb{N}\mathbf{r} = \{\mathbf{p} - k\mathbf{r} | k \in \mathbb{N}\}$. Because of the special form of the smoothness potential (5.2), $L_{\mathbf{r}}(\mathbf{p}, \cdot)$ can be computed with just 7 instructions per disparity, as will be shown in section 5.2.3. These costs are computed in each direction $\mathbf{r}$ and are then added to obtain an aggregated cost volume

$$S(\mathbf{p}, d) = \sum_{\mathbf{r}} L_{\mathbf{r}}(\mathbf{p}, d). \tag{5.4}$$

The final disparity for each pixel is then selected by minimizing the aggregated cost $S(\mathbf{p}, d)$ with respect to d.

5.2.1 Energy minimization is NP-hard

The NP-hardness proof is adapted from [Boykov et al. 2001]. It is based on the reduction in polynomial time of the NP-hard multiway cut problem, that we shall first present.

The multiway cut problem. Let $G = (\mathcal{V}, \mathcal{E})$ be a graph. Consider $\mathcal{T} \subseteq \mathcal{V}$ a set of vertices, called *terminals*. A *multiway cut*, sometimes referred to as a *cut* for short, is a set of edges $\mathcal{C} \subseteq \mathcal{E}$ verifying:

- The terminals are *separated* by $\mathcal{C}$, i.e. each terminal is in a distinct connected component in the induced graph $(\mathcal{V}, \mathcal{E} - \mathcal{C})$.
- No proper subset of $\mathcal{C}$ separates the terminals.

The *weight* of a cut is defined as its cardinal, i.e. the number of edges contained in the cut. The multiway cut problem is to find a cut of minimal weight. An example is shown on Figure 5.1. If the graph is not planar, which is the case for the 8-connected image graph as soon as the image is larger than 3×3 pixels, or if the number of terminals is not fixed, then this problem is NP-hard [Dahlhaus et al. 1992].

With $P_1 = 0$ and $P_2 = 1$ the pairwise term $V(\mathrm{D}_{\mathbf{p}}, \mathrm{D}_{\mathbf{q}})$ defined in (5.2) is equal to $\mathbf{1}_{\mathbb{R}^*}(\mathrm{D}_{\mathbf{p}} - \mathrm{D}_{\mathbf{q}})$, thus in that particular case the energy (5.1) is a Potts energy. But even in that simpler case, the minimization problem is NP-hard.

Result 5. *Minimizing a Potts energy is NP-hard.*

Proof. The plan of the proof is the following: given an arbitrary graph G, we build, in polynomial time, a Potts energy E_G such that a minimal cut on G can be determined from $\mathrm{D}^* = \arg\min E_G$ in polynomial time. As the multiway cut problem is NP-hard, minimizing a Potts energy is NP-hard.

For not connected graphs, a multiway cut is just the union of multiway cuts on each connected component, so in the following we shall consider only connected graphs.

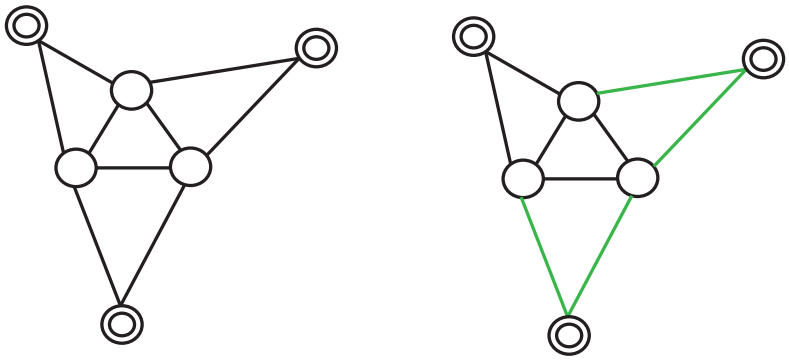


Figure 5.1 – A graph G in which the terminals are surrounded. On the right, a minimal multiway cut on G is shown in green.

Let $G = (\mathcal{V}, \mathcal{E})$ be a graph with $\mathcal{T} = \{\boldsymbol{t}_1, ..., \boldsymbol{t}_k\} \subseteq \mathcal{V}$ the set of terminals. Considering the search space $\mathcal{D} = \{1, ..., k\}$, let define the energy of a disparity map D as:

$$E_G(\mathrm{D}) = \sum_{\mathbf{p} \in \mathcal{V}} C(\mathbf{p}, \mathrm{D}_\mathbf{p}) + \sum_{(\mathbf{p},\mathbf{q}) \in \mathcal{E}} \mathbf{1}_{\mathbb{R}^*}(\mathrm{D}_\mathbf{p} - \mathrm{D}_\mathbf{q}), \tag{5.5}$$

where

$$C(\mathbf{p}, j) = \begin{cases} |\mathcal{E}| & \text{if } \mathbf{p} = \boldsymbol{t}_i \text{ and } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

As $|\mathcal{E}|$ is an upper bound for the regularity term $\sum_{(\mathbf{p},\mathbf{q}) \in \mathcal{E}} \mathbf{1}_{\mathbb{R}^*}(\mathrm{D}_\mathbf{p} - \mathrm{D}_\mathbf{q})$, this definition forces the minimizing disparity map D^* to verify $\mathrm{D}^*_{\boldsymbol{t}_j} = j$ for all $j \in \mathcal{D}$, and if $\mathbf{p}$ is not a terminal, $C(\mathbf{p}, j) = 0$ means that there is no privileged disparity for $\mathbf{p}$.

A map D is said *feasible* if, for all $j \in \{1, \ldots, k\}$, $\mathrm{D}_{\boldsymbol{t}_j} = j$ and $\mathrm{D}^{-1}(j)$ form a connected component of G (i.e. the graph $(\mathrm{D}^{-1}(j), \mathcal{E} \cap (\mathrm{D}^{-1}(j) \times \mathrm{D}^{-1}(j)))$ is connected). Yet, D^* is feasible. Indeed, suppose there is a disparity j such that $\mathrm{D}^{-1}(j)$ does not form a connected component of G. Let S be a connected component of $\mathrm{D}^{-1}(j)$, not containing $\boldsymbol{t}_j$. As G is connected and $\mathrm{D}(\Omega) \neq \{j\}$, there is a vertex $\mathbf{p}_0 \in S$ that is connected to a certain $\mathbf{q}_0$ of disparity $i \neq j$. Then the disparity map D^- defined by

$$\mathrm{D}^-(\mathbf{p}) = \begin{cases} i & \text{if } \mathbf{p} \in S \\ \mathrm{D}^*(\mathbf{p}) & \text{otherwise} \end{cases} \tag{5.7}$$

verifies $E(\mathrm{D}^-) \leq E(\mathrm{D}^*) - 1 < E(\mathrm{D}^*)$, which contradicts the minimality of D^*.

Lemma 3 states that there is a one-to-one correspondence between feasible maps and multiway cuts such that the weight of a cut is the energy of the corresponding map, and this correspondence can be computed in polynomial time. Hence, once D^* is known, a solution to the multiway cut problem can be computed in polynomial time. □

Lemma 3. *There is a one-to-one correspondence between feasible maps and multiway cuts, such that the weight of a cut is the energy of the corresponding map.*

Proof. Let us consider a feasible map D. Call $\mathcal{C}_\mathrm{D} = \{(\mathbf{p},\mathbf{q}) \in \mathcal{E} \mid \mathrm{D}_\mathbf{p} \neq \mathrm{D}_\mathbf{q}\}$. As D is feasible, $\mathcal{C}_\mathrm{D}$ separates the terminals. Furthermore $\mathcal{C}_\mathrm{D}$ has no proper subset that separates the terminals. Indeed, if $\mathcal{C}' \subsetneq \mathcal{C}_\mathrm{D}$, there is an edge $(\mathbf{p},\mathbf{q})$ such that $i = \mathrm{D}_\mathbf{p} \neq \mathrm{D}_\mathbf{q} = j$ but $(\mathbf{p},\mathbf{q}) \notin \mathcal{C}'$. Thus, terminals $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$ are in the same connected component and $\mathcal{C}'$ does not separate the terminals. Thus $\mathcal{C}_\mathrm{D}$ is a multiway cut, and its weight is

$$|\mathcal{C}_\mathrm{D}| = \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{C}_\mathrm{D}} 1 = \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{E}} \mathbf{1}_{\mathbb{R}^*}(\mathrm{D}_\mathbf{p} - \mathrm{D}_\mathbf{q}) = E_G(\mathrm{D}) \tag{5.8}$$

where the last equality comes from the fact that all the unary cost terms $C(\mathbf{p},\mathrm{D}_\mathbf{p})$ are equal to zero.

Conversely, if $\mathcal{C}$ is a multiway cut, as G is connected each pixel $\mathbf{p}$ is in the connected component of a single terminal $\boldsymbol{t}_j$ in the induced graph $\mathcal{V},\mathcal{E}-\mathcal{C}$. Define the disparity map $\mathrm{D}_\mathcal{C}$ by $\mathrm{D}_\mathcal{C}(\mathbf{p}) = j$. Then obviously $\mathcal{C}_{\mathrm{D}_\mathcal{C}} = \mathcal{C}$ and $\mathrm{D}_{\mathcal{C}_\mathrm{D}} = \mathrm{D}$ which proves that $\mathrm{D} \mapsto \mathcal{C}_\mathrm{D}$ is bijective. □

5.2.2 1D Energy minimization is linear

For the usual 4-connected or 8-connected image graph the energy minimization problem (5.1) is NP-hard. But if the image graph is one dimensional, i.e. if $\mathcal{V} = \{1,\cdots,n\}$ and the set of edges $\mathcal{E}$ contains only pairs of consecutive vertices, then the minimization problem is easier. In this subsection we study this particular case. To emphasize the fact that the image domain has only one dimension, pixels (or graph vertices) are denoted by italic letters such as p instead of $\mathbf{p}$. The energy (5.1) can be rewritten as

$$E(\mathrm{D}) = \sum_{p=1}^{n} C(p,\mathrm{D}_p) + \sum_{p=0}^{n} V(\mathrm{D}_p,\mathrm{D}_{p+1}) \tag{5.9}$$

where $(\mathrm{D}_0,\mathrm{D}_{n+1}) = (\mathrm{D}_1,\mathrm{D}_n)$, in order to make the extra terms $V(\mathrm{D}_0,\mathrm{D}_1)$ and $V(\mathrm{D}_n,\mathrm{D}_{n+1})$ be zero. Given $p \in \{1,\ldots,n\}$, the two sums can be split in two and reorganized as

$$E(\mathrm{D}) = \underbrace{\sum_{q=1}^{p}(C(q,\mathrm{D}_q) + V(\mathrm{D}_{q-1},\mathrm{D}_q))}_{E_{p,+}(\mathrm{D})} + \underbrace{\sum_{q=p}^{n}(C(q,\mathrm{D}_q) + V(\mathrm{D}_q,\mathrm{D}_{q+1}))}_{E_{p,-}(\mathrm{D})} - C(p,\mathrm{D}_p). \tag{5.10}$$

The first term $E_{p,+}$ of this sum is a function that actually depends only on $\mathrm{D}_1,\ldots,\mathrm{D}_p$. It can be minimized with respect to $\mathrm{D}_1,\ldots,\mathrm{D}_{p-1}$ in linear time with a dynamic programming algorithm. The value of the minimum, which depends on p and D_p, is stored in the *aggregated cost table* denoted by L_+. It is computed by algorithm 7. Note that the term $\min_i L_+(p-1,i)$ intervening in the recursion formula (5.11) needs to be computed only once for each value of p. The complexity of the algorithm is hence $O(nk)$.

Result 6. *The aggregated cost table L_+ computed by algorithm 7 verifies*

$$L_+(p,d) = \min_{\mathrm{D},\mathrm{D}_p=d} E_{p,+}(\mathrm{D}) \tag{5.12}$$

for all $p \in \{1,\ldots,n\}$ and $d \in \{d_{min},\ldots,d_{max}\}$.

Algorithm 7: Directional cost aggregation in 1D to minimize term $E_{p,+}$ defined in formula (5.10)

Input: cost table C of size $n \times k$
set of disparities $\mathcal{D} = \{d_{\min}, \dots, d_{\max}\}$
Output: aggregated cost table L_+ of size $n \times k$
begin
  for $d \in \mathcal{D}$ **do** *// initialization*
    $L_+(1, d) \leftarrow C(1, d)$
  for $p \in \{2, \dots, n\}$ **do**
    for $d \in \mathcal{D}$ **do** *// aggregation*

$$\begin{aligned} L_+(p,d) \leftarrow C(p,d) + \min(&L_+(p-1,d), \\ &L_+(p-1,d+1) + P_1, \\ &L_+(p-1,d-1) + P_1, \\ &\min_i L_+(p-1,i) + P_2) \end{aligned} \tag{5.11}$$

Proof. The result is proved by induction on p. Any d verifies $L_+(1,d) = C(1,d) = C(1,d) + V(\mathrm{D}_1, \mathrm{D}_0) = E_{1,+}(d)$ for all disparity maps D such as $\mathrm{D}_1 = d$. This proves the equality (5.12) for $p = 1$.

If the result (5.12) holds for $p - 1 \geq 1$, then by replacing $L_+(p-1, \cdot)$ in (5.11) with the induction hypothesis (5.12) we have:

$$\begin{aligned} L_+(p,d) = C(p,d) + \min\Big(&\min_{\mathrm{D},\mathrm{D}_{p-1}=d} E_{p-1,+}(\mathrm{D}), \\ &\min_{\mathrm{D},\mathrm{D}_{p-1}\in\{d\pm1\}} E_{p-1,+}(\mathrm{D}) + P_1, \\ &\min_i \min_{\mathrm{D},\mathrm{D}_{p-1}=i} E_{p-1,+}(\mathrm{D}) + P_2\Big). \end{aligned} \tag{5.13}$$

In the first two arguments of the most external min operator, the value of D_{p-1} is fixed thus $V(\mathrm{D}_{p-1}, d)$ is equal to zero and P_1 respectively. In the third argument, the value of D_{p-1} is not fixed thus we only know that $V(\mathrm{D}_{p-1}, d) \leq P_2$. But if the value of the third argument is obtained with a disparity map such that $V(\mathrm{D}_{p-1}, d) < P_2$, then $V(\mathrm{D}_{p-1}, d)$ is equal to either P_1 or zero, and then the third argument is equal to one of the first two arguments. Hence

$L_+(p,d)$ can be rewritten as

$$\begin{aligned} L_+(p,d) &= C(p,d) + \min\Big(\min_{\mathrm{D},\mathrm{D}_{p-1}=d} E_{p-1,+}(\mathrm{D}) + V(\mathrm{D}_{p-1},d), \\ &\qquad \min_{\mathrm{D},\mathrm{D}_{p-1}\in\{d\pm1\}} E_{p-1,+}(\mathrm{D}) + V(\mathrm{D}_{p-1},d), \\ &\qquad \min_{\mathrm{D}} E_{p-1,+}(\mathrm{D}) + V(\mathrm{D}_{p-1},d)\Big) \\ &= C(p,d) + \min_{\mathrm{D}}\Big(E_{p-1,+}(\mathrm{D}) + V(\mathrm{D}_{p-1},d)\Big) \\ &= \min_{\mathrm{D}}\Big(E_{p-1,+}(\mathrm{D}) + C(p,d) + V(\mathrm{D}_{p-1},d)\Big) \end{aligned} \tag{5.14}$$

As the argument of the min does not depend on the value of D_p, this can be rewritten as

$$\begin{aligned} L_+(p,d) &= \min_{\mathrm{D},\mathrm{D}_p=d}\Big(E_{p-1,+}(\mathrm{D}) + C(p,\mathrm{D}_p) + V(\mathrm{D}_{p-1},\mathrm{D}_p)\Big) \\ &= \min_{\mathrm{D},\mathrm{D}_p=d} E_{p,+}(\mathrm{D}). \end{aligned} \tag{5.15}$$

This proves the result for p and hence ends the proof. □

In the same way the second term $E_{p,-}(\mathrm{D})$ of the sum (5.10) can be minimized with respect to $\mathrm{D}_{p+1},\dots,\mathrm{D}_n$ in linear time by computing the *aggregated cost table* L_- such that for all $p\in\Omega$ and $d\in\{d_{\min},\dots,d_{\max}\}$

$$L_-(p,d) = \min_{\mathrm{D},\mathrm{D}_p=d} E_{p,-}(\mathrm{D}). \tag{5.16}$$

The algorithm to compute L_- is analogous to algorithm 7, the only difference being the loop order that goes from n to 1 instead of going from 1 to n.

Then given $p\in\Omega$ and a disparity value d, the value of $\min_{\mathrm{D},\mathrm{D}_p=d} E(\mathrm{D})$ is obtained simply from L_+, L_- and C:

$$(L_+ + L_- - C)(p,d) = \min_{\mathrm{D},\mathrm{D}_p=d} E_{p,+}(\mathrm{D}) + \min_{\mathrm{D},\mathrm{D}_p=d} E_{p,-}(\mathrm{D}) - C(p,d) = \min_{\mathrm{D},\mathrm{D}_p=d} E(\mathrm{D}), \tag{5.17}$$

where the last equality is a consequence of formula (5.10) and of the fact that the two added minima actually operate on disjoint sets, namely $\{\mathrm{D}_1,\dots,\mathrm{D}_{p-1}\}$ and $\{\mathrm{D}_{p+1},\dots,\mathrm{D}_n\}$. If we assume that E has a unique global minimum on $\mathcal{D}^n$, it follows from lemma 4 and equality (5.17) that the minimum point can be computed componentwise as

$$\mathrm{D}_p = \arg\min_{\mathcal{D}} (L_+ + L_- - C)(p,\cdot). \tag{5.18}$$

The global minimization problem is hence reduced to a pointwise one, whose complexity is $O(nk)$.

Lemma 4. *Let $I_1,\dots,I_n$ be n finite sets, and $f: I_1\times\dots\times I_n\to\mathbb{R}_+$ a non-negative function. Assume that f has a unique global minimum point at $(x_1^*,\dots,x_n^*)$. Then, for any $1\le p\le n$ the function $g: I_p\to\mathbb{R}_+$ defined by*

$$g(x) = \min_{x_1,\dots,x_{p-1},x_{p+1},\dots,x_n} f(x_1,\dots,x_{p-1},x,x_{p+1},\dots,x_n) \tag{5.19}$$

has a unique global minimum point at x_p^*. Hence $x_p^* = \arg\min g$.

Figure 5.2 – Disparity map obtained on the Middlebury stereo pair *Cones* by minimizing the 1D energy (5.9) row by row. The streaking artifacts are due to the absence of regularity constraints in the vertical direction.

Proof. As all the I_k are finite sets the minimum used to define g is attained and $g(x) \geq f(x_1^*, \ldots, x_n^*)$ for all $x \in I_p$. But we have

$$g(x_p^*) = \min_{x_1,\ldots,x_{p-1},x_{p+1},\ldots,x_n} f(x_1, \ldots, x_{p-1}, x_p^*, x_{p+1}, \ldots, x_n) \leq f(x_1^*, \ldots, x_n^*). \tag{5.20}$$

Hence x_p^* is a global minimum point for g and $g(x_p^*) = f(x_1^*, \ldots, x_n^*)$. If g has another global minimum point x, then there exists $(\bar{x}_1, \ldots, \bar{x}_{p-1}, \bar{x}_{p+1}, \ldots, \bar{x}_n) \in \prod_{k=1,k\neq p}^{n} I_k$ such that

$$f(\bar{x}_1, \ldots, \bar{x}_{p-1}, x, \bar{x}_{p+1}, \ldots, \bar{x}_n) = g(x) = f(x_1^*, \ldots, x_n^*) \tag{5.21}$$

and as the global minimum point of f is unique it follows that $x = x_p^*$. □

Remark. *As the energy function E is real-valued and defined on a finite set, it has in general a unique global minimum.*

5.2.3 Aggregation of Costs in 1D from Several Directions

As minimization of the 2D energy (5.1) is NP-hard while minimization of the 1D energy (5.9) is linear, one could try to minimize energy (5.9) along individual image rows. This leads to streaking artifacts, as shown by figure 5.2. The problem is that nothing relates the disparities of an image row to those of the surrounding rows, as the non-horizontal interaction potentials of energy (5.1) have been removed.

The solution proposed by SGM

To solve this issue while preserving the linear minimization, Hirschmüller [2008] proposed to aggregate matching costs in 1D from several directions equally. The number of proposed directions was 8 or 16. For the sake of simplicity we focus on the 8 directions case, i.e. when the direction $\mathbf{r}$ verifies

$$\mathbf{r} \in \{(1,0), (-1,0), (0,1), (0,-1), (1,1), (-1,-1), (1,-1), (-1,1)\}. \tag{5.22}$$

The aggregated cost $S(\mathbf{p}, d)$ for a pixel $\mathbf{p}$ with disparity d is obtained by adding the costs $L_{\mathbf{r}}(\mathbf{p}, d)$ over all the directions $\mathbf{r}$:

$$S(\mathbf{p}, d) = \sum_{\mathbf{r}} L_{\mathbf{r}}(\mathbf{p}, d), \tag{5.23}$$

where $L_{\mathbf{r}}(\mathbf{p}, d)$ is the cost of the 1D minimum $\mathbf{r}$-oriented path ending in pixel $\mathbf{p}$ with disparity d. As for L_- and L_+ in formula (5.11), $L_{\mathbf{r}}$ is computed recursively by

$$\begin{aligned} L_{\mathbf{r}}(\mathbf{p}, d) = C(\mathbf{p}, d) + \min(&L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d), \\ &L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d + 1) + P_1, \\ &L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d - 1) + P_1, \\ &\min_i L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, i) + P_2) \end{aligned} \tag{5.24}$$

As the term $\min_i L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, i)$ is computed only once per each $\mathbf{p}$, this update required only five additions and the computation of the minimum of four numbers. The computation of the $L_{\mathbf{r}}$ cost volume is detailed in algorithm 8. Its complexity is $O(whk)$ because each pixel is covered once. The disparity for each pixel $\mathbf{p}$ is then obtained by minimizing the aggregated cost with respect to d:

$$\mathrm{D}_{\mathbf{p}} = \arg\min_d S(\mathbf{p}, d). \tag{5.25}$$

What does SGM really minimize?

By doing so the original 2D energy (5.1) is approximated, for each pixel $\mathbf{p}$, by another energy $E_{\mathbf{p}}$ defined on a *Union Jack* domain centered on $\mathbf{p}$, as shown on Figure 5.3. In the previous section we computed the aggregated cost L_+ in the horizontal direction $\mathbf{r} = (1, 0)$, and showed that its value in $(\mathbf{p}, d)$ gives the minimum of $E_{\mathbf{p},+}(\mathrm{D})$ with respect to all the disparity maps D such that $\mathrm{D}_{\mathbf{p}} = d$. Given a pixel $\mathbf{p}$ and a direction $\mathbf{r}$, let us define

$$E_{\mathbf{p},\mathbf{r}}(\mathrm{D}) = \sum_{\mathbf{q} \in \Omega \cap (\mathbf{p} - \mathbb{N}\mathbf{r})} C(\mathbf{q}, \mathrm{D}_{\mathbf{q}}) + V(\mathrm{D}_{\mathbf{q}}, \mathrm{D}_{\mathbf{q}-\mathbf{r}}). \tag{5.26}$$

When $\mathbf{r} = (1, 0)$ this definition is the same as that of $E_{\mathbf{p},+}$ given in equation (5.10). It can be shown as in proposition 6 that

$$L_{\mathbf{r}}(\mathbf{p}, d) = \min_{\mathrm{D}, \mathrm{D}_{\mathbf{p}} = d} E_{\mathbf{p},\mathbf{r}}(\mathrm{D}). \tag{5.27}$$

Then, for a given $\mathbf{p}$, as the minimizations (5.27) for various directions $\mathbf{r}$ are carried out on disjoint sets of variables, namely $\{\mathrm{D}_{\mathbf{q}} | \mathbf{q} \in \Omega \cap (\mathbf{p} - \mathbb{N}^*\mathbf{r})\}$, the sum of the minima is the minimum of the sum:

$$\begin{aligned} \sum_{\mathbf{r}} L_{\mathbf{r}}(\mathbf{p}, d) &= \min_{\mathrm{D}, \mathrm{D}_{\mathbf{p}} = d} \sum_{\mathbf{r}} E_{\mathbf{p},\mathbf{r}}(\mathrm{D}) \\ &= \min_{\mathrm{D}, \mathrm{D}_{\mathbf{p}} = d} E_{\mathbf{p}}(\mathrm{D}), \end{aligned} \tag{5.28}$$

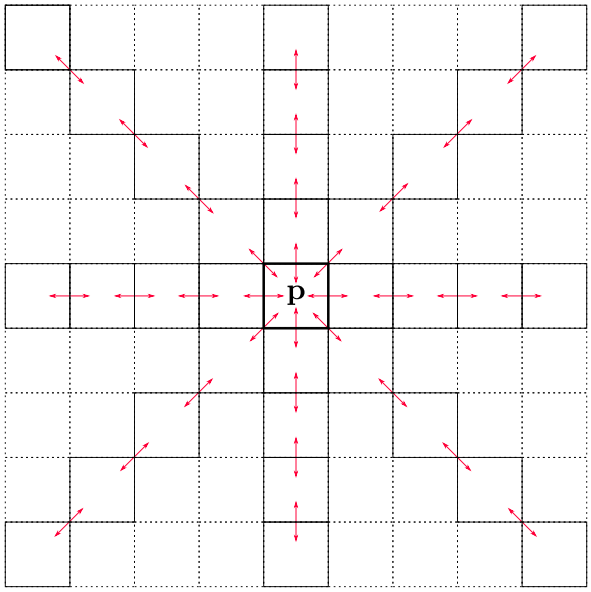


Figure 5.3 – Aggregation of costs in 8 directions around pixel $\mathbf{p}$. The shape of the set of considered pixels reminds the *Union Jack*, as noted by Drory et al. [2014]. The red arrows show the interaction potentials that are taken into account.

where we introduced the notation $E_{\mathbf{p}} = \sum_{\mathbf{r}} E_{\mathbf{p},\mathbf{r}}$. Hence it follows from (5.23) that

$$\arg\min_d S(\mathbf{p}, d) = \arg\min_d \min_{\mathrm{D},\mathrm{D}\mathbf{p}=d} E_{\mathbf{p}}(\mathrm{D}). \tag{5.29}$$

Then if we assume that $E_{\mathbf{p}}$ has a unique global minimum, lemma 4 ensures that this pointwise minimization of $S(\mathbf{p}, \cdot)$ gives the value in $\mathbf{p}$ of a disparity map that minimizes $E_{\mathbf{p}}$. The energy *really* minimized by SGM is thus

$$E_{\mathbf{p}}(\mathrm{D}) = \sum_{\mathbf{r}} \sum_{\mathbf{q}\in\Omega\cap(\mathbf{p}-\mathbb{N}\mathbf{r})} C(\mathbf{q}, \mathrm{D}_{\mathbf{q}}) + V(\mathrm{D}_{\mathbf{q}}, \mathrm{D}_{\mathbf{q}-\mathbf{r}}). \tag{5.30}$$

The domain of that energy, i.e. the Union Jack shown in Figure 5.3, changes for every pixel like in local stereo methods such as block-matching. From that perspective, SGM belongs to the family of *local* stereo method.

How well does this energy approximate the real energy?

While all the cost and interaction terms related to pixels out from the union jack are neglected, the cost of the current pixel $\mathbf{p}$ appears in each of the $E_{\mathbf{p},\mathbf{r}}$ and thus is counted eight times. Hence, as was reported by Drory et al. [2014], a closer approximation is obtained by subtracting $7C(\mathbf{p}, D_{\mathbf{p}})$ from the aggregated cost. The *over counting corrected* (occ) aggregated cost

$$S^{\mathrm{occ}}(\mathbf{p}, d) = \sum_{\mathbf{r}} L_{\mathbf{r}}(\mathbf{p}, d) - 7C(\mathbf{p}, d) \tag{5.31}$$

permits to compute the minimum of the energy

$$E_{\mathbf{p}}^{\text{occ}}(\mathrm{D}) = E_{\mathbf{p}}(\mathrm{D}) - 7C(\mathbf{p}, \mathrm{D}_{\mathbf{p}}) = C(\mathbf{p}, \mathrm{D}_{\mathbf{p}}) + \sum_{\mathbf{r}} \sum_{\mathbf{q} \in \Omega \cap (\mathbf{p} - \mathbb{N}^* \mathbf{r})} C(\mathbf{q}, \mathrm{D}_{\mathbf{q}}) + V(\mathrm{D}_{\mathbf{q}}, \mathrm{D}_{\mathbf{q}-\mathbf{r}}) \quad (5.32)$$

which is a better approximation of the initial energy (5.1).

Algorithm 8: Directional 1D cost aggregation in direction $\mathbf{r}$

Input: cost table C
disparity bounds $d_{\min}, d_{\max}$
direction $\mathbf{r} = (r_x, r_y)$
Output: aggregated cost table $L_{\mathbf{r}}$

```
if r_x ≠ 0 then
    if r_x > 0 then                                   // determines the first scanned column
        x_0 ← 0
    else
        x_0 ← w − 1
    for y ∈ {0, ..., h − 1} do
        for d ∈ {d_min, ..., d_max} do                // initialization of the first scanned column
            L_r((x_0, y), d) ← C((x_0, y), d)
        p ← (x_0, y) + r
        while p ∈ Ω do                                // aggregation from the first column
            for d ∈ {d_min, ..., d_max} do
                Compute L_r(p, d) from L_r(p − r, ·) and C(p, d) according to
                  formula (5.24)
            p ← p + r
if r_y ≠ 0 then
    if r_y > 0 then                                   // determines the first scanned row
        y_0 ← 0
    else
        y_0 ← h − 1
    for x ∈ {0, ..., w − 1} do
        for d ∈ {d_min, ..., d_max} do                // initialization of the first scanned row
            L_r((x, y_0), d) ← C((x, y_0), d)
        p ← (x, y_0) + r
        while p ∈ Ω do                                // aggregation from the first row
            for d ∈ {d_min, ..., d_max} do
                Compute L_r(p, d) from L_r(p − r, ·) and C(p, d) according to
                  formula (5.24)
            p ← p + r
```

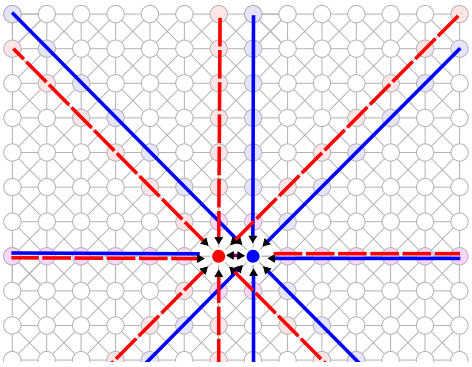


Figure 5.4 – Star-shaped graphs associated in SGM to two adjacent pixels.

5.2.4 Interpretation of SGM as Min-Sum Belief Propagation

In [Drory et al. 2014] SGM is interpreted as a Belief Propagation (BP) [Pearl 1988] algorithm on a star-shaped graph (see Figure 5.3). The Min-sum BP is a *message-passing* algorithm, that computes each node's belief (energy min marginal) by sending messages along the edges of the graph. A message from node $\mathbf{q}$ to node $\mathbf{p}$ is defined recursively from the messages to node $\mathbf{q}$ as:

$$m_{\mathbf{q}\to\mathbf{p}}(d) = \min_{d'\in\mathcal{D}}(C_{\mathbf{q}}(d') + \sum_{(\mathbf{q},\mathbf{k})\in\mathcal{E},\mathbf{k}\neq\mathbf{p}} m_{\mathbf{k}\to\mathbf{q}}(d') \; + V(d,d')). \tag{5.33}$$

And the state belief of a node $\mathbf{p}$ is computed from the messages as

$$B(\mathbf{p},d) = C(\mathbf{p},d) + \sum_{(\mathbf{q},\mathbf{p})\in\mathcal{E}} m_{\mathbf{q}\to\mathbf{p}}(d). \tag{5.34}$$

Upon convergence $\arg\min_d B(\mathbf{p},d)$ yields the estimated solution.

On a tree, BP computes the exact minimum of the energy in two passes sending message from the leaves to the root and back. On a general graph however, BP is implemented as an iterative algorithm, updating messages according to some schedule. The sequential schedule [Tappen and Freeman 2003, Szeliski et al. 2008], for instance, propagates messages in raster order updating the nodes immediately. Messages are typically initialized to all-zero.

Equation (5.33) is reminiscent of SGM. Indeed, Drory et al. [2014] established the connection between SGM and the min-sum BP algorithm. In their interpretation the SGM recursive update formula (5.3) is computing the state belief of the node $\mathbf{p}$ during the $\mathbf{r}$-oriented scan of the image. To make it explicit:

$$B_{\mathbf{r}}(\mathbf{p},d) = L_{\mathbf{r}}(\mathbf{p},d) = C(\mathbf{p},d) + \overbrace{\min_{d'\in\mathcal{D}}(L_{\mathbf{r}}(\mathbf{p}-\mathbf{r},d') + V(d,d'))}^{m_{(\mathbf{p}-\mathbf{r})\to(\mathbf{p})}(d)}. \tag{5.35}$$

They also show that the aggregate of state beliefs for all 8-directions given by (5.4), corresponds (modulo the *over counting* correction (5.31)) to the min-marginals for the star-shaped graph centered at $\mathbf{p}$ (shown in figure 5.5). Then, minimizing $S^{\text{occ}}(\mathbf{p},\cdot)$ gives the exact minimizer of the energy restricted to the star-shaped graph, as was shown in section 5.2.3.

SGM and streaking artifacts. In the light of this result we can understand the occurrence of the streaking artifacts in SGM. Although the optimization is exact over the star-shaped graph, the graphs for two adjacent pixels (as shown in figure 5.4) are loosely related as they share only the nodes on a single line (plus 4 intersection points). In this case, when the data term on the horizontal line is weak, i.e. all disparity hypothesis are equally plausible, the messages from the vertical directions, which are completely unrelated, can and will produce different results for each pixel. This also means that the smoothness constraint is poorly enforced by SGM because the messages are restricted to the 8 paths of the graph.

5.2.5 Effortless linear and truncated linear potentials with min-convolution

Felzenszwalb and Huttenlocher [2006] propose several strategies for accelerating the message computation of BP algorithms. One of the techniques consists in expressing linear, truncated linear, quadratic, and truncated quadratic potentials as a *min convolution* [Felzenszwalb and Huttenlocher 2012]. This allows to update all the labels at a given node in linear time. Given the tight connection between SGM and BP it is straightforward to adapt the min-convolution strategy to SGM's update rules (5.24). This allows to incorporate the linear and truncated linear models in SGM. This change does not affect the performance of SGM.

Following Felzenszwalb and Huttenlocher [2006], let us first consider an interaction potential of the form

$$V(p,q) = \delta|p-q|. \tag{5.36}$$

The recursive update of $L_{\mathbf{r}}(\mathbf{p},\cdot)$ for this potential is written as

$$L_{\mathbf{r}}(\mathbf{p},d) \leftarrow C(\mathbf{p},d) + \min_{d'}(L_{\mathbf{r}}(\mathbf{p}-\mathbf{r},d') + \delta|p-q|). \tag{5.37}$$

A naive implementation of this formula amounts, for k labels, to performing $k \times k$ comparisons. In [Felzenszwalb and Huttenlocher 2006] it is observed that since the labels are ordered, the previous formula could be seen as a min convolution of the form $m(d) \leftarrow \min_p(h(p)+\delta|p-d|)$, which can be computed in linear time $O(k)$ with the simple algorithm 9.

Algorithm 9: Min-convolution algorithm to compute: $m(d) \leftarrow \min_p(h(p) + \delta|p-d|)$

Input: vector h of length k and the parameter δ
Output: vector m

$m \leftarrow h$ *// initialize m copying h*
for $d \in \{1,\dots,k-1\}$ **do**
  $m(d) \leftarrow \min(m(d), m(d-1)+\delta)$ *// forward pass*
for $d \in \{k-2,\dots,0\}$ **do**
  $m(d) \leftarrow \min(m(d), m(d+1)+\delta)$ *// backward pass*

To compute a truncated distance of the form $V(p,q) = \min(\delta|p-q|, \tau)$ a last step is added:
for $d \in \{0,\dots,k-1\}$ **do**
  $m(d) \leftarrow \min(m(d), \min_p(m(p)) + \tau)$ *// point-wise truncation*
Note that $\min_p(m(p))$ could be easily precomputed.

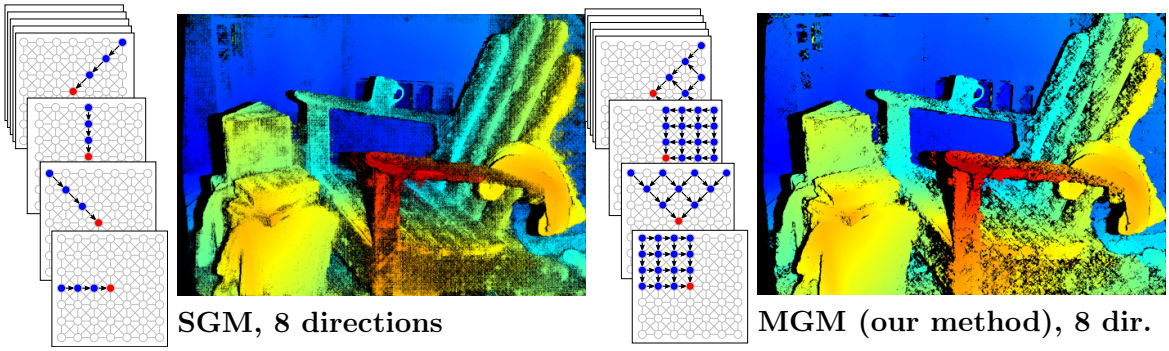


Figure 5.5 – Results of the baseline implementation of SGM and MGM for the *Adirondack* pair (figure 5.11). The diagrams on the left depict for each method the information used by the recursive update of the red pixel in each of the 8 scans of the algorithms.

5.3 More Global Matching

In the spirit of the belief update formula (5.34) we propose to update $L_{\mathbf{r}}$ using information from more than one direction. Concretely our strategy injects information from the 2D problem in the processing of SGM's 1D paths (see figure 5.5). This is efficiently done by incorporating messages from the nodes visited in the previous scanline (i.e. the pixel above).

Let us consider the left-to-right direction. The image is traversed in raster order (left-to-right, top-to-bottom) and SGM updates each node $\mathbf{p}$ using only the beliefs from the node on its left $L_{\mathbf{r}}(\mathbf{p}-\mathbf{r},\cdot)$. Instead we propose to access as well the beliefs from the node directly above $\mathbf{p}$ (indicated by the direction $\mathbf{r}^{\perp}$) . Thus our proposed recursion is:

$$L_{\mathbf{r}}(\mathbf{p},d) = C(\mathbf{p},d) + \sum_{\mathbf{x}\in\{\mathbf{r},\mathbf{r}^{\perp}\}} \frac{1}{2} \min_{d'\in\mathcal{D}}(L_{\mathbf{r}}(\mathbf{p}-\mathbf{x},d') + V(d,d')). \tag{5.38}$$

As a result of this multiple recursion, the belief at a given pixel is influenced by its entire upper-left quadrant (as illustrated in figure 5.5). In comparison SGM recursion only sees information from the line of pixels to its left.

For each propagation direction $\mathbf{r}$ we compute $L_{\mathbf{r}}$ using an adequate traversal order (depicted in figure 5.6). The resulting beliefs are then combined using the over-counting corrected formula (5.31), and the disparity is estimated by WTA. Compared to SGM, MGM only requires a few extra operations per pixel.

Each pass of MGM can be seen as the first iteration of a sequential belief propagation algorithm [Tappen and Freeman 2003, Szeliski et al. 2008] with a particular scheduling order dictated by the propagation direction (see figure 5.6). Since the messages from the non visited nodes (below $\mathbf{p}$) are initialized to 0, the update rule (5.38) corresponds exactly to (5.33).

Algorithm. The pseudo code of the MGM algorithm is given in Algorithm 10.

Parallelization considerations. Since MGM introduces dependency among the scan lines, these cannot be processed in parallel as it is usually done with SGM. A naive parallelization of MGM consists in performing the 4 or 8 image traversals in parallel. However, the performance improvement with this approach is limited to the number of traversals.

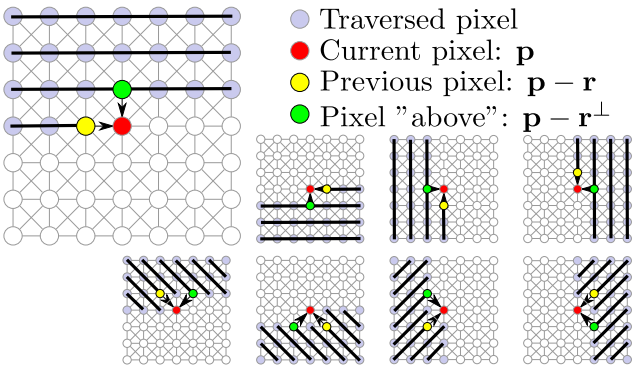


Figure 5.6 – Depiction of the 8 image traversals and the corresponding recursion directions $\mathbf{r}$ and $\mathbf{r}^{\perp}$ in MGM.

Algorithm 10: More Global Matching (MGM)

Input: pixel-wise cost volume C, search space $\mathcal{D} = \{d_{min}, \dots, d_{max}\}$
Output: aggregated and over-counting corrected cost volume S^{occ}
begin

$(L_i)_{i=0,\dots,7} \longleftarrow 0$ *// initialize L_i, each one has the size of the image $\times|\mathcal{D}|$*
for $i \in \{0, \dots, 7\}$ **do** *// 8 traversals as in figure 5.6*
  $(\mathbf{r}, \mathbf{r}^{\perp}) \longleftarrow$ `compute_previous_and_above_directions_for_traversal`(i)
  $(x_n)_{n=0,\dots,N-1} \longleftarrow$ `compute_adequate_pixel_ordering_for_traversal`(i)
  for $n \in \{0, \dots, N-1\}$ **do** *// for all the N pixels of the image*
    $\mathbf{p} \longleftarrow x_n$
    $m_{\mathbf{r}} \longleftarrow \min_{d'} L_i(\mathbf{p}-\mathbf{r}, d')$ *// precompute the two* min
    $m_{\mathbf{r}^{\perp}} \longleftarrow \min_{d'} L_i(\mathbf{p}-\mathbf{r}^{\perp}, d')$
    for $d \in \mathcal{D}$ **do**

$$L_i(\mathbf{p}, d) \leftarrow C(\mathbf{p}, d) + \tfrac{1}{2}\min(L_i(\mathbf{p}-\mathbf{r}, d), L_i(\mathbf{p}-\mathbf{r}, d\pm 1)+P_1, m_{\mathbf{r}}+P_2) + \tfrac{1}{2}\min(L_i(\mathbf{p}-\mathbf{r}^{\perp}, d), L_i(\mathbf{p}-\mathbf{r}^{\perp}, d\pm 1)+P_1, m_{\mathbf{r}^{\perp}}+P_2)$$

$S^{\text{occ}} \longleftarrow (\sum_{i=0}^{7} L_i) - 7\,C$

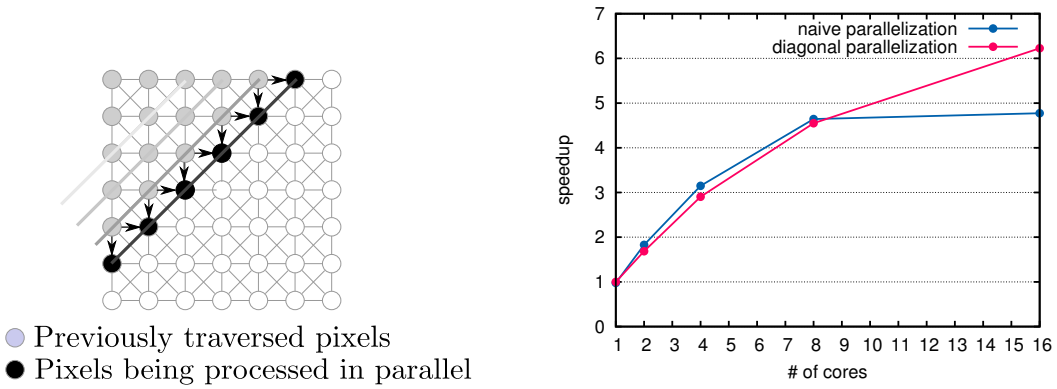


Figure 5.7 – Speedup of MGM applying different parallelization strategies. The naive strategy consists in computing the 8 image traversals in parallel. This does not scale beyond 8 cores. The *diagonal* parallelization (illustrated on the left for the raster ordered image traversal) scales better with the number of cores.

Instead, parallelization in MGM is achieved diagonal-by-diagonal. That is, during the image traversal in raster order (first diagram of figure 5.6) the pixels of a diagonal going from top-right to bottom-left can be processed in parallel, because they only depend on their top and left neighbors (which are computed by the previous diagonal). This strategy is illustrated in figure 5.7. A nearly linear speed-up is observed (figure 5.7) when applying this strategy on a 16-core test machine (Xeon@2.60GHz).

5.4 Experiments

We first evaluate the effectiveness of MGM as a fast approximate minimization tool for pairwise Markov Random Fields. Then the improvement of MGM over SGM is quantified in terms of performance on a stereo benchmark.

5.4.1 Evaluation for MRF minimization

We evaluated our method on pairwise MRFs from the Middlebury energy minimization benchmark [Szeliski et al. 2008]. For the stereo matching problem the MRF graphs consist of regular 4-connected grids. We compared with well known optimization algorithms provided with the benchmark.

Experimental setup. The disparity map from images u and v was estimated by minimizing

$$E(\mathrm{D}) = \sum_{\mathbf{p}\in\mathcal{V}} C(\mathbf{p}, \mathrm{D}_{\mathbf{p}}) + \lambda \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{E}} V(\mathrm{D}_{\mathbf{p}}, \mathrm{D}_{\mathbf{q}}), \tag{5.39}$$

where $\mathcal{E}$ is the edge set of the 4-connected image graph and the disparities $\mathrm{D}_{\mathbf{p}}$ are discretized at one pixel precision. We used the absolute differences of intensities for the data term $C(\mathbf{p}, d) = |u(\mathbf{p}) - v(\mathbf{p} + (d, 0))|$ (summed over all channels for color images). This matching cost, although not robust to radiometric changes, provides a common ground for comparison

	Tsukuba(16 labels)			Teddy(60 labels)			Venus(20 labels)			Fountain(143 labels)		
Method	**Egap%**	**$\mathcal{B}$(%)**	**t(s)**	**Egap%**	**$\mathcal{B}$(%)**	**t(s)**	**Egap%**	**$\mathcal{B}$(%)**	**t(s)**	**Egap%**	**$\mathcal{B}$(%)**	**t(s)**
TRW-S	**ref.**	**4.5**	17	**ref.**	20.5	483	**ref.**	**4.7**	51	ref.	**16.2**	2170
α-expansion	0.09	4.6	5.9	0.13	21.2	58	0.07	**4.7**	12	**-0.05**	16.3	196
BP-S	1.78	8.2	4.7	0.68	**20.1**	382	0.68	6.3	18	1.32	16.9	1697
SGM4	48.3	6.6	1.0	21.7	24.2	3.7	31.4	7.4	1.8	29.0	14.5	14
ocSGM4	41.9	**6.3**	1.0	18.2	21.9	3.8	25.6	6.9	1.8	26.5	**13.6**	14
MGM4	**7.5**	6.7	1.2	**5.5**	**21.4**	4.5	**4.2**	**5.8**	2.0	**10.7**	15.8	17

Table 5.1 – Results. **Egap%** is the energy gap ($\frac{E-E_{ref}}{E_{ref}} \times 100$) of the solution with respect to the reference solution of TRW-S (Tree-Reweighted Message Passing [Kolmogorov 2006]), **t(s)** is the time in seconds to compute the solution, and **$\mathcal{B}$(%)** is the percentage of pixels that differ more than one disparity level from the ground truth. Two other reference methods are also included: alpha-expansion algorithm [Boykov et al. 2001], and BP-S a sequential Belief-Propagation algorithm from [Szeliski et al. 2008]. The running times were measured using a single core of an Intel Core 2 Duo CPU @1.8GHz.

with the methods listed by Szeliski et al. [2008]. The smoothness term $V(\cdot,\cdot)$ is the L^1-norm truncated at 2, which amounts to taking $P_1 = \lambda$ and $P_2 = 2\lambda$ in (5.2). For simplicity the fine tuning of the matching costs and variable weights were not considered here (essentially disabling the intensity cues and the Birchfield and Tomasi [1998] costs for all the methods). However different matching costs and intensity cues can be easily incorporated in MGM as in SGM [Hirschmüller 2008, Bleyer and Gelautz 2008]. For instance, in section 5.4.2 we use census [Zabih and Woodfill 1994] as matching cost, which is robust to radiometric changes. In this section the parameter λ is chosen for each stereo pair but is unchanged across algorithms.

Four test image pairs were considered. Three from the Middlebury benchmark [Szeliski et al. 2008], *Tsukuba* (16 disparity labels, $\lambda = 20$), *Venus* (20 disparity labels, $\lambda = 20$), *Teddy* (60 disparity labels, $\lambda = 10$), and one stereo pair from the multi-view dataset published by Strecha et al. [2008], *Fountain* (143 depth labels, $\lambda = 8$).

Evaluation. We compared MGM with five minimization algorithms. First, three reference MRF optimization techniques, TRW-S (*Tree-Reweighted Message Passing*) [Kolmogorov 2006], the *alpha-expansion* graph-cut algorithm [Boykov et al. 2001], and BP-S, a sequential Belief-Propagation algorithm from [Szeliski et al. 2008]. We also tested our own baseline implementation of SGM [Hirschmüller 2008], one variant incorporating the over-counting correction [Drory et al. 2014], and the proposed algorithm MGM. Since the energy (5.39) is defined on a 4-connected graph, these variants explore only four cardinal directions. Hence they are denoted by SGM4, ocSMG4 and MGM4 respectively.

We assessed the performance by computing the *energy gap*, the *bad pixel ratio* and the *computing time*. The energy gap is the relative gap between the energy of the current solution and the energy of a reference (top performing) strategy, while the bad pixel ratio is the percentage of pixels that differ more than one level from the ground-truth.

Results and discussion. The results are reported in table 5.1 and figure 5.8. We note in table 5.1 that, in terms of energy, SGM4 and the over-counting corrected ocSGM4 [Drory

et al. 2014] produce rather distant solutions from the reference optimum. Looking at figure 5.8 we can see that this error comes mainly from the regularity term (see section 5.4.3), which confirms that in SGM the smoothness term of (5.39) is only weakly enforced. Our algorithm (MGM4), on the other hand, consistently produces better minima than SGM4 and ocSGM4. It yields approximations that are within 10% of the optimum produced by TRW-S (the reference method), an improvement by a factor five with respect to SGM4 or ocSGM4. Another experiment presented in section 5.4.6 was performed on a large set of images (all the stereo pairs from the Middlebury page: 2005, 2006, and 2014). It confirms that on average MGM yields a systematic improvement of 40% in the energy minima with respect to SGM.

In terms of efficiency MGM can be 20% slower than SGM, however the results are well worth the time: they are more regular, show less artifacts, and better approach the minima of the underlying energy. For problems with larger label sets, such as *Fountain*, MGM (with its current non-optimized implementation) already provides an approximate solution in a tenth of the time needed by a classic technique. We can safely say that MGM can be used as an accurate and rather inexpensive approximation for solving problems with truncated smoothness terms or with the Potts interaction penalty [Sinha et al. 2014].

In the horizontal plane of the *Fountain* image (figure 5.8) we see that SGM4 produced smoother results than TRW-S and MGM4. Piecewise constant solutions are expected in this experiment because the smoothness term (5.2) is a truncated L^1-norm. It is more convenient to jump three or more disparity levels once in a while (with penalty 2λ) rather than changing disparity at each row (each with penalty λ). This serves as evidence that SGM4 does not actually enforce the smoothness defined in (5.39). Indeed, SGM4 only enforces the regularity by lines with a weak bond between parallel lines. On the other hand, this issue is easily solved with MGM, since it minimizes a closer approximation of the energy (5.39). For instance, lowering the value of the parameter P_1 (for instance $P_1 = \lambda/4$, while keeping $P_2 = 2\lambda$) prevents this effect. The results are shown in the last column of figure 5.8.

5.4.2 Evaluation on Stereo Pairs with Ground Truth

Experimental setup. We evaluated MGM using 37 full resolution stereo pairs from the Middlebury datasets (2005, 2006, and 2014) [Hirschmüller and Scharstein 2009, Scharstein et al. 2014]. The images of the 2005 and 2006 datasets have a resolution of 1.4MP and the disparity range is about 150 pixels. The 2014 dataset contains 5MP images and the disparity range goes up to 500 pixels.

Evaluation. We were interested in comparing the performance of the proposed algorithm (MGM) with a baseline implementation of SGM and ocSGM (the over-counting corrected SGM). We evaluated the bad pixel ratio of each method. A *bad pixel* is a point whose disparity differs from the ground truth by more than one pixel. The three algorithms use the same settings: 8 propagation directions and, as matching cost $C_{\mathbf{p}}$, the Hamming distance of census transform [Zabih and Woodfill 1994, Zhu et al. 2012] on a 5×5 neighborhood (normalized by the number of channels). The parameters P_1 and P_2 were set for all images to $P_1 = 8$ and $P_2 = 32$ (as in OpenCV's SGBM). No intensity cues (adapting P_2) were used, though they could easily be incorporated [Hirschmüller 2008, Bleyer and Gelautz 2008]. To prevent influence of the post-processing steps on the evaluation, we performed this evaluation on the

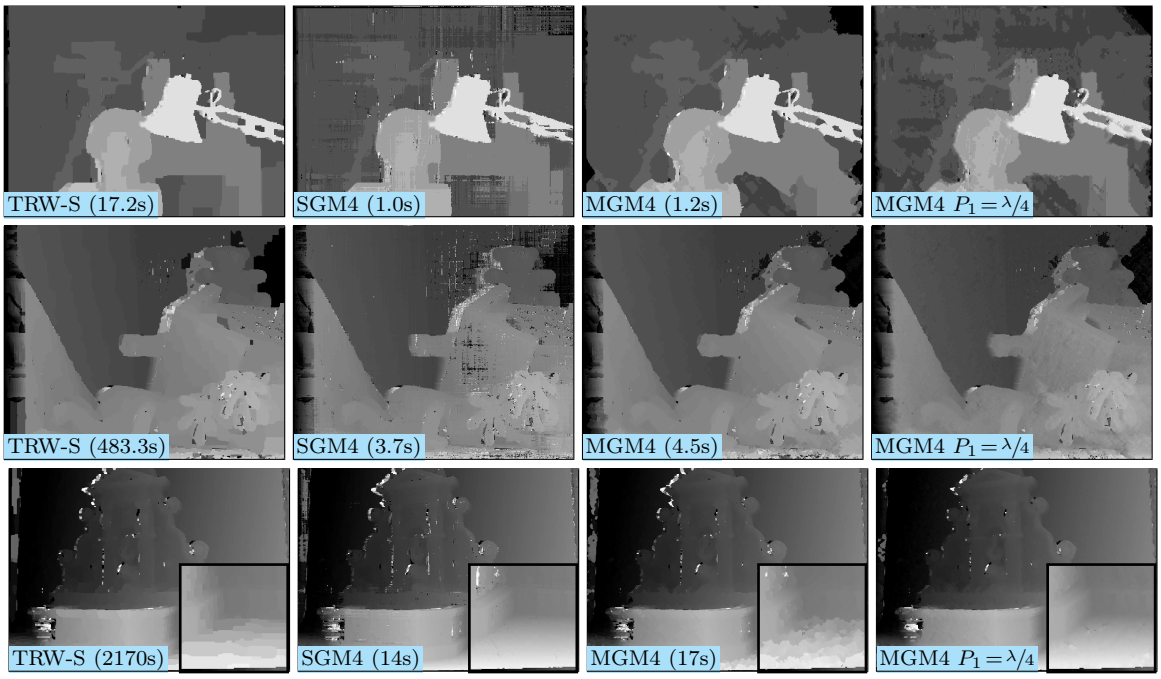


Figure 5.8 – Qualitative result comparison for the images *Tsukuba* and *Teddy* from [Szeliski et al. 2008], plus a low resolution *Fountain* from [Strecha et al. 2008]. Note that in the result of MGM4 there are no streaking artifacts as in the baseline SGM4. The last column (MGM4 $P_1 = \lambda/4$) and the detail magnified in the last figure are commented in the text.

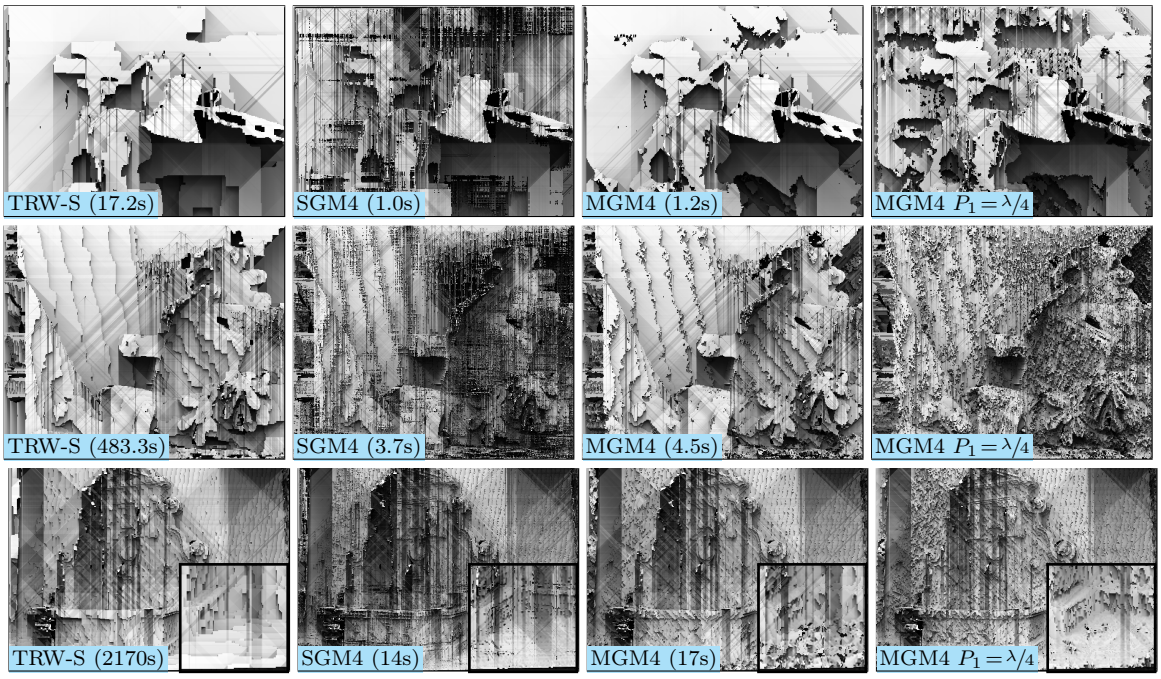


Figure 5.9 – Shaded versions of the disparity maps presented in figure 5.8, obtained with the tool `GrShade` from MicMac [Pierrot Deseilligny 2015].

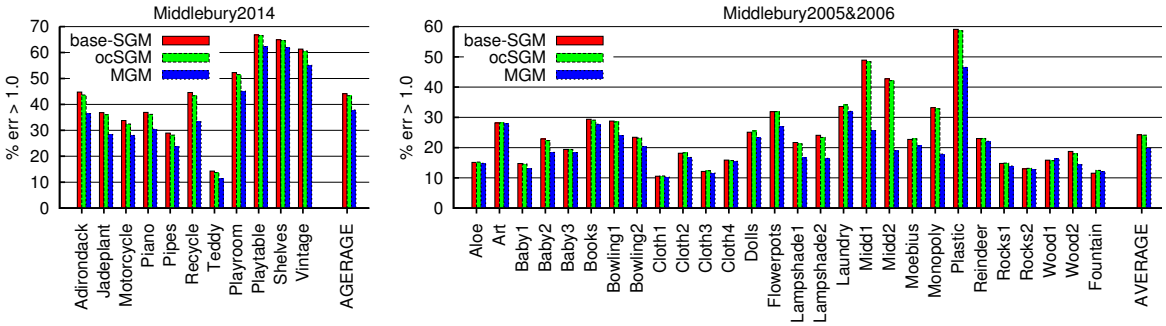


Figure 5.10 – Bad pixel ratio (% of pixels with error larger than 1 disparity level) on the two sets of test images. Our method yields the lowest average errors on the two datasets.

unmodified outputs of the winner-take-all stage of all methods. Section 5.4.5 presents results after post-processing the outputs of SGM and MGM with a 3×3 median filter [Hirschmüller and Scharstein 2009]. The conclusion is that, even after filtering, the results of SGM present more errors than those of MGM, specially in poorly textured areas.

Discussion. The accuracy of the three methods is shown in figure 5.10. It can be seen that MGM is the most accurate on both datasets. This improvement is attained with negligible computation overhead. Our current implementation of SGM runs in 130s on an 8-core Xeon@2.60GHz computer for a 2964×1988 image with 226 disparity levels. On the same image the equivalent implementation of MGM runs in 137s.

A qualitative analysis of the disparity maps (figure 5.11) shows that MGM produces results that are denser and cleaner, with less streaking artifacts. To facilitate the qualitative evaluation these results were filtered with the left-to-right consistency check [Fua 1993] with threshold set to 1. Moreover, for all these experiments MGM yielded (with respect to SGM) a systematic reduction of about 40% in the energy minima for the 8-connected energy, as shown in section 5.4.4.

5.4.3 In SGM the regularity term is only weakly enforced

As mentioned in section 5.4.1, we observe that in the SGM algorithm the regularity defined in the energy

$$E(\mathrm{D}) = \underbrace{\sum_{\mathbf{p} \in \mathcal{V}} C(\mathbf{p}, \mathrm{D}_{\mathbf{p}})}_{\mathbf{C}} + \lambda \underbrace{\sum_{(\mathbf{p},\mathbf{q}) \in \mathcal{E}} V(\mathrm{D}_{\mathbf{p}}, \mathrm{D}_{\mathbf{q}})}_{\mathbf{V}} \tag{5.40}$$

is only weakly enforced, whereas MGM yields a result closer to the reference optimization techniques. To corroborate this, we evaluated the energy of the solutions reported in section 5.4.1. In table 5.2 we distinguish the contributions of the data term ($\mathbf{C}$) and the regularity term ($\mathbf{V}$) for each solution, and compare the ratios $\mathbf{C}/\mathbf{V}$. First, we note that for all the methods the data term $\mathbf{C}$ has approximately the same energy. However the regularity terms for the results of SGM4 and ocSGM4 are abnormally high, which means that the regularity

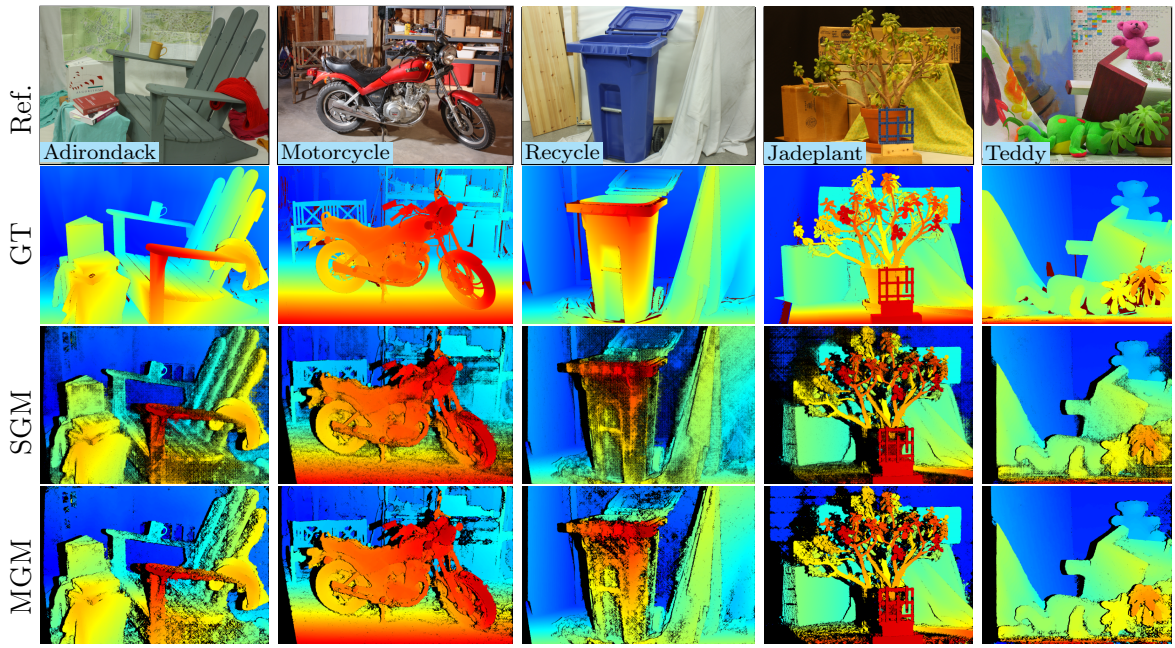


Figure 5.11 – Results on some of the test images with ground truth.

term does not enforce regularity. MGM4, on the other hand, yields much lower values for the regularity term which means that the obtained solutions are more regular.

5.4.4 MGM systematically improves the energy minima

To complement the experiment of section 5.4.1 we evaluated the energy gap between the solutions obtained with MGM and the 8-directions SGM (the larger the gap the better). We used for this experiment full resolution stereo-pairs from the datasets published with [Hirschmüller and Scharstein 2009] and [Scharstein et al. 2014]. The energy is measured on the 8-connected graph, and the smoothness term is

$$V(d,d') = \begin{cases} 0 & \text{if } d = d' \\ P_1 & \text{if } |d-d'| = 1 \\ P_2 & \text{otherwise} \end{cases} , \tag{5.41}$$

with $P_1 = 8$ and $P_2 = 32$ for all the experiments.

In table 5.3 we see that the energy of the solutions obtained with MGM is on average 40% lower than with SGM, while with ocSGM it is only 10% lower on average.

5.4.5 Post-processing with median filter

As the evaluation presented in section 5.4.2 focuses on energy minimization, no post-processing of the outputs is considered. However, a key component of the SGM algorithm [Hirschmüller 2008] is the 3×3 median filtering of the disparity maps to fill-in small holes. Here we compare the impact of this post-process (denoted $+med$) on the baseline SGM and MGM.

	Tsukuba(16 labels)				Teddy(60 labels)				Venus(20 labels)				Fountain(143 labels)			
Method	**Egap**	**C**	**V**	**C/V**	**Egap**	**C**	**V**	**C/V**	**Egap**	**C**	**V**	**C/V**	**Egap**	**C**	**V**	**C/V**
TRW-S	0.00	933	192	4.8	0.00	3293	432	7.6	0.00	2164	205	10.5	0.00	2899	703	4.1
Expansion	0.09	928	199	4.7	0.13	3305	425	7.8	0.07	2171	200	10.8	-0.05	2911	688	4.2
BP-S	1.78	939	206	4.5	0.68	3295	456	7.2	0.68	2163	223	9.7	1.32	2932	717	4.1
SGM4	48.3	844	826	1.0	21.7	2968	1565	1.9	31.4	2003	1110	1.8	29.0	2611	2036	1.3
ocSGM4	41.9	904	693	1.3	18.2	3225	1179	2.7	25.6	2130	846	2.5	26.5	2794	1763	1.6
MGM4	7.51	918	292	3.1	5.53	3229	703	4.6	4.19	2144	325	6.6	10.7	2779	1207	2.3

Table 5.2 – Energy ratios. **E** is the energy gap ($\frac{E_{ref}-E}{E_{ref}} \times 100$) of the solution with respect to the reference solution of TRW-S (Tree-Reweighted Message Passing [Kolmogorov 2006]), **C/V** is the ratio in the final energy between the data term (**C**) and the regularity term (**V**). All the reported energies are measured in thousands. Two other reference methods are also included: Expansion-move algorithm [Boykov et al. 2001], and BP-S a sequential Belief-Propagation algorithm from [Szeliski et al. 2008]. The energy ratios **C/V** indicate the current balance of the data and regularity term in the solution. In both SGM4 and ocSGM4, the regularity term (**V**) is always much larger than in the solutions of the other methods. This means that the regularity term is weakly enforced by SGM4 and ocSGM4. On the other hand, for MGM4 the energy ratios are much closer to those of reference optimization algorithms.

Figure 5.12 shows that the results of SGM improve substantially after median filtering, while the results of MGM do not change much. Nevertheless, even after filtering the results of SGM present more errors than MGM specially in poorly textured areas (as shown in figure 5.13).

5.4.6 Middlebury 2014 dataset

Figures 5.15 and 5.16 show more results on the full Middlebury benchmark (version 3) [Scharstein et al. 2014]. We compare the results of our baseline implementation of SGM, MGM, and the publicly available ELAS [Geiger et al. 2011]. The two figures show the sparse and dense results of each method. SGM and MGM use the same settings for all the images: 8 propagation directions and the Hamming distance of census transform [Zabih and Woodfill 1994, Zhu et al. 2012] on a 5×5 neighborhood (normalized by the number of channels) as matching cost $C(\mathbf{p}, \cdot)$. The parameters P_1 and P_2 were set for all images to $P_1 = 8$ and $P_2 = 32$. No intensity cues (adaptation of P_2) were used but they could easily be incorporated [Hirschmüller 2008, Bleyer and Gelautz 2008] in both SGM and MGM. To prevent influence of the post-processing steps on the evaluation, the results shown for base-SGM and MGM in figure 5.15 correspond to the unaltered outputs of the winner-take-all stage. The bad pixel ratios for the dense results are compared in table 5.10. The sparse results (figure 5.16) are obtained by applying the left-to-right consistency check [Fua 1993] with threshold set to 1.

5.4.7 KITTI dataset

Results on the KITTI training and testing datasets [Geiger et al. 2012] are shown in tables 5.4 and 5.5. We didn't optimize the parameters, instead we used the ones proposed in [Spangenberg et al. 2014]: $P_1 = 7$, $P_2 = 100$. We used the census distance (5×5 windows) as cost, refined the disparities with V-fit sub-pixel interpolation [Haller and Nedevschi 2012],

	Image	$1 - \left(\frac{E_{ocSGM}}{E_{SGM}}\right)$	$1 - \left(\frac{E_{MGM}}{E_{SGM}}\right)$
Middlebury 2014	Motorcycle	0.090	0.419
	Jadeplant	0.059	0.401
	Adirondack	0.050	0.358
	Playtable	0.093	0.482
	Playroom	0.057	0.289
	Teddy	0.142	0.461
	Vintage	0.075	0.523
	Pipes	0.096	0.381
	Shelves	0.058	0.549
	Recycle	0.060	0.515
	Piano	0.092	0.504
Middlebury 2006	Baby1	0.126	0.446
	Baby3	0.131	0.449
	Baby2	0.115	0.451
	Plastic	0.046	0.482
	Aloe	0.153	0.311
	Cloth2	0.124	0.298
	Cloth3	0.174	0.271
	Cloth1	0.175	0.269
	Cloth4	0.154	0.311
	Lampshade1	0.084	0.514
	Wood1	0.133	0.511
	Wood2	0.076	0.541
	Lampshade2	0.062	0.520
	Flowerpots	0.071	0.423
	Monopoly	0.091	0.312
	Bowling1	0.073	0.501
	Bowling2	0.090	0.475
	Midd2	0.078	0.518
	Midd1	0.099	0.502
	Rocks1	0.178	0.376
	Rocks2	0.185	0.369
	Books	0.133	0.382
Midd. 2005	Moebius	0.150	0.378
	Reindeer	0.142	0.410
	Dolls	0.124	0.322
	Laundry	0.129	0.371
	Art	0.115	0.374
	Average	0.107	**0.420**

Table 5.3 – Energy gap (**the larger the better**) between the solutions obtained with MGM and the baseline SGM with 8 directions. The images are from the stereo datasets [Hirschmüller and Scharstein 2009, Scharstein et al. 2014]. The energy is measured on the 8-connected graph, in all the test we use $P_1 = 8$ and $P_2 = 32$ for the smoothness term (5.2). The energy obtained with MGM is on average 40% lower than with SGM, while with ocSGM it is only 10% lower.

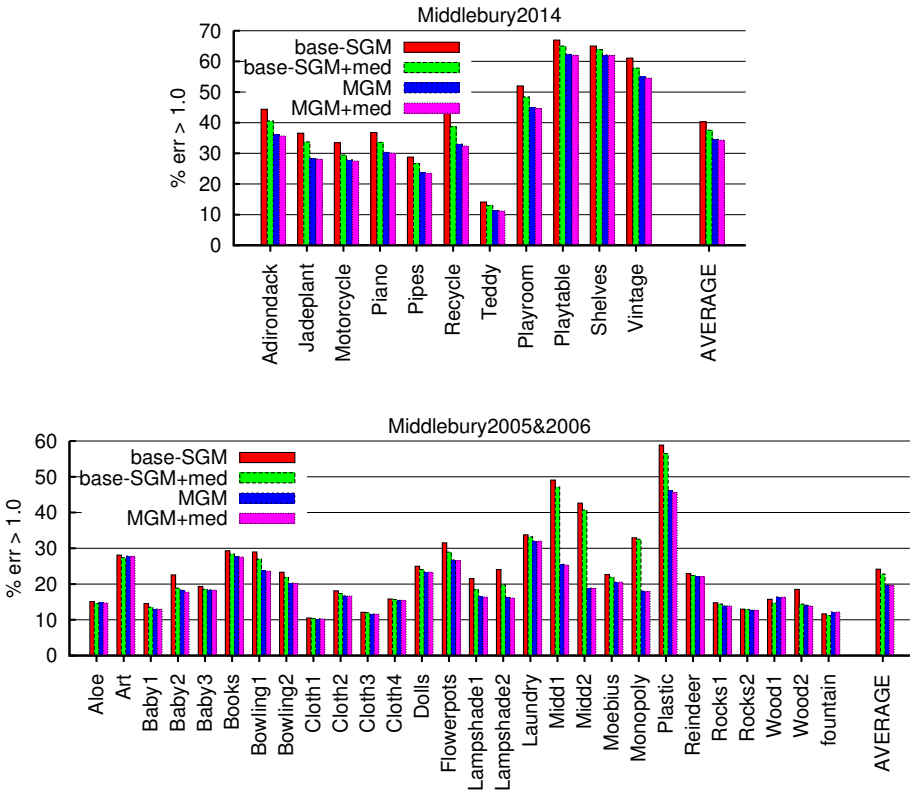
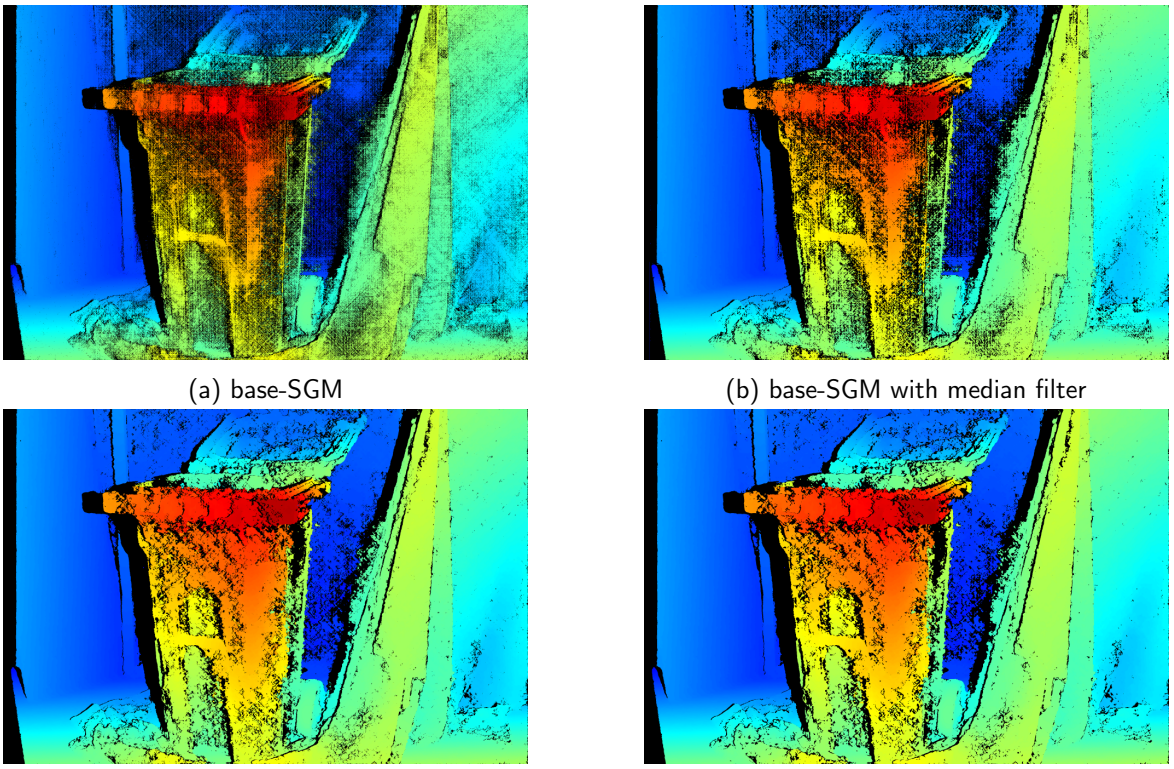


Figure 5.12 – Bad pixel ratios (% of pixels with error larger than 1 disparity level) on the Middlebury 2005, 2006 and 2014 datasets. The statistics corresponds to results of the baseline SGM and MGM, with and without the post-processing using the 3×3 median filter (+med).



(a) base-SGM (b) base-SGM with median filter

(c) MGM (d) MGM with median filter

Figure 5.13 – Comparison of the impact of the median filter post-process when applied to the baseline SGM and MGM.

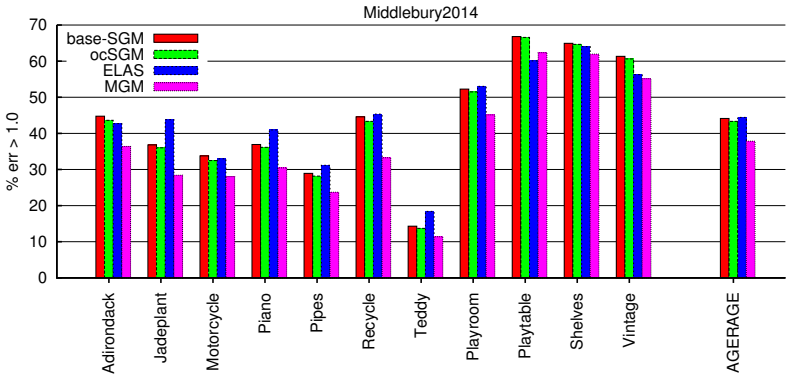


Figure 5.14 – Bad pixel ratios (% of pixels with error larger than 1) on the Middlebury 2014 test images.

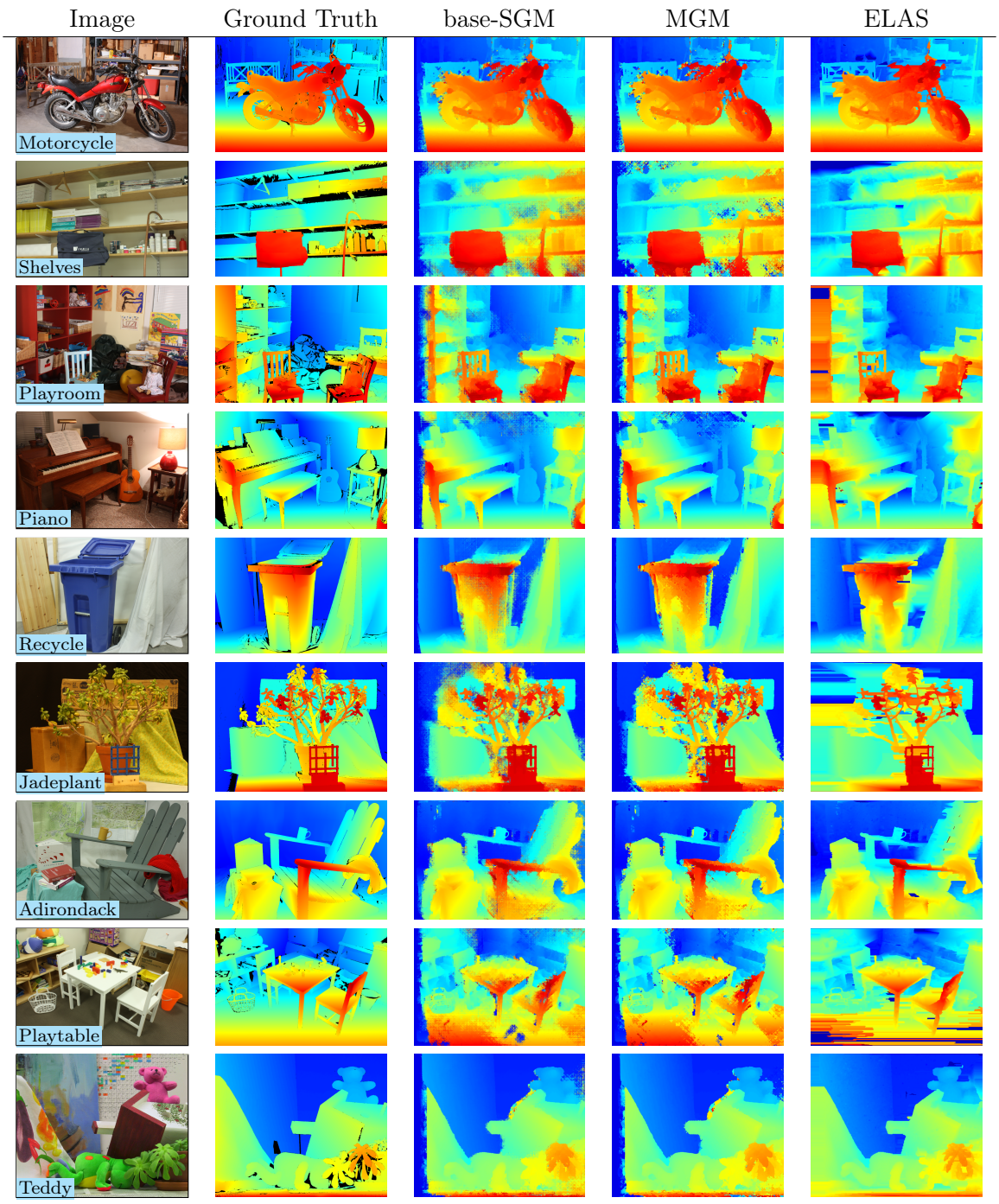


Figure 5.15 – Dense results on stereo pairs with ground truth from the 2014 Middlebury dataset. Continues on the next page.

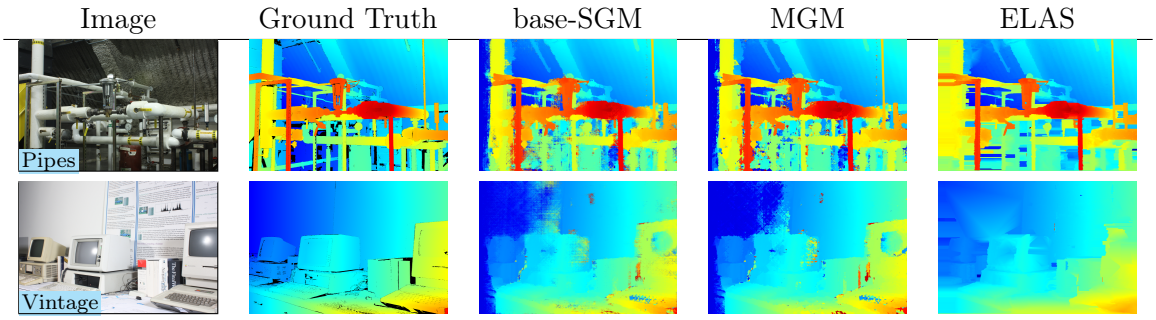


Figure 5.15 – Dense results on stereo pairs with ground truth from the 2014 Middlebury dataset. We compare the results of our baseline implementation of SGM, MGM, and ELAS [Geiger et al. 2011].

method	**Out-Noc (all pixels)**	**Out-Noc (estimated pixels)**	**Density**
base-SGM	7.30 %	**3.94 %**	88.78 %
MGM	**6.33** %	4.09 %	**92.18**%

Table 5.4 – Average results of the MGM and base-SGM algorithms on the KITTI **training** dataset. *Out-Noc* is the fraction of erroneous pixels (computed using the threshold at 3 pixels) in non-occluded areas, evaluated on all the pixels (interpolated), or restricted to the estimated ones. The average densities of the disparity maps are shown in the last column.

then filtered with a 3 median filter and with the left-to-right consistency check. The running time for computing each frame on a 16-core computer is about 6 seconds for MGM, and 5.3 seconds for base-SGM. The results shown in figure 5.17 confirm that MGM yields slightly denser results than base-SGM.

5.5 Conclusion

In this chapter we studied SGM, which is still one of the most popular algorithms for stereo matching. Based on this study we proposed a new algorithm called MGM. Our method is a variant of SGM where messages are propagated on a quadrant of the whole graph instead of a line subgraph. This elaborates on a recent interpretation of SGM in terms of belief

Error	**Out-Noc**	**Out-All**	**Avg-Noc**	**Avg-All**
2 pixels	8.51 %	9.95 %	1.2 px	1.4 px
3 pixels	5.70 %	6.89 %	1.2 px	1.4 px
4 pixels	4.31 %	5.30 %	1.2 px	1.4 px
5 pixels	3.45 %	4.28 %	1.2 px	1.4 px

Table 5.5 – Results of the MGM algorithm on the KITTI **testing** dataset. We used the census distance (5×5 windows), with parameters $P_1 = 7$, $P_2 = 100$ (as in [Spangenberg et al. 2014]) the disparities are refined with V-fit sub-pixel interpolation [Haller and Nedevschi 2012], then filtered with a 3 median filter, and with the left-to-right consistency check. The running time for each image is about 6 seconds on a 16-core computer.

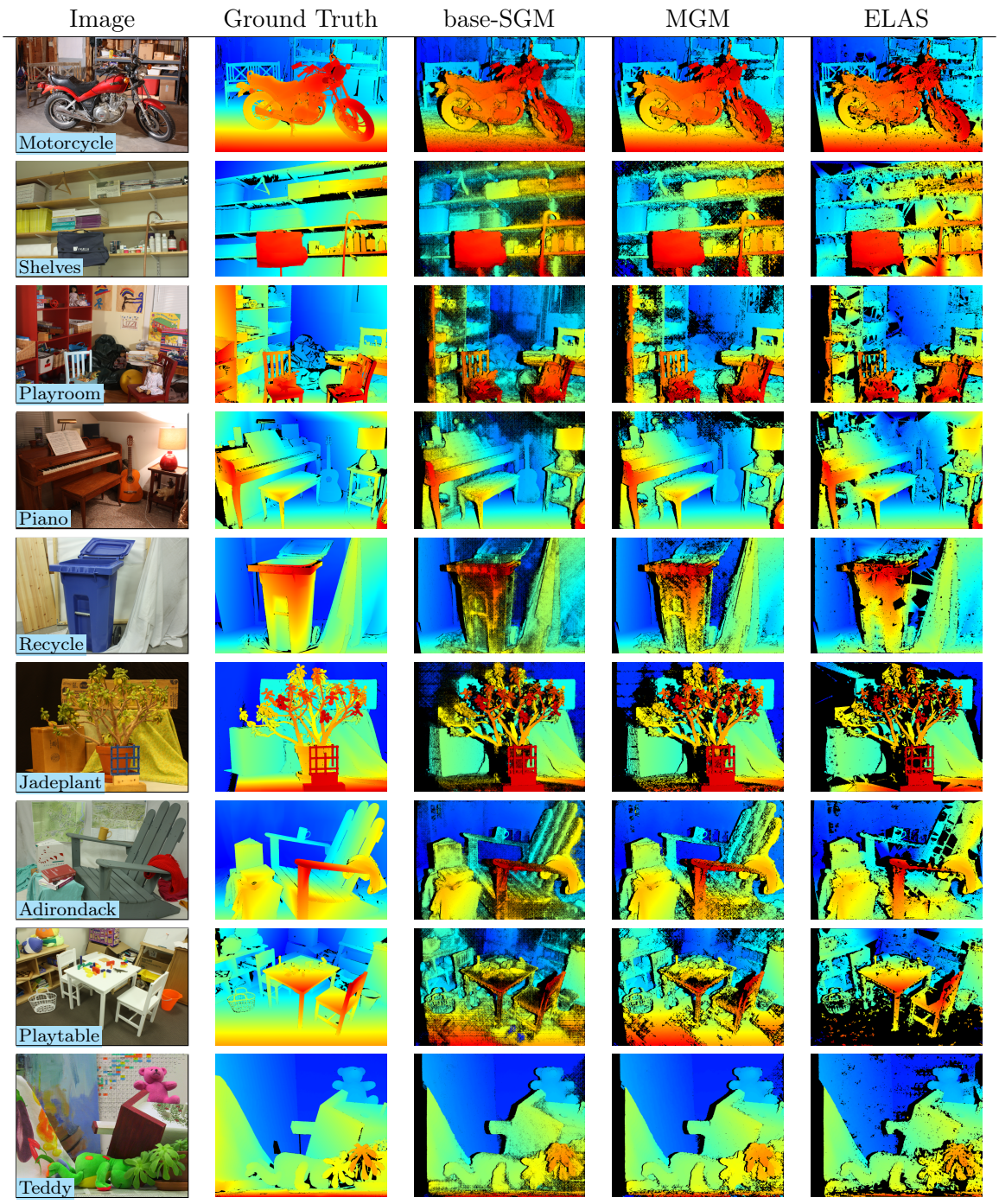


Figure 5.16 – Sparse results on stereo pairs with ground truth from the 2014 Middlebury dataset. Continues on the next page.

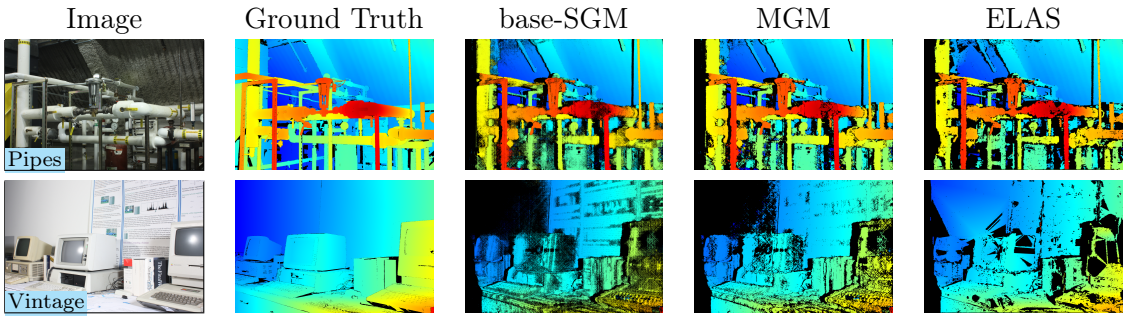
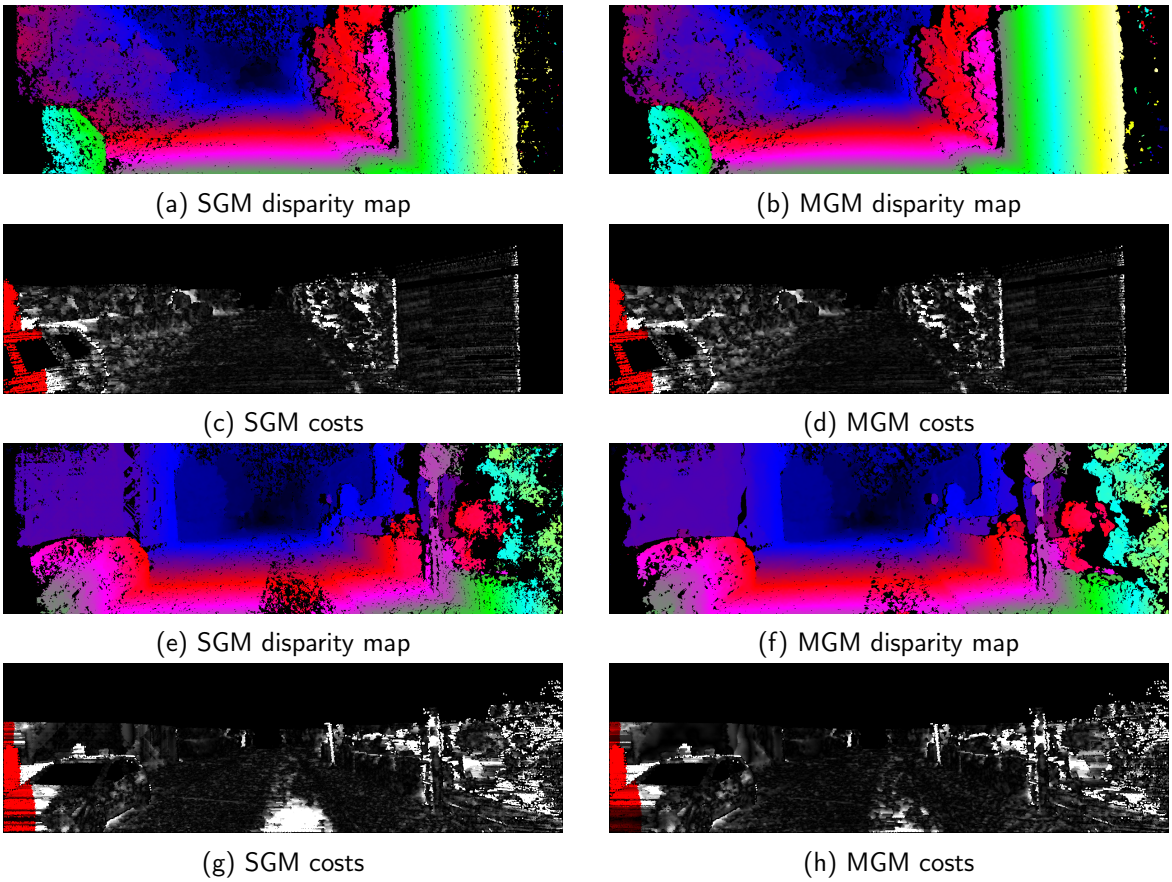


Figure 5.16 – Sparse results on stereo pairs with ground truth from the 2014 Middlebury dataset. We compare the results of our baseline implementation of SGM, MGM, and ELAS [Geiger et al. 2011].



(a) SGM disparity map

(b) MGM disparity map

(c) SGM costs

(d) MGM costs

(e) SGM disparity map

(f) MGM disparity map

(g) SGM costs

(h) MGM costs

Figure 5.17 – Results of base-SGM and MGM on two stereo pairs from the training dataset of the KITTI challenge.

propagation. With a very small overhead MGM improves up to a factor five the energy gap of SGM with respect to the best minimization algorithms. We validated experimentally that MGM produces denser results than the baseline SGM, without streaking artifacts. In summary, MGM produces better results than the baseline SGM with practically no computation overhead. It is hence an excellent candidate to be used in our 3D reconstruction pipeline presented in chapter 6.

6 A Stereo Pipeline for Pushbroom Images: S2P

In this chapter we propose a fully automatic and modular stereo pipeline to produce digital elevation models from satellite images. The aim of this new pipeline is to use (and test) off-the-shelf computer vision tools, while abstracting from the complexity associated to satellite imaging. To this aim, images are cut in small tiles for which we proved in chapter 3 that the pushbroom geometry is accurately approximated by the affine camera model. These tiles are then processed with standard rectification and stereo matching tools. The specifics of satellite imaging such as pointing accuracy refinement, estimation of the initial elevation from SRTM data, and geodetic coordinate systems are handled transparently by the pipeline. We demonstrate the robustness of this approach on a large database of satellite images and by providing an online demo of its implementation.

The 3D reconstruction pipeline presented in this chapter can be tested online on any crop of a satellite image from our test database at `http://dev.ipol.im/~carlo/s2p`.

6.1 Motivation

This chapter wraps together the results of chapters 3, 4 and 5 to present `S2P`, an automatic 3D reconstruction pipeline for satellite images meant to be modular and generic. Its development was motivated by the recent availability of high resolution images from new satellites with stereo capabilities such as Pléiades, which were briefly presented in the general introduction (see chapter 1, page 16). Even if most of the experiments described here were carried on Pléiades images, `S2P` can also be used on images from other satellites such as WorldView, QuickBird, SPOT and Ikonos.

The philosophy of `S2P` is to isolate the 3D reconstruction problem from the complexities associated to satellite imaging. To that aim the satellite images are processed by small tiles. This permits to locally approximate the pushbroom geometry with an affine model, which in turn allows to rectify the tiles using standard computer vision tools [Hartley and Zisserman 2004]. As was seen in chapter 3, the rectification error obtained on the tiles is below the tenth of pixel, improving the state of the art by one order of magnitude [Oh et al. 2010]. Each rectified tile is then processed using off-the-shelf stereo matching algorithms such as SGM or MGM, which were presented in chapter 5.

The pipeline deals transparently with inaccuracies of the sensor attitude, by estimating relative corrections for each tile without need of *ground control points* [Hanley et al. 2002, Grodecki and Dial 2003, Fraser and Hanley 2005], as was exposed in chapter 4. These local corrections are then combined in a global correction for the entire image, which is used to perform a consistent 3D triangulation. Information from the SRTM database (see appendix B) is automatically incorporated to identify corresponding regions in both images, and to estimate the disparity range of the scene.

The proposed pipeline also handles three-view stereo datasets. In this case two stereo pairs are processed independently, then the resulting elevation models are merged to increase the coverage. Figure 6.5 shows an example of digital elevation model obtained with this method. This automatic pipeline is available online[1] for testing. It can be tested on any images, as soon as the associated RPC coefficients (see section 2.5) are provided.

Similarly to previous works [Wohlfeil et al. 2012, D'Angelo and Kuschk 2012, D'Angelo and Reinartz 2012, Kuschk 2013], and similarly to reference open source stereo pipelines such as ASP (Ames Stereo Pipeline) [Moratto et al. 2014], developed by NASA, and MicMac [Pierrot Deseilligny 2007, 2015], developed by IGN, `S2P` is fully automated. All tasks that used to be performed manually such as disparity range estimation, tie points selection for geometric calibration data refinement, and water masking, are performed automatically thanks to the proper use of SRTM data [Farr et al. 2007] and feature detectors such as SIFT [Lowe 2004, Rey Otero and Delbracio 2014]. But unlike these works, `S2P` does not depend on a particular stereo matching algorithm. Instead the main contribution of our work is a complete framework to evaluate any stereo matching algorithm (that works with stereo-rectified images) on pushbroom images. Indeed, the matching part is modular and anyone can plug his own algorithm, while `S2P` provides the end-to-end plumbing for tile-wise processing, geometric calibration, point cloud colorization, and raster digital surface model synthesis. As such, it greatly facilitates the assessment and comparison of stereo matching algorithms performance on complex remote sensing datasets.

6.2 Pipeline Overview

The `S2P` pipeline deals with pairs or triplets of images. Pairs and triplets are the standard stereo products proposed by the main commercial providers of satellite images such as DigitalGlobe and Airbus Defense and Space (formerly Astrium). The acquisition of multi-view datasets with more than three views during a single orbit is technically feasible, as demonstrated by the use of Pléiades *video mode* during the in flight commissioning [de Lussy et al. 2012]. But image vendors do not consider this acquisition mode as economically viable and do not sell it as a standard product.

In case of a stereo triplet, each pair out of the six possible (ordered) pairs is processed independently, and the resulting 3D point clouds are then merged. The merging procedure is not discussed in this thesis. Figure 6.1 gives an overview of the processing pipeline for a stereo pair of images. The input images are cut in small tiles, to allow a very precise rectification. The optimal size of the tiles was discussed in chapter 3. Then for each tile the RPC functions are refined (see chapter 4) and the images are rectified (see chapter 3). Each rectified tile pair is matched using some standard stereo matching algorithm, such as the popular Semi-Global Matching (SGM), or its variant More Global Matching (MGM) that we both presented in chapter 5. The local refinements from all the processed tiles are combined to compute a global correction of the RPC functions. The triangulation uses the globally corrected RPC functions, which are the same for all tiles. This ensures a perfect continuity between the 3D points computed from different tiles.

[1] `http://dev.ipol.im/~carlo/s2p`

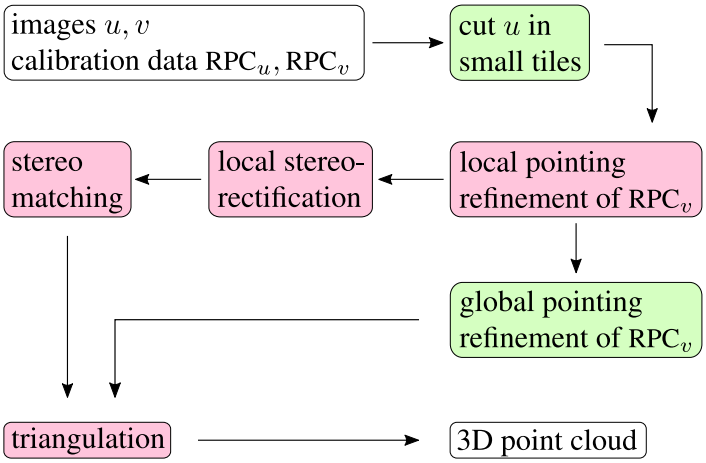


Figure 6.1 – Stereo pipeline overview. The input is a pair of images with their respective rational polynomial camera models, and the output is a digital elevation model given as a georeferenced 3D point cloud. Green blocks are applied to the whole images, while pink blocks are applied on small independent tiles. They can be processed in parallel.

6.2.1 Size of the tiles.

Extensive experiments carried out on several Pléiades datasets (see section 3.3) show that with a tile size of 1000×1000 pixels the epipolar error is always less than 0.05 pixels. This precision fits all the stereo matching algorithms, in particular SGM and MGM (chapter 5), thus our pipeline uses a tile size fixed to 1000×1000 pixels for all Pléiades images.

In case of satellites with a different behaviour, the epipolar error can be computed as a preliminary step, and the optimal tile size is automatically selected accordingly. It is important to note that this approximation is limited to satellite images. Aerial pushbroom images such as those obtained from Leica's ADS 40 or 80 cannot be rectified in that way since the trajectory of a plane is not as smooth as that of a satellite.

6.2.2 Global relative pointing error model.

The model we use to correct the pointing error on a tile relies on the validity of the affine approximation. From chapter 3 we know that this model is valid on image regions of size smaller than 1000×1000 pixels. For larger regions, the local pointing correction model may not be valid. Several authors [Fraser and Hanley 2005, Grodecki and Dial 2003, D'Angelo and Reinartz 2012] reported that the global RPC bias can be corrected with an affine transformation in image space. A simple way to estimate the optimal affine transformation is to use the local corrections computed for each tile. An example of this procedure is given in figure 6.2.

6.2.3 Triangulation

For each rectified tile we compute the disparity by applying an off-the-shelf stereo matching algorithm. The SRTM data (see appendix B) is used to estimate the initial disparity range,

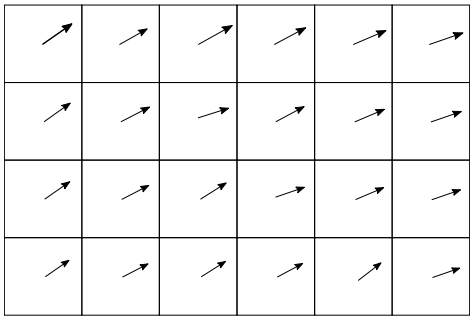


Figure 6.2 – A global pointing correction is estimated for the whole processed region of interest from the local corrections that were computed in each tile. In this example, the region was cut into 24 tiles. In each tile a corrective translation was computed and is represented by an arrow starting from the center of the tile. The 24 corrective translations are used to estimate an affine transformation that corrects the relative pointing error on the whole region.

together with the SIFT point correspondences that were used in chapter 4 to correct the local pointing error.

The disparities are then interpreted as point correspondences with the coordinates of the original (non rectified) images, as illustrated in figure 6.3. From these correspondences the 3D position of the point is triangulated using the refined RPC camera models.

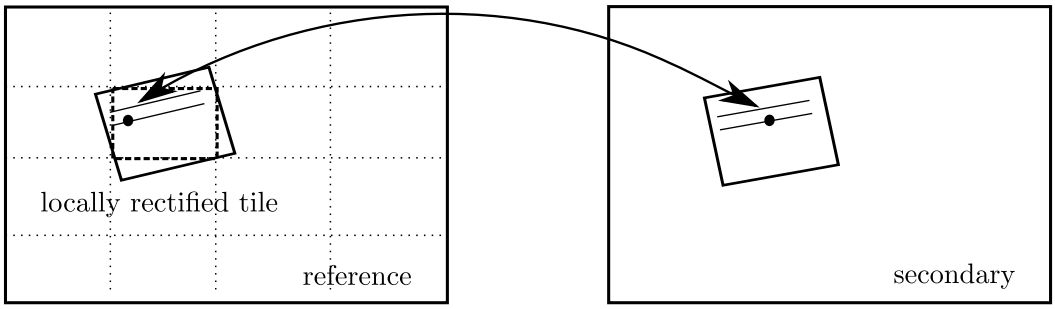


Figure 6.3 – A match on a rectified tile is interpreted as a point correspondence in the coordinate systems of the original images.

Triangulation with RPC functions. As studied in chapter 4, the limited precision of the RPC data may cause a point $\mathbf{x}'$ to be displaced from the epipolar line $\text{epi}_{uv}^{\mathbf{x}}(\mathbf{R})$ of the corresponding point $\mathbf{x}$. Algorithm 11 determines iteratively the altitude of the 3-space point defined by the point correspondence $(\mathbf{x}, \mathbf{q})$, where $\mathbf{q}$ is the projection of $\mathbf{x}'$ on the epipolar curve $\text{epi}_{uv}^{\mathbf{x}}(\mathbf{R})$. The algorithm updates an altitude hypothesis h for the point $\mathbf{x}$ in such a way that its correspondent point $\mathbf{r}_0$ is as close as possible to the match $\mathbf{x}'$. The altitude hypothesis is updated by linearly approximating the epipolar curve. Figure 6.4 illustrates the simple rationale behind this algorithm. As the epipolar lines are very smooth the parameter h_{step} can be large. We set it to 1 m.

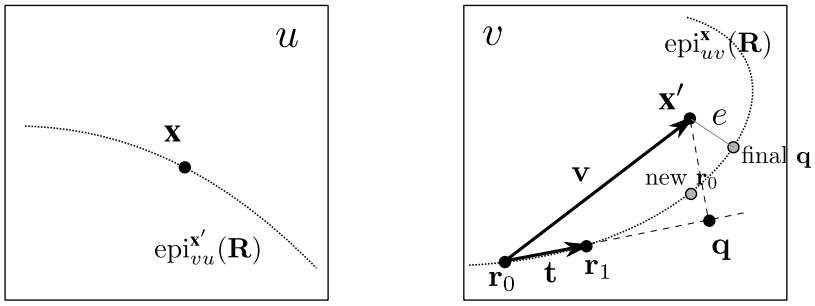


Figure 6.4 – Illustration of one iteration of Algorithm 11. The images u and v, two corresponding point $\mathbf{x}$ and $\mathbf{x}'$, and the epipolar curve $\text{epi}^{\mathbf{x}}_{uv}(\mathbf{R})$.

Algorithm 11: Altitude of a 3-space point from an image match using RPC functions.

Input: $\text{RPC}_u, \text{RPC}_v$: RPC coefficients of images u and v
$\mathbf{x}, \mathbf{x}' \in \mathbf{R}^2$: pair of corresponding points in images u and v
Output: h, e : altitude h of the triangulated 3-space point and distance e from $\mathbf{x}'$ to the epipolar curve $\text{epi}^{\mathbf{x}}_{uv}(\mathbf{R})$.

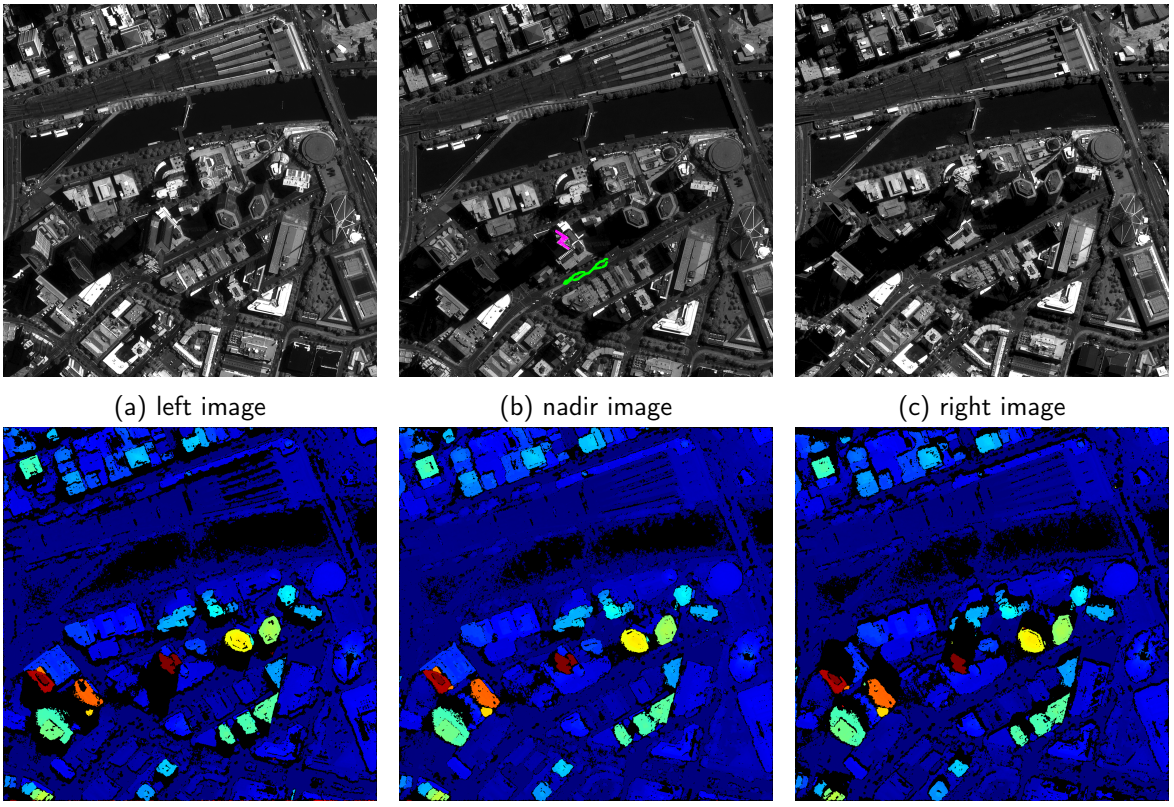
$h = 0$
$h_{\text{inc}} = \infty$
$h_{\text{step}} = 1$
while $h_{inc} > 10^{-7}$ **do**
  $\mathbf{r}_0 = \text{epi}^{\mathbf{x}}_{uv}(h)$
  $\mathbf{r}_1 = \text{epi}^{\mathbf{x}}_{uv}(h + h_{\text{step}})$
  $\mathbf{t} = \mathbf{r}_1 - \mathbf{r}_0$
  $\mathbf{v} = \mathbf{x}' - \mathbf{r}_0$
  $h_{\text{inc}} = \frac{\mathbf{t}^T \mathbf{v}}{\|\mathbf{t}\|^2}$
  $\mathbf{q} = \mathbf{r}_0 + h_{\text{inc}} \cdot \mathbf{t}$ // $\mathbf{q}$ *projection of* $\mathbf{v}$ *over* $\mathbf{t}$
  $e = \|\mathbf{q} - \mathbf{x}'\|$
  $h = h + h_{\text{inc}} \cdot h_{\text{step}}$

6.3 Results and Discussion

6.3.1 Melbourne tri-stereo

As a validation of the proposed pipeline we processed a region from a three-view stereo dataset of Melbourne. Our validation does not include ground control points. Thus we evaluated the relative precision by measuring the height of a known building. The Eureka Tower is a 297.3-meter skyscraper located in the Southbank precinct of Melbourne, which has been highlighted in figure 6.5 (a-c). The altitude estimates were computed by averaging the heights at the street level (yielding 16.15 ± 0.23 meters) and on the roof (yielding 312.97 ± 0.49 meters). Thus our estimated height of the Eureka Tower is 296.82 ± 0.72 meters.

Figure 6.5 (d-f) shows the elevation models obtained from the nadir-left and nadir-right pairs for the Melbourne dataset. Note that both images contain large occluded areas close to tall structures, however these areas are complementary. The fusion of both models exploits



(a) left image (b) nadir image (c) right image

(d) digital elevation model from nadir-left pair (e) fused digital elevation model (f) digital elevation model from nadir-right pair

Figure 6.5 – Three Pléiades images of Melbourne (a-c). The roof and street areas used for the evaluation are highlighted in (b). The elevation maps obtained by taking only two images are shown in (d) and (f), while (e) corresponds to the fusion with outlier filtering. Black areas represent rejected pixels.

this complementarity to produce a much denser elevation model.

6.3.2 La Réunion island

To illustrate the automatic power of `S2P`, the whole Réunion island was processed from Pléiades images. As the island is larger than the swath width of Pléiades, several acquisitions were needed to cover the whole island. Table 6.1 lists the 9 stereo datasets that were used for the 3D reconstruction. The geographic positions of the images are shown on figure 6.7. The reference image of each dataset was cut into tiles of size 1000×1000 pixels, which led to a total number of approximately 20 000 tiles. All these tiles were processed automatically without any human intervention and without any parameter tuning. A subsampled version of the complete 3D point cloud is shown on figure 6.8, while a full resolution version of the *Piton de la Fournaise* crater is shown on figure 6.6. Note that even if the coverage of the island is complete, some areas were hidden by clouds and consequently were not reconstructed by the algorithm. These areas are visible as blue holes in figure 6.8. A portion of the obtained digital elevation model, around the central peak of *Piton des Neiges*, is used in chapter 7 to simulate landscape evolution and water runoff.

6.4 Code and Online Demo

The `S2P` stereo pipeline described here is completely implemented and its source code has been released as open source software. It is available on github[2]. The implementation can be tested online[3] thanks to the demo framework of the IPOL journal[4]. Several stereo datasets from Pléiades and a stereo pair from WorldView-1 are available for testing. Users can also upload their own stereo datasets and run `S2P` on it. The implementation is compatible with all the stereo datasets provided by Airbus DS and DigitalGlobe, and thus could be tested on images from WorldView-2, QuickBird-1 and SPOT-6.

The choice of the stereo matching algorithm is left to the user. Several options are proposed, among which the OpenCV[5] implementation of SGM, which is included because of its low computation time. However, any other stereo matching algorithm can be used instead and several other options will soon be added to the demo.

Figure 6.9 shows several 3D point clouds that were obtained after a single click on a Pléiades image. These reconstructions can be performed online.

[2] `https://github.com/carlodef/s2p`

[3] `http://dev.ipol.im/~carlo/s2p`

[4] `http://www.ipol.im`

[5] *StereoSGBM* module in OpenCV 2.4.8 (`http://opencv.org/`), with default parameters. To filter more outliers we compute a second disparity map reversing the reference and secondary images and enforce the consistency of both maps [Fua 1993]

Dataset name	Nb. of images	Images dimensions (pix)
ST-PAUL_ST-LEU	2	$39\,164 \times 102\,038$
ST-DENIS_MAIDO	2	$39\,869 \times 116\,069$
SAINT-PIERRE	3	$38\,345 \times 38\,532$
ST-BENOIT	2	$39\,982 \times 39\,486$
ST-PIERRE_Volcan	2	$40\,000 \times 42\,192$
VOLCAN	2	$40\,000 \times 38\,582$
ST-BENOIT_STE-ROSE	2	$31\,595 \times 38\,802$
VOLCAN-ST-PHILIPPE	2	$31\,012 \times 38\,366$
ST-BENOIT-ST-JOSEPH	2	$39\,977 \times 77\,597$

Table 6.1 – Nine overlapping Pléiades stereo datasets were used to cover the whole Réunion island.

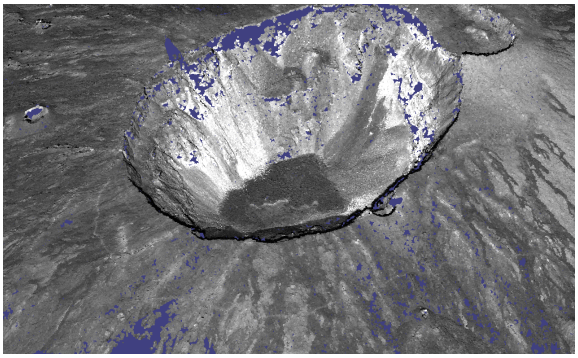


Figure 6.6 – Crater of the *Piton the la Fournaise*, located on the eastern side of La Réunion island. It was obtained with S2P from one Pléiades stereo pair at full resolution.

Figure 6.7 – Coverage of the Réunion island by Pléiades stereo images. Eight stereo pairs and one tri-stereo dataset were used. Each rectangle corresponds to a dataset. The largest image has dimensions of approximately $40\,000 \times 120\,000$ pixels. Some clouds are visible in white. They correspond to the holes of the digital elevation model of figure 6.8.

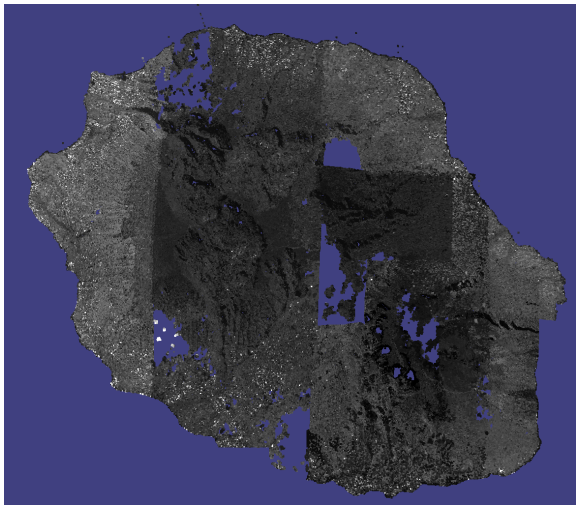
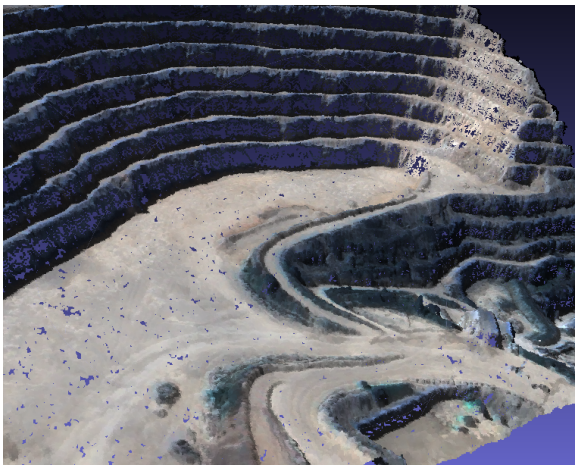
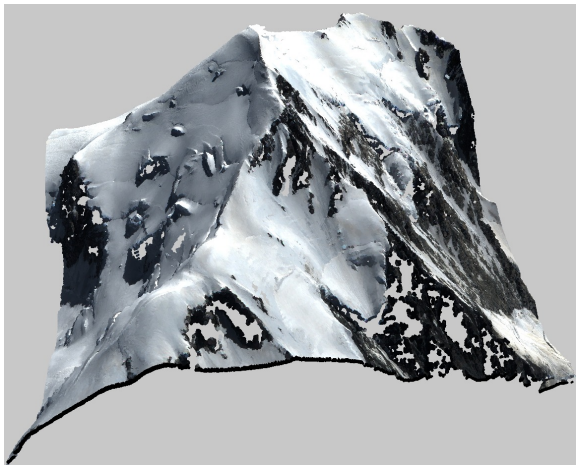


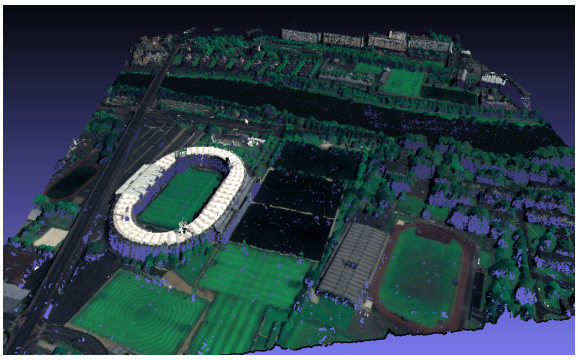
Figure 6.8 – View of the 3D point cloud produced by S2P. The points are colored with the original panchromatic channel of the respective reference images. The input images were downsampled 16 times before applying S2P, thus the point cloud shown here has a resolution of 8 meters per point. The large holes are due to clouds occluding parts of the ground.



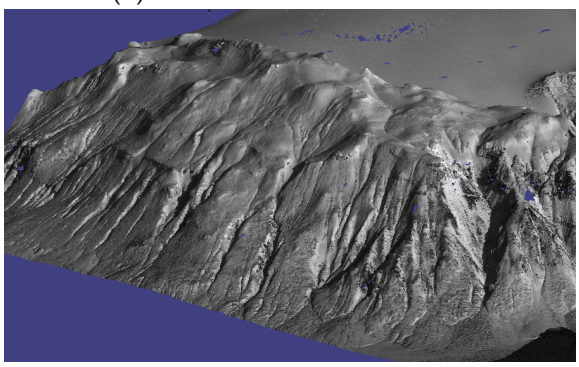
(a) Terrace in the Massif des Calanques (Marseille)



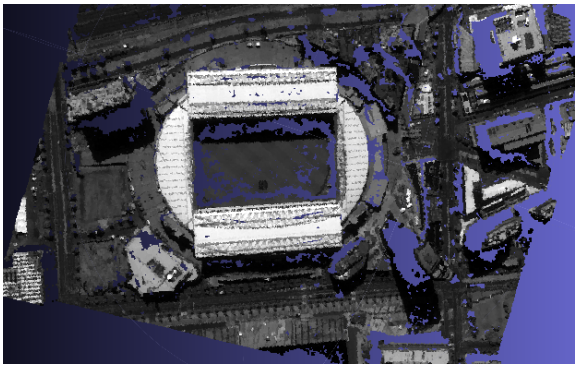
(b) The summit of Mont Blanc



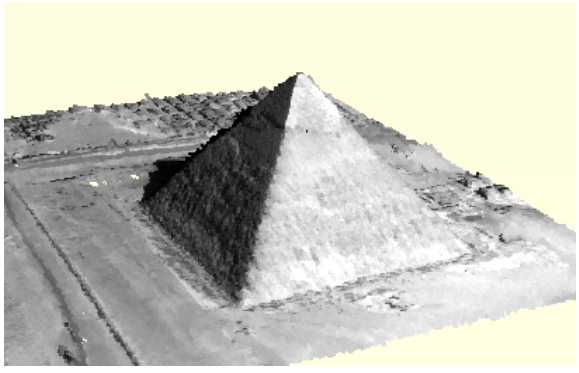
(c) Stadium Municipal (Toulouse)



(d) Glacier in Iceland



(e) Docklands Stadium (Melbourne)



(f) The Pyramid of Khafre

Figure 6.9 – 3D point clouds automatically generated from Pléiades stereo datasets, without any manual intervention, with the S2P stereo pipeline. To obtain these 3D point clouds, the user simply selected a region of interest on one of the images. This can be tested online on the IPOL S2P workshop associated to this chapter.

7 Simulation of landscape evolution on digital elevation models obtained with S2P

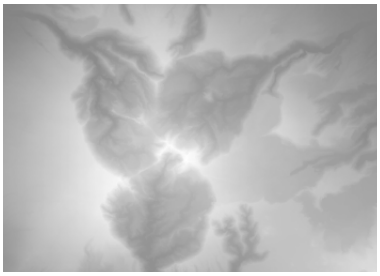
This chapter illustrates how the main physical laws proposed in landscape evolution models (LEMs) can be modeled by a system of three partial differential equations governing water runoff, stream incision, hill slope evolution and sedimentation. Several numerical experiments on high resolution digital elevation models (DEMs) obtained from stereo image pairs of the satellite Pléiades, thanks to the algorithm `S2P` presented in chapter 6, illustrate its potential to simulate the fine structure of the river network, and to understand the landscape morphology and its causes. They also permit to simulate plausible evolutions. The experiments on digital elevation models of the same site (La Réunion) are made at three different resolutions: the SRTM resolution, 90 m, and then 12 m and 4 m on digital elevation models derived from several Pléiades pairs of the same site. The experiments show that the water runoff simulation critically depends on the DEM resolution. This justifies the use of high resolution DEMs like those that can now be produced from Pléiades pairs or triplets.

The landscape evolution model presented in this chapter can be tested online on any digital elevation model at
`http://dev.ipol.im/~carlo/ipol_demo/landscape_evolution`.

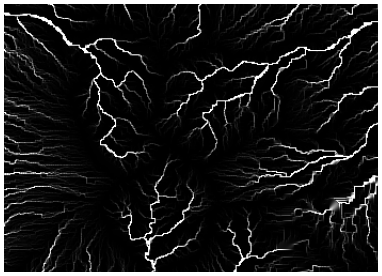
7.1 Introduction

Among other goals, geomorphological research attempts to model and explain the evolution of landscape morphology under the conjugated effects of erosion, sedimentation, chemical weathering, creep, tectonic motions, etc. Erosion (runoff) is the removal of sediment from the land surface by a fluid agent such as water or air, while sedimentation is the converse process, in which sediment from the fluid mixture settles onto the land surface. During storm events, rainfall increases the erosion and sedimentation activity. By estimating the amount of erosion and sedimentation, an estimate for the dynamics of the landscape can be obtained. This effect can be complemented by a terrain smoothing effect called *creep*, whose description and causes are multiple (water splash, chemical and thermal weathering, gravitational flow of soft soil). All of these effects and others can be modeled by a *landscape evolution model (LEM)*, summarized in a few partial differential equations. The simulation of a landscape evolution model requires a thorough knowledge of the Earth morphology, at the highest possible resolution. To that effect, accurate digital elevation models (DEMs) are indispensable and must cover large pieces of the landscape, including whole basins, islands or continents. Numerically simulated DEM evolution and its comparison with the observed landscape morphologies will eventually permit the estimation of the main model parameters for each landscape.

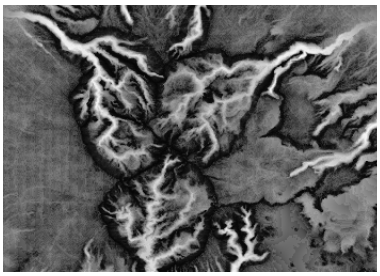
Landscape evolution modeling based on photographs goes back to Dutton and Gilbert [1877]. The observation of real landscapes and photographs and clever qualitative reasoning



(a) Initial Digital Elevation Model (DEM).

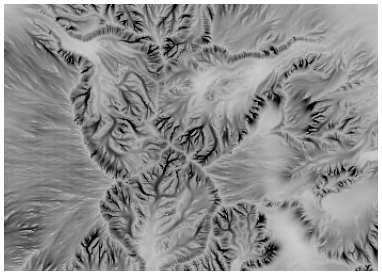


(b) River network obtained by the classic method which identifies it with the drainage basin area.

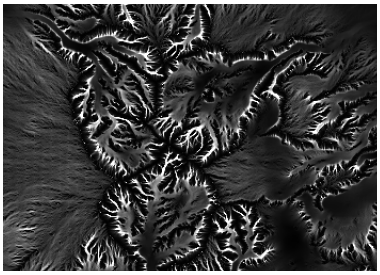


(c) The steady state value under uniform rain of the water elevation $\theta(x, y)$ when water runs on the landscape without any erosion.

(d) Final landscape.



(e) Last evolution of the landscape (elevation derivative with respect to t) just before stopping, where black denotes large values and white small values.



(f) Map $\lambda(x, y)$ of the sediment load in water at the beginning of the erosion process.

Figure 7.1 – This simulation, on a digital elevation model extracted from the SRTM database (see B) of Piton des Neiges, La Réunion, of resolution $90\,\mathrm{m}$ shows all state variables that are being displayed in a numerical landscape evolution model. The bottom row shows the evolution for parameters rain $r = 10$, erosion $\epsilon = 0.1$, creep $c = 0.1$, after removal by erosion of 5% of the landscape. See section 7.2 for more explanations on the parameters.

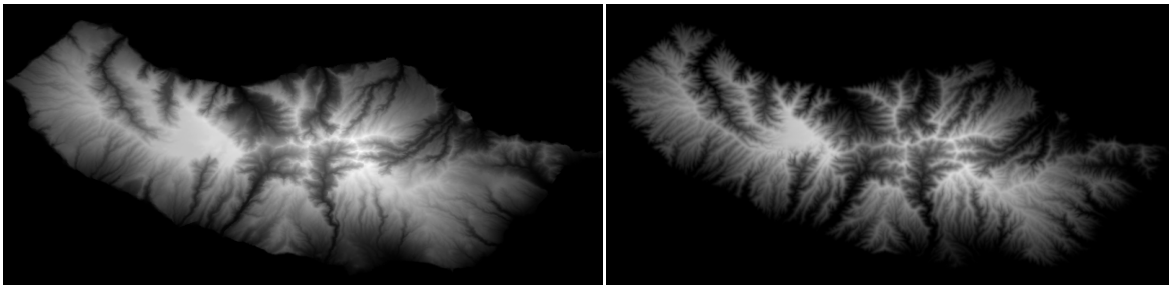


Figure 7.2 – The simulation of the correct equations with the right parameters should maintain a stable morphology (on stable landscapes). The challenge is to identify automatically the parameters for each stable landscape. This figure shows on the left a digital elevation model of Madeira island extracted from the SRTM database. On the right is the simulated model after 40% of the emerged ground has been eroded and transported away. The morphology of the two landscape models is visually similar.

led Gilbert [1909] to establish a mix of quantitative and qualitative principles governing all landscapes. The first mathematical explanation of the convexity of hilltops, ascribed to creep, is attributable to Davis [1892]. Ever since the analysis of Gilbert, landscape evolution models have involved mainly two competing factors: on the higher slopes, where water currents are weak and dispersive, soil creep dominates and the profiles are convex. On lower slopes, water flow concentrates and profiles become concave leading to the formation of valleys. Gilbert proved that creep was dominant over other factors, such as wind and rain beat.

In this chapter we show experiments on high resolution Pléiades DEMs. We base our numerical experiments on a recent LEM derived through a simplification of a rich list of complex numerical LEMs and software proposed in the past twenty years. The prominent ones are LEGS [Paik 2012], SIGNUM [Refice et al. 2012], SIBERIA, CAESAR [Hancock et al. 2011], CASCADE [Braun and Sambridge 1997], DELIM [Howard 1994], ZSCAPE, LANDSAP [Luo 2001], GOLEM [Tucker and Slingerland 1994], APERO [Carretier and Lucazeau 2005], EROS [Crave and Davy 2001], and Fastscape [Braun and Willett 2013]. The model which we follow is introduced in [Chen et al. 2014a] and detailed in [Chen et al. 2014b], where it is demonstrated that most features of previous numerical LEMs can be summarized in a system of three partial differential equations governing water runoff, erosion and sedimentation. We shall recall these equations in the next section. The objective of generic LEM models is to identify the parameters responsible for the morphology of a landscape.

The aforementioned numerical LEMs and software contain dozens and sometimes hundreds of landscape specific parameters. Yet an exhaustive numerical exploration cannot identify more than four or five independent parameters. The LEM which we shall use involves only four parameters. The numerical LEM yields for each initial DEM $z(x, y, 0)$ and each set of parameters an evolved DEM $z(x, y, t)$ under the conjugate effects of erosion and sedimentation. It also yields the final maps of the water height $\theta(x, y, t)$ and of the sediment load $\lambda(x, y, t)$ for each given steady rain rate r. Figure 7.1 shows such a simulation on a digital elevation model of the central peak of La Réunion, the Piton des Neiges, extracted from the SRTM database (see B). It was processed by the test system of three equations (7.3) – (7.5) described in section 7.2.

The validity of such a numerical program is sustainable if one can show that a landscape evolution model indeed gives a correct account of landscape evolution on various geological

time ranges. Yet, this would require the observation of DEMs on geological time scales incommensurable with human time scales. Fortunately, we found that a simple (but costly) numerical procedure can deliver a sound estimate of the parameters of a landscape. The (currently manual) procedure explores potential and reasonable parameter sets, and evolves for each one the landscape until a fixed percentage of the landscape has been scraped and transported by the erosion-sedimentation process. If the landscape morphology is stable and if the right parameters have been found, one can expect that the evolved landscape maintains its morphology (same valleys, similar slopes, etc.). This manual procedure should evolve into a systematic exploration on a large set of (stable) landscapes. This will require an adequate criterion to decide if a morphology is stable under the simulated weathering process. A tentative norm for such morphology stability measurements could be the Sobolev semi-norm

$$\int |\nabla(z(x,y,t) - z(x,y,0))| dxdy \tag{7.1}$$

where $z(x,y,t)$ is the DEM elevation at time t and $t = 0$ denotes the observed DEM.

We already succeeded in some examples to find manually the right parameter set. On a digital elevation model of Madeira island, extracted from the SRTM database (see B), we tested our three equations model with many different parameter sets, until we found one for which the landscape morphology was stable even after 40% of the emerged ground had been eroded and transported away. A visual exploration shows a completely stable morphology after this drastic evolution. These results are presented in Figure 7.2.

Yet our first numerical experiments on Pléiades data turn out to have been applied on a very young and unstable volcanic landscape, the La Réunion island. For such an unstable landscape, where basins are still in formation, we can nevertheless find sets of parameters for which the existing basins seem to extend in a natural way. One can thus envisage various evolutions of this landscape after the volcanic activity gets extinct. Figure 7.3 displays such plausible evolutions toward a stable landscape, mainly depending on the balance between the creep, rain and erosion rates.

7.2 The main landscape evolution equations

The numerical LEM used here was introduced in [Chen et al. 2014a] and extended in [Chen et al. 2014b]. In the forthcoming equations, lowercase letters denote functions of the landscape depending on two geographic coordinates x, y and on time t. $\nabla\phi$ denotes the gradient of a scalar function $\phi(x,y)$, $\Delta\phi$ its Laplacian, and $\nabla \cdot \mathbf{v}$ the divergence of a vector field $\mathbf{v}(x,y)$. The function $\theta(x,y,t)$ denotes the water height at (x,y) and time t, and similarly $z(x,y,t)$ is the bedrock surface elevation. The landscape altitude, i.e. land surface elevation plus water height, is thus given by $z + \theta$, and $|\nabla(z+\theta)|$ is the landscape slope. The sediment load in water is denoted by λ, so that $\frac{\lambda}{\theta}$ is the sediment density in water. Finally $\mathbf{v}$ is the water velocity. All of these state variables depend on (x,y,t), but for the sake of simplicity, we shall in general omit the triplet (x,y,t) in the equations.

There are also landscape specific parameters in the equations, that for simplicity we assume constant in time and space on each landscape. These parameters, that can be tuned to find the right set, are the creep rate c, the erosion rate ϵ, the rain rate r, the sedimentation

rate s and the erosion exponent m. Since $m = \frac{1}{2}$ was fixed, the numerical model actually depends on four parameters.

We picked the simplest possible version for the generalized Gauckler-Manning, Saint-Venant law [Gauchler 1867] so that the water velocity is simply the negative of the slope,

$$\mathbf{v} = -\nabla(z + \theta). \tag{7.2}$$

There is no simpler way to express the common sense observation that water runs downs the slope. The system used in all experiments governing the evolution of the landscape elevation z, the water depth θ and the sediment load λ in water is composed of three equations:

- the water conservation and transport law

$$\frac{\partial \theta}{\partial t} = \nabla \cdot (\theta \nabla(z + \theta)) + r \tag{7.3}$$

- the landscape evolution equation with creep, incision law and sediment deposition

$$\frac{\partial z}{\partial t} = c\Delta z - \epsilon \theta^m |\nabla(z + \theta)|^{2m} + s\frac{\lambda}{\theta} \tag{7.4}$$

- the conservation of sediment law

$$\frac{\partial \lambda}{\partial t} = \nabla \cdot (\lambda \nabla(z + \theta)) + \epsilon \theta^m |\nabla(z + \theta)|^{2m} - s\frac{\lambda}{\theta} \tag{7.5}$$

Simulating this system still depends on five parameters, the erosion exponent m, the rain rate r, the creep rate c, the sedimentation rate s, and the erosion rate ϵ. A stopping time must also be specified and we decided to fix it as the percentage of DEM erosion, namely the ratio p of the average eroded elevation in the DEM to the initial average elevation above its minimal level. We explored typical values of 5% to 30% that make the evolution visually conspicuous. Since $m = \frac{1}{2}$ the actual number of parameters remains four.

The first equation (7.3) in this three equation system is the simplest possible water runoff formulation, which can be viewed as a minimal version of Saint-Venant shallow water equations. It simply states that water runs off at each point (x, y) in the direction opposite to its elevation gradient $\nabla(z+\theta)(x, y)$. The source term r expresses that rain is falling at constant rate all over the landscape. Of course nothing prevents using a space or time variable $r(x, y, t)$ if such rainfall data rates are available.

The second equation (7.4) contains all of the water-ground interactions. The first one is Gilbert's creep evolution

$$\frac{\partial z}{\partial t} = c\Delta z \tag{7.6}$$

by which a landscape tends to get smooth and round by a diffusion process. The constant c reflects a diffusion speed depending on soil conditions and on the previously mentioned various perturbing factors (rain splash, wind, chemical weathering, ...). Such a creep term is found, among others, in [Fernandes and Dietrich 1997], the GOLEM numerical simulation system [Tucker and Slingerland 1994, Moglen et al. 1998, Simpson 2003, Culling 1960, Willgoose et al. 1991]. The second term $-\epsilon\theta^m|\nabla(z + \theta)|^{2m}$ in (7.4) expresses the stream incision law

[Niemann et al. 2001, Dietrich and Perron 2006]. It states that the erosion rate in a channel increases with the flux of water in the channel and with the local gradient. The preferred exponent in many models is $m = \frac{1}{2}$. The third term in the second equation (7.4) $-s\frac{\lambda}{\theta}$ is Exner's sedimentation law by which the sedimentation rate is proportional to the density of sediment $\frac{\lambda}{\theta}$ in water. To summarize, the second equation expresses that the elevation $z(x, y, t)$ evolves under the conjugate actions of creep, erosion and sedimentation.

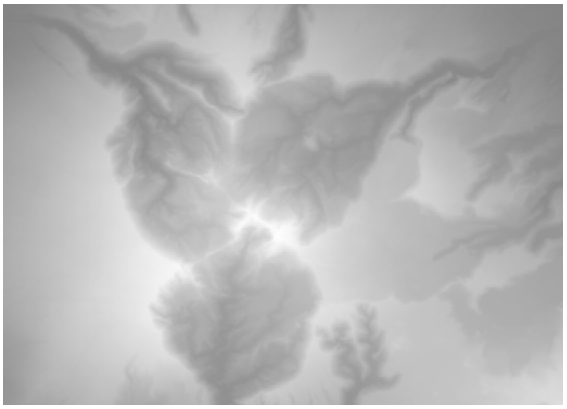
The third equation (7.5) simply expresses the conservation and transport law of the sediment $\lambda(x, y)$ carried by the water. The first term is the runoff term, strictly analogous to the one present in (7.3) for water runoff, as sediment is carried by water at the same horizontal velocity. The second and third terms are the opposite of the terms present in (7.4). Indeed, the terrain scraped by erosion becomes transported sediment, and can sediment to become terrain again.

The simulations of Figure 7.3 illustrate the variety of morphologies that can be reached from an initial DEM by varying the parameters of the two-equation model, namely equations (7.3 – 7.4) (in that way, no sedimentation occurs, which makes sense in the sloppy La Réunion). Some of the evolutions maintain a qualitative landscape morphology similar to the original, while others create new basins and rivers and evolve the DEM toward a different morphology. For example results 7.3(b) and 7.3(c) are obtained by fixing the same erosion percentage, 20%. But the first is still very similar to the original, showing a slow morphological evolution, while the second has created or expanded basins, as the landscape evolves to a mature form. In the last result, the exponent $m = 0.6$ in the incision law has been changed from $m = 0.5$ in the other experiments. Observe that this modification is enough to modify valley spacing on the left slope. It seems sound to deduce from this variety of results that, with only four control parameters, a large variety of landscape morphologies can be modeled by a three-equation model.

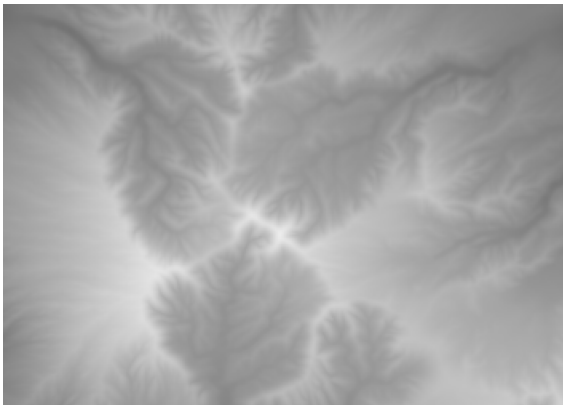
7.3 La Réunion at Pléiades resolution

This section presents two experiments for a comparative study of numerical landscape evolution simulation on the same scene, the Piton des Neiges of La Réunion, at three different resolutions. The results show a coherence of the aspect of the evolved landscapes at all resolutions. But they also show that the numerical landscape evolution leads to the emergence of a very fine resolution network. So the better the initial resolution, the better the prediction. We have shown the results of landscape evolution simulations on a digital elevation model of La Réunion extracted from the SRTM database (i.e. with resolution 90 m per pixel) in Figure 7.3. Figures 7.4 and 7.5 show analogue results at much higher resolutions on digital elevation models obtained from Pléiades stereo pairs by using the S2P *stereo reconstruction pipeline* described in chapter 6. The algorithm was applied to entire Pléiades images downsampled by factors 32 and 8, yielding digital elevation models at resolutions of respectively 16 m per pixel and 4 m per pixel.

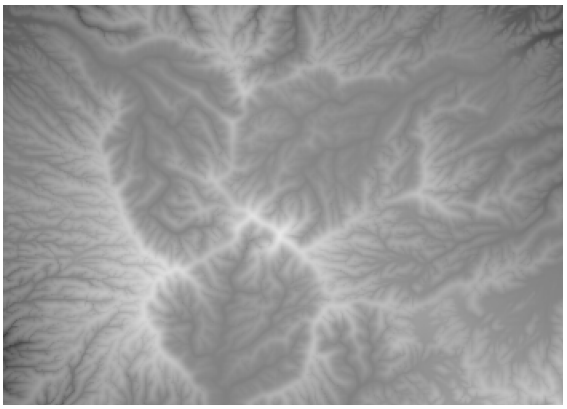
In Figure 7.4, we display first a Pléiades image of a stereo pair of Piton des Neiges, La Réunion. An image at circa 8 m per pixel is obtained from it by numerical zoom out by a factor 16. With this zoom out the image stereo pair has become virtually noiseless. Thus our automatic digital elevation model reconstruction chain S2P (see chapter 6) yields a fairly dense reconstruction. Nevertheless, interpolation remains necessary to remove holes



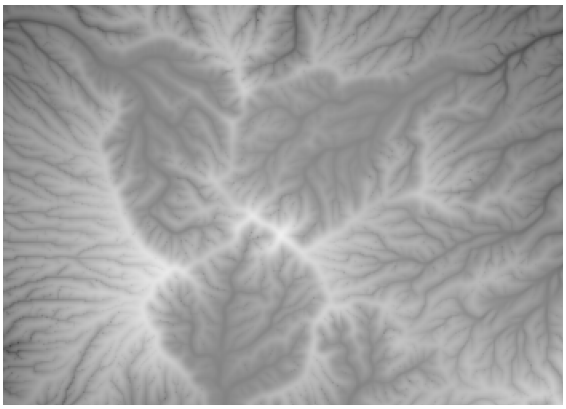
(a) Initial digital elevation model extracted from the SRTM database. The resolution is $90\,\mathrm{m}$ per pixel.



(b) Evolved landscape for $(r, \epsilon, c, p) = (5, 1, 10, 20)$.



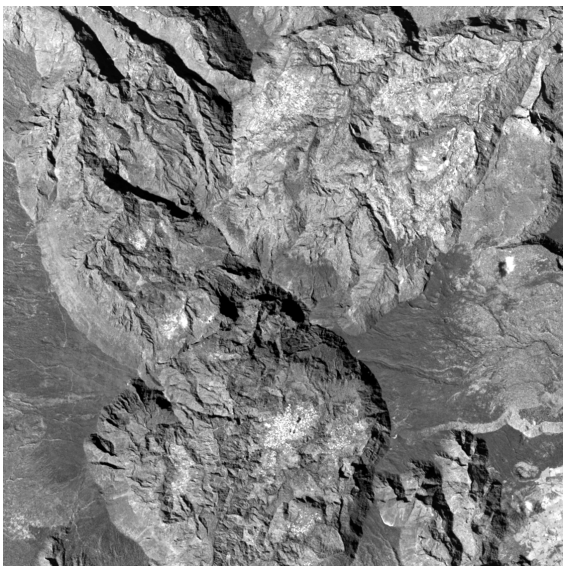
(c) Evolved landscape for $(r, \epsilon, c, p) = (1, 1, 2, 20)$.



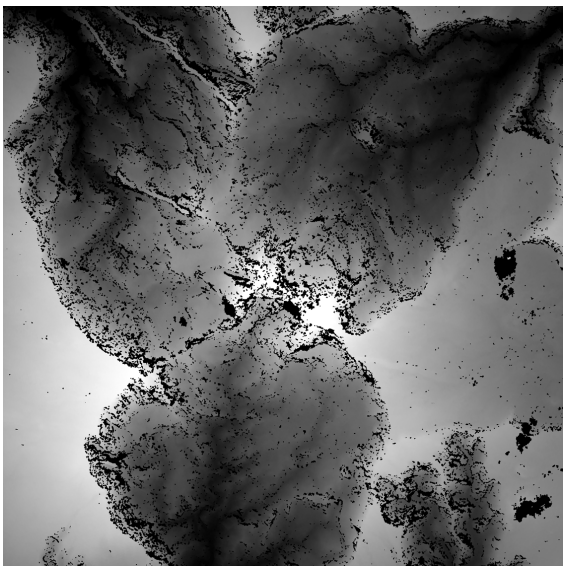
(d) Evolved landscape for $(r, \epsilon, c, p) = (1, 1, 0.5, 30)$.

Figure 7.3 – This experiment on the Piton des Neiges, La Réunion, illustrates the variety of morphologies that can be reached from an initial DEM by varying the parameters of a three-equation model. Notice how some of the evolutions maintain a similar landscape, while others create new basins and rivers and evolve it toward a different morphology. For example results (b) and (c) are obtained with the same final amount of scrapped land, 20%. But (b) is still very similar to the original landscape (a), showing a slow morphological evolution, while (c) has created or expanded basins, as the landscape evolves to a mature form. In the last experiment (d), the exponent $m = 0.6$ has been changed. In all other experiments it is $m = 0.5$. Observe that this modification seems to modify valley spacing on the left slope. See section 7.2 for more detail on the equations and parameters.

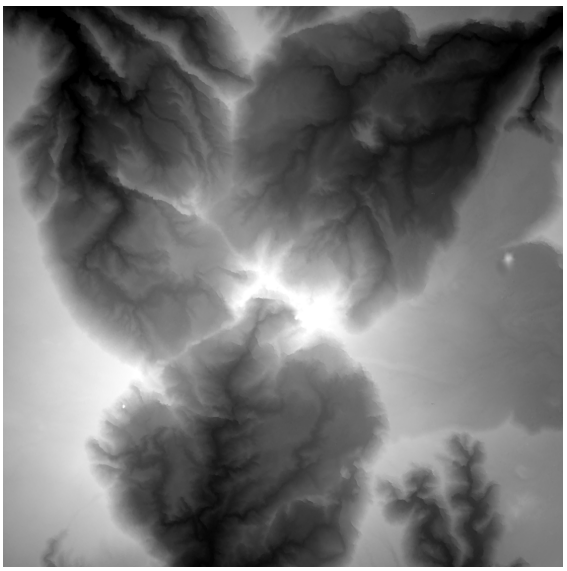
left by the stereo matching in dark regions (not lit directly by the sun) and in zones with very fine texture, which both correspond to a low SNR, making block matching fail. The image 7.4(c) shows a plausible DEM evolution, which is stopped when 10% of the terrain has been removed by erosion. The parameters in the equation are rain $r = 1$, erosion $\epsilon = 1$, creep $c = 16$, sedimentation $s = 1$. The images 7.4(e) and 7.4(f) show the water network at the end of the evolution $\theta(x, y)$ and the sediment density contained in water $\lambda(x, y)$, revealing still better the fine structure of the hydrological network, as sediment has high concentration in the fine network. This experiment illustrates the sensitivity of numerical landscape evolution to the initial resolution. Clearly the simulated network tends to increase the DEM resolution by recreating a fine channel network that was missing in the original DEM. This suggests that much will be gained by increasing still the resolution, as it is by now possible with Pléiades Earth stereo imaging. The nominal resolution of Pléiades being 0.7 m, nothing prevents from simulating landscape evolution at this resolution. This makes sense, because water runoff is very shallow except in big rivers, and therefore definitely affected by landscape roughness at the very scale of Pléiades observation. We shall be contented here to compare our previous landscape evolution at 16 m to the result of the same landscape evolution model applied this time to a circa 4 m resolution digital elevation model, shown on Figure 7.5. The stereo pair from which S2P computed the elevation model was obtained by zoom out from the same Pléiades images presented in Figure 7.4. The DEM therefore gets a four times finer resolution and was again interpolated to remove the holes where stereo matching failed. Again, a plausible evolution toward a stable landscape is shown, after 10% of the terrain has been removed by erosion. The parameters in the equation are rain $r = 1$, erosion $\epsilon = 1$, creep $c = 32$, sedimentation $s = 1$. We display the landscape evolution results in the same format as in Figure 7.4: water network at the end of the evolution $\theta(x, y)$; sediment contained in water $\lambda(x, y)$, revealing the still finer structure of the hydrological network. While both networks look compatible, it is clear that the landscape evolution at a finer scale is different. If follows from this observation, as we already anticipated, that applying LEMs at 50 cm scale is probably necessary to obtain a realistic simulation and get to the critical scales at stake in landscape evolution. This without any doubt requires a huge but feasible numerical machinery, and it requires stereo pairs at Pléiades resolution or finer.



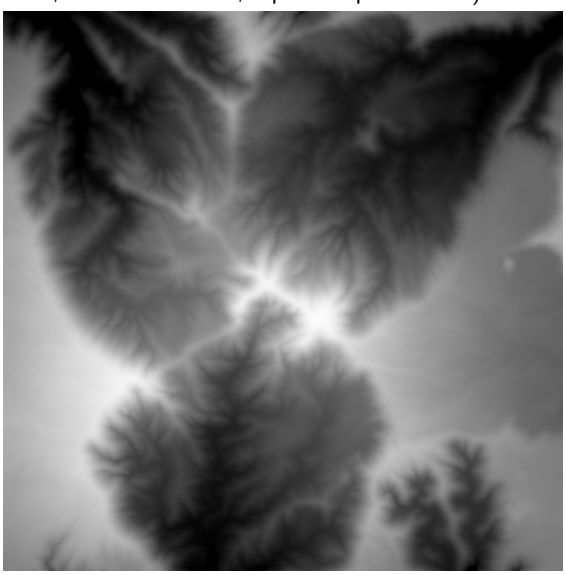
(a) Piton des Neiges, La Réunion. Crop of size $36\,000 \times 36\,000$ pixels of one of the two Pléiades panchromatic images of the stereo pair. The original resolution is of circa $0.5\,\mathrm{m}$ per pixel. The image shown here has been zoomed out for display. Copyright ©CNES 2011-15, distribution Airbus DS / Spot Image.



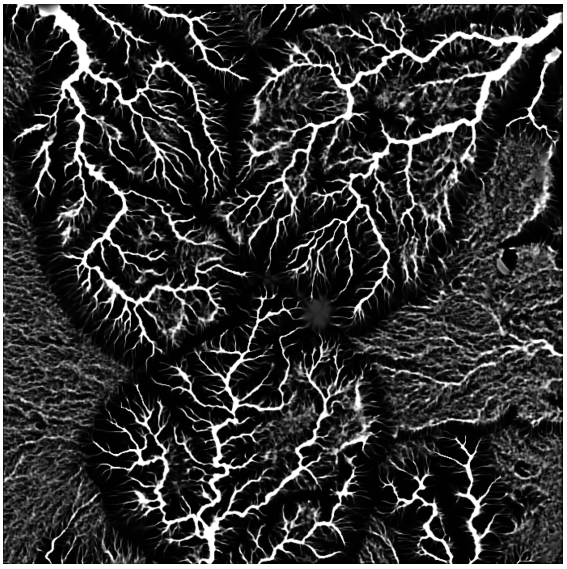
(b) Digital elevation model obtained with S2P from Pléiades images zoomed out by a factor 16 (resolution of circa $8\,\mathrm{m}$ per pixel). Bright pixels correspond to high altitudes while dark pixels correspond to low altitudes. Black pixels indicate that no altitude was computed, due to lack of information (occlusions, low light, reflections, untextured areas, repetitive patterns. . .).



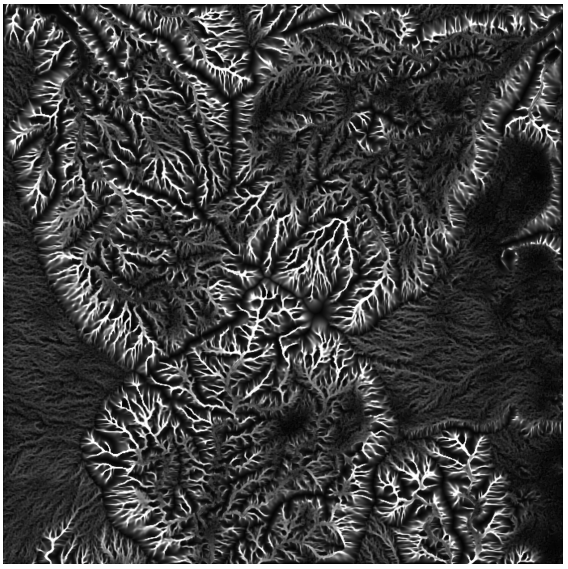
(c) Same digital elevation model, interpolated to fill the holes left by S2P in the regions where the stereo matching algorithms did not compute matches. The interpolation was performed using the *Absolutely Minimizing Lipschitz Extension* (AMLE) algorithm [Almansa et al. 2002].



(d) A plausible evolution, where 10% of the terrain has been removed by erosion. The parameters in the equation are rain $r = 1$, erosion $\epsilon = 1$, creep $c = 16$, sedimentation $s = 1$.

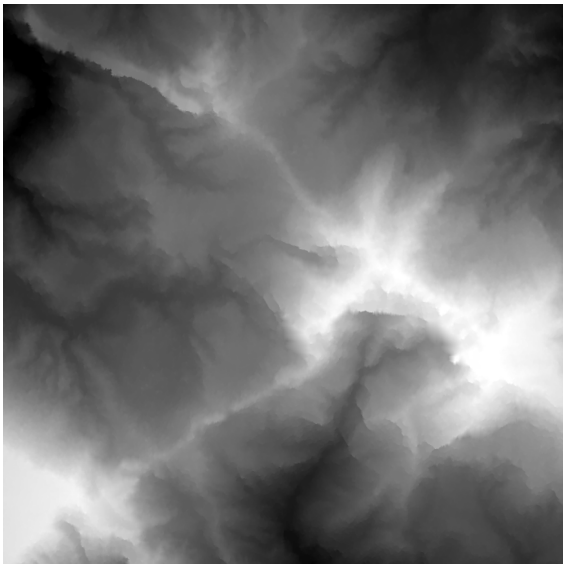


(e) Water height $\theta(x, y)$ at the end of the evolution. Bright pixels represent high values, as in terrain elevation models, showing the tree structure of the water network in white.

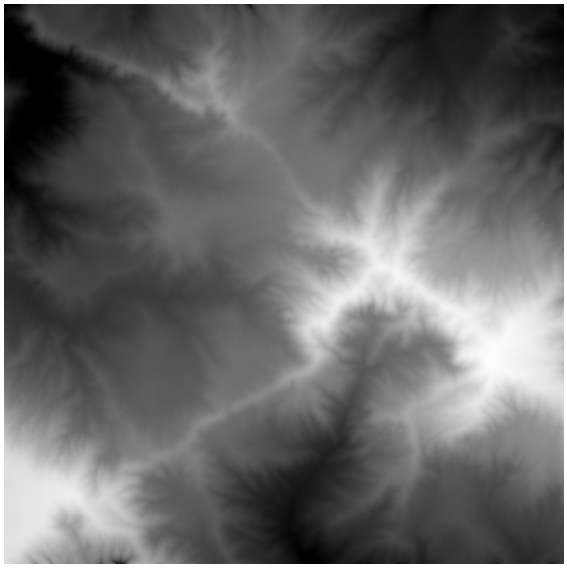


(f) Sediment load $\lambda(x, y)$ contained in water, revealing the fine structure of the hydrological network. White denotes large values and black small values.

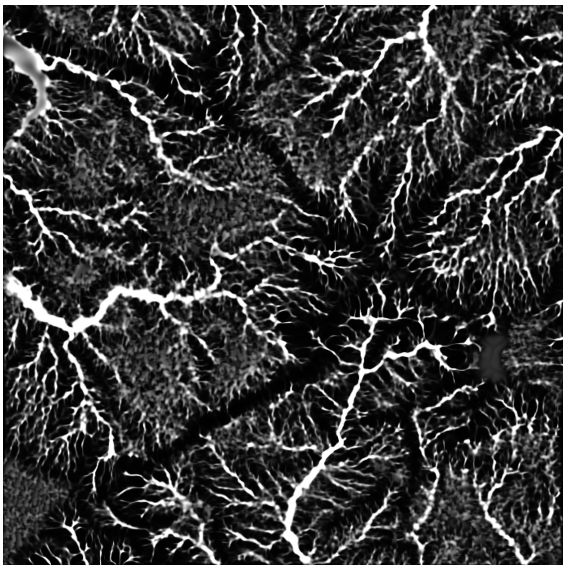
Figure 7.4 – Results of the landscape evolution simulation on an elevation model of the Piton des Neiges, La Réunion, of resolution $16\,\mathrm{m}$ per pixel, obtained from S2P (see chapter 6). The input Pléiades images, of size $36\,000 \times 36\,000$ pixels, have been downsampled by a factor 16 before being processed. The digital elevation model produced by S2P was downsampled by a factor 2 before running the landscape evolution simulation.



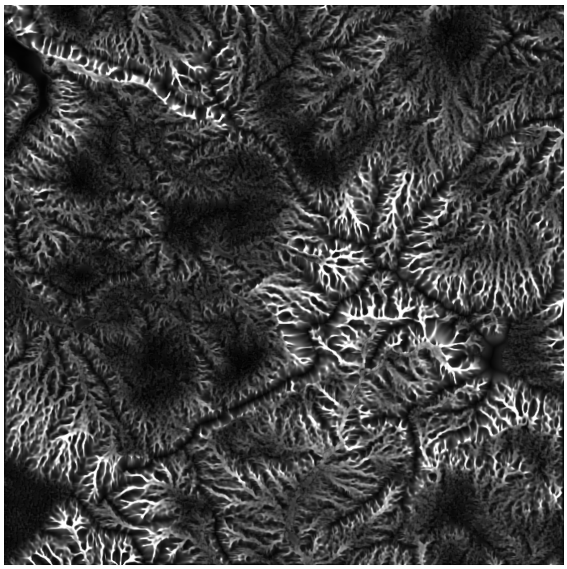
(a) Digital elevation model computed by S2P and interpolated with AMLE [Almansa et al. 2002]. The input Pléiades images, of size 9000×9000 pixels and resolution $0.5\,\mathrm{m}$ per pixel, were downsampled 8 times bringing the resolution to $4\,\mathrm{m}$ per pixel.



(b) A plausible evolution toward a stable landscape, after 10% of the terrain has been removed by erosion. The parameters in the equation are rain $r = 1$, erosion $\epsilon = 1$, creep $c = 32$, sedimentation $s = 1$.



(c) Water height $\theta(x, y)$ at the end of the evolution. Bright pixels represent high values, as in terrain elevation models, showing the water network in white.



(d) Sediment load $\lambda(x, y)$ contained in water, revealing the fine structure of the hydrological network. White denotes large values and black small values.

Figure 7.5 – Results of the landscape evolution simulation on a smaller area of the Piton des Neiges, La Réunion, at a resolution of $4\,\mathrm{m}$ per pixel.

8 Conclusion

This thesis studies all the elements needed to build a complete stereo pipeline for images taken with pushbroom sensors by Earth observation satellites. The main achievement of this work is the `S2P` software (chapter 6) which is currently used in production at CNES, the French space agency, to supply large scale digital elevation models to a broad range of academic and civilian users.

Nevertheless, many problems are still open. The solution proposed in chapter 2 to refine the orientation parameters of a pushbroom camera relies on ground control points. But the acquisition of ground control points is an expensive and tedious task that requires a lot of manual interventions. Future work should investigate the ways to refine the absolute localization of satellite images while reducing the need for manual interventions.

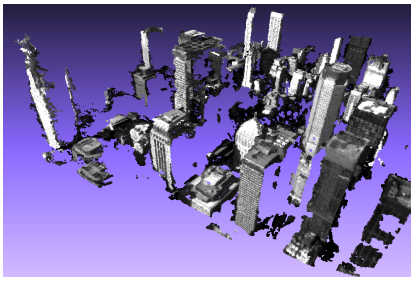
The inconsistencies between the orientation parameters of the images of a stereo dataset were shown in chapter 4 to perturb the matching process. It was shown that, on small image tiles, the induced transformation is close to a translation in image space. But it was also shown that for large image tiles the translation approximation is not valid. The correct error model for complete images should be investigated in future work.

The core of the 3D reconstruction problem is the stereo matching step, for which the popular SGM algorithm and its variant MGM were studied in chapter 5. There is still room for improvement, as shown by the error rates obtained on the Middlebury benchmark (version 3), reported on figures 5.10 (page 97) and 5.12 (page 101). Indeed, stereo matching is still a very active research field, with many papers published each year at the major computer vision and image processing conferences.

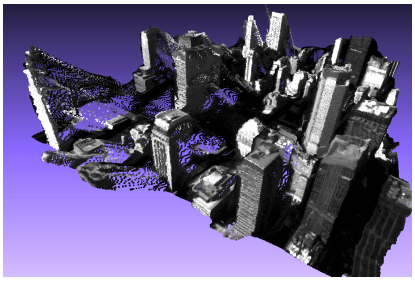
As most of the bad pixels are detected by rejection filters such as the left-to-right consistency check [Fua 1993] or the median filter [Scharstein and Szeliski 2002] (see section 5.4.5), the produced disparity maps usually present many holes, i.e. areas in which no depth information is available. For many applications, such as the water runoff simulations presented in chapter 7, these holes in the elevation model have to be filled. An automatic way to interpolate the missing data should be investigated. This is a challenging problem because the interpolation strategy must be adapted to the nature of the scene, as shown by the example presented in figure 8.1.

An alternative way to retrieve missing data in a computed 3D model is to compute another model of the same area from different images, acquired at a different date, from a different orbit, or by another remote sensor. More generally, the fusion of multi-date 3D models computed from several stereo datasets could eventually lead to a completely dense model. The main difficulty of the fusion process was illustrated by figure 4.3, page 70: the 3D models computed from different pairs of images may not be properly aligned, due to inaccuracies of the RPC functions. Each model computed by `S2P` might have an absolute bias that the relative pointing correction algorithm of chapter 4 did not correct. Thus, prior to fusion, all

(a) Pléiades view from a stereo pair over Manhattan.



(b) 3D point cloud computed by S2P from the Pléiades stereo pair, non dense.



(c) Dense result obtained from (b) by interpolating with TVL1.

Figure 8.1 – The interpolation strategy has to be adapted to the type of scene. In this example the dense results obtained with usual methods ignore the fact that the occluded surfaces are vertical. Copyright ©CNES 2011-15, distribution Airbus DS / Spot Image

the models have to be registered in 3-space on top of one of them.

A possible approach could be to find sparse correspondences between the reference frames of the multi-date stereo datasets, using SIFT [Lowe 2004, Rey Otero and Delbracio 2014]. These correspondences can be transported in 3-space thanks to the 3D models computed by S2P. A transformation of 3-space can then be estimated from the set of 3-space correspondences. The correct class of transformations has still to be investigated, but promising results were obtained using rigid transformations. Classical approaches for this 3D registration problem are the *Iterated Closest Point* (ICP) algorithm [Chen and Medioni 1992] and the *Scale Space Merging* algorithm [Digne et al. 2010, Digne 2014]. ICP estimates a parametrized global transformation between two 3D models, while scale space merging estimates a global deformation of the scene based on low spatial frequencies.

A Guidance algorithm

In our implementation of the pushbroom camera geometric simulator, the true attitude coefficients $\bar{\Phi}$ are not set directly by the user. Instead, the user chooses an initial pointing direction Ψ_x, Ψ_y and a heading γ which gives the direction of the movement of the projection of the pushbroom sensor on the ground, with respect to the local North. The attitude coefficients are computed from Ψ_x, Ψ_y and γ thanks to Algorithm 12.

Algorithm 12: How to compute attitude coefficients from a given heading and initial pointing.

Input: Ψ_x, Ψ_y: initial pointing direction
γ: heading
h_0: rough altitude of the scene
Δ_t, δ_t: duration of the acquisition and dwell time
a, i, λ_0: orbit parameters
α_0: satellite initial position on the orbit
n: number of samples used to estimated the attitude polynomials
d: degree of the polynomials used to describe the attitude functions

Output: $\varphi, \psi, \omega : [0, \Delta_t] \to \mathbf{R}^3$: attitude polynomial functions

begin

$\mathbf{X} = (\mathrm{X}, \mathrm{Y}, \mathrm{Z})^\top \leftarrow$ `target_point_orbital_frame`(Ψ_x, Ψ_y, h_0) *// algorithm 13*

for $k \in \{0, \dots, n-1\}$ **do**

  $\mathbf{v} \leftarrow$ `pushbroom_direction_on_the_ground`$(\mathbf{X}, \gamma, \frac{k\Delta_t}{n-1})$ *// algorithm 15*

  $\varphi^k, \psi^k, \omega^k \leftarrow$ `attitude_from_point_and_speed`$(\mathbf{X}, \mathbf{v})$ *// algorithm 14*

  $\mathbf{X} \leftarrow \mathbf{X} +$ `pixel_projection_width`$(\mathtt{R}(\varphi, \psi, \omega))\frac{\Delta_t}{(n-1)\delta_t}\mathbf{v}$ *// algorithm 16*

  $\mathrm{Z} \leftarrow \mathrm{Z} - (R_E + a)$

  $\mathbf{X} \leftarrow \mathtt{P}_{\text{rotational, orbital}}(\frac{(k+1)\Delta_t}{n-1}, \lambda_0, i, \alpha_0)^{-1}\mathtt{P}_{\text{rotational, orbital}}(\frac{k\Delta_t}{n-1}, \lambda_0, i, \alpha_0)\mathbf{X}$

  $\mathrm{Z} \leftarrow \mathrm{Z} + (R_E + a)$

for $\theta \in \{\varphi, \psi, \omega\}$ **do**

  $\theta \leftarrow$ least squares degree d polynomial fit of points $\{\frac{k\Delta_t}{n-1}, \theta^k\}_{k=0,\dots,n-1}$

return φ, ψ, ω

Algorithm 13: Orbital coordinates of the intersection of the optical axis and the ground

Input: Ψ_x, Ψ_y: initial pointing direction
h_0: altitude of the scene
a: flying altitude of the satellite
Output: $\mathbf{X}$: coordinates of the intersection of the optical axis and the ground
begin
1 $\mathbf{E} = (0, 0, R_E + a)^\top$
2 $\mathbf{v} = (\tan(\Psi_y), -\tan(\Psi_x), 1)^\top$
3 $\mathbf{X} \leftarrow$ intersection of the line directed by $\mathbf{v}$ and the sphere with center $\mathbf{E}$ and radius $R_E + h_0$

Algorithm 14: Compute the satellite attitude from a targeted point and a pushbroom direction

Input: $\mathbf{X} = (\mathrm{X}, \mathrm{Y}, \mathrm{Z})$: point on Earth, given by its coordinates in the orbital frame
$\mathbf{v}$: movement of the pushbroom array, projected on the ground, given in the orbital frame.
Output: φ, ψ, ω: attitude angles
begin
1 $\varphi \leftarrow -\arctan(\frac{\mathrm{Y}}{\mathrm{Z}})$
2 $\psi \leftarrow \arcsin(\frac{\mathrm{X}}{\|\mathbf{X}\|})$
3 $\mathbf{n} \leftarrow \mathbf{X} - (0, 0, R_E + a)^\top$
4 $\mathbf{w} = \mathbf{n} \times \mathbf{v}$
5 $\mathbf{X} = -\mathbf{n} \times \mathtt{R}(\varphi, \psi, 0)(0, 1, 0)^\top$
6 $\mathbf{Y} = -\mathbf{n} \times \mathtt{R}(\varphi, \psi, 0)(1, 0, 0)^\top$
7 **if** $\mathbf{w} \cdot \mathbf{Y} = 0$ **then**
8 $\omega \leftarrow \frac{\pi}{2}$
9 **else**
10 $\omega \leftarrow \arctan(\frac{\mathbf{w} \cdot \mathbf{X}}{\mathbf{w} \cdot \mathbf{Y}})$
11 **if** $\mathbf{v} \cdot \mathbf{Y} \sin\omega < \mathbf{v} \cdot \mathbf{X} \cos\omega$ **then**
12 **if** $\omega < 0$ **then**
13 $\omega \leftarrow \omega + \pi$
14 **else**
15 $\omega \leftarrow \omega - \pi$
16 return φ, ψ, ω

Algorithm 15: Pushbroom direction on the ground

Input: $\mathbf{X}$: orbital coordinates of the ground point currently targeted
γ: heading, i.e. direction of the pushbroom array movement, on the ground
t: time elapsed since the beginning of the acquisition, in seconds

Output: $\mathbf{v}$: orbital coordinates of a vector giving the pushbroom movement direction, projected on the ground

begin

$\mathbf{E} \leftarrow (0, 0, R_E + a)^\top$

$\lambda, \theta \leftarrow$ longitude, latitude of $\mathtt{P}_{\text{rotational, orbital}}(t, \lambda_0, i, \alpha_0)(\mathbf{X} - \mathbf{E})$

$\mathbf{v} \leftarrow (0, \sin\gamma, \cos\gamma)^\top$

return $\mathtt{P}_{\text{rotational, orbital}}(t, \lambda_0, i, \alpha_0) \begin{bmatrix} c_\lambda & -s_\lambda & 0 \\ s_\lambda & c_\lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & -s_\theta \\ 0 & 1 & 0 \\ s_\theta & 0 & c_\theta \end{bmatrix} \mathbf{v}$

Algorithm 16: Compute the width of the projection of a pixel on the ground

Input: $\mathtt{M}$: change of coordinates matrix between orbital frame and camera frame
$\mathbf{x} = (x, y)$: coordinates of the pixel in the image plane

Output: width, in meters, of the projection of the pixel on the ground, in the pushbroom direction (orthogonal to the pushbroom movement)

begin

$\mathbf{v} \leftarrow \mathtt{M}(-x, -y, f)^\top$

$\mathbf{E} \leftarrow (0, 0, R_E + a)^\top$

$\mathbf{X} \leftarrow$ intersection of the line directed by $\mathbf{v}$ and the sphere with center $\mathbf{E}$ and radius R_E

$\Pi \leftarrow$ plane normal to $\mathbf{X} - \mathbf{E}$ passing by $\mathbf{X}$

$\mathbf{v}_1 \leftarrow \mathtt{M}(-x - \frac{w}{2}, -y, f)^\top$

$\mathbf{v}_2 \leftarrow \mathtt{M}(-x + \frac{w}{2}, -y, f)^\top$

$\mathbf{p}_1 \leftarrow$ intersection of the line directed by $\mathbf{v}_1$ with the plane Π

$\mathbf{p}_2 \leftarrow$ intersection of the line directed by $\mathbf{v}_2$ with the plane Π

return $\|\mathbf{p}_1 - \mathbf{p}_2\|$

B The SRTM database

A freely available model of the whole surface of the Earth was acquired by the Shuttle Radar Topography Mission (SRTM) in the early 2000's [Farr et al. 2007]. This international research effort obtained a digital surface model on a near-global scale from latitude 56°S to latitude 60°N and generated a complete digital topographic database of the Earth. The resolution of the data is three arcseconds, i.e. 90 m.

Given the longitude λ and latitude θ of a point on the ground, we can thus interpolate its altitude from a 90 m sampling, obtaining a mapping h_{SRTM} from the Earth surface to $\mathbf{R}$:

$$\begin{aligned} h_{\text{SRTM}} \colon \mathbf{R}^2 &\longrightarrow \mathbf{R} \\ (\lambda, \theta) &\mapsto h. \end{aligned} \tag{B.1}$$

This mapping can be combined with the RPC localization and projection functions of a given image u to estimate the altitude of the 3D point corresponding to a specified pixel $\mathbf{x}$. Using a fixed point algorithm, we can define a mapping $h_{\text{SRTM},u}$ from the domain of image u to $\mathbf{R}$ that associates an altitude to each pixel:

$$\begin{aligned} h_{\text{SRTM},u} \colon \mathbf{R}^2 &\longrightarrow \mathbf{R} \\ \mathbf{x} &\mapsto h. \end{aligned} \tag{B.2}$$

The SRTM data can thus be used, together with the RPC localization and projection functions, to estimate the altitude range of the 3-space points imaged in a given tile of the image u. This estimation is needed for the computation of the virtual matches used in the image rectification algorithm 6. The SRTM resolution is 90 m, which corresponds to 130 pixels in a Pléiades nadir image. Thus using a regular grid on the image domain Ω, with a step smaller than 130 pixels, and evaluating the $h_{\text{SRTM},u}$ mapping on it allows to get all the available altitude samples. Then we can extract the extrema altitudes on the image domain Ω:

$$h_m = \inf_{\mathbf{x} \in \Omega} h_{\text{SRTM},u}(\mathbf{x}), \tag{B.3}$$

$$h_M = \sup_{\mathbf{x} \in \Omega} h_{\text{SRTM},u}(\mathbf{x}). \tag{B.4}$$

This altitude range is expanded by 100 m in each direction, to take into account the low sampling of SRTM data: an offset of 100 m should be enough to compensate SRTM smoothing, except for the few areas containing very high skyscrapers.

C Notations

As far as possible we try to use the same notations as Hartley and Zisserman in their reference book on multiple view geometry [Hartley and Zisserman 2004]. Table C.1 summarizes them.

Table C.1 – Notations used in the thesis

Notation	Meaning
t	time in seconds from the beginning of the acquisition of the current image
$\mathbf{x}$	image point, usually given by its row and column coordinates
x, y	row and column coordinates of an image point
$\mathbf{X}$	3-space point, given either by its geographic or Cartesian coordinates
$\mathrm{X}, \mathrm{Y}, \mathrm{Z}$	Cartesian coordinates of a 3-space point
λ, θ	longitude, latitude of a point in the Earth-centered rotational frame
h	height of a point above the WGS 84 reference ellipsoid
f	focal length
y_0	principal point coordinate in the sensor frame
w	pixel width
$\varphi(t)$	sensor roll angle at time t measured in the local orbital frame
$\psi(t)$	sensor pitch angle at time t measured in the local orbital frame
$\omega(t)$	sensor yaw angle at time t measured in the local orbital frame
$\mathtt{R}(\varphi, \psi, \omega)$	3D rotation defined by intrinsic roll, pitch and yaw angles
Φ	set of attitude coefficients $\{(\varphi_k, \psi_k, \omega_k)\}_{k=0,\ldots,3}$
Ψ_x, Ψ_y	initial direction of the optical axis in the local orbital frame
γ	direction of the pushbroom sensor projected on the ground w.r.t. to North
$\mathbf{u}$	direction of a back-projected ray from an image point
a	flying altitude of the satellite above the Earth surface
i	inclination of the orbital plane
λ_0	longitude of the reference node
α_0	initial angular position of the satellite on the orbit
α_t	current angular position of the satellite
Δ_t	duration of the acquisition
T_S	orbital period of the satellite
R_E	radius of the Earth (assumed to be spherical)
μ	standard gravitational parameter of the Earth
T_E	duration of an Earth stellar day
$\mathbf{E}$	Earth's center of mass
$\mathbf{S}$	satellite center of mass
L	localization function, converting image to geographic coordinates

Table C.1 – (continued)

Notation	Meaning
P	projection function, converting geographic to image coordinates
P_0	initial estimation of the projection function
$J_{x,y}L$	Jacobian of L with respect to x and y
D	swath width of the satellite
n	number of ground control points
d	degree of a polynomial function
σ_{image}	std dev of the zero-mean Gaussian noise added to the image coordinates
σ_{world}	std dev of the zero-mean Gaussian noise added to the ground coordinates
η	maximum of the polynomial error functions added to the true attitudes
u, v	two stereo rectified images
w, h	images width and height
Ω	domain of the images: $\{0, ..., w-1\} \times \{0, ..., h-1\}$
N	number of pixels, i.e. $w \times h$
$\mathbf{p}, \mathbf{q}$	pixels
d	disparity value
$\mathrm{D}_{\boldsymbol{p}}$	value of disparity map D in pixel $\boldsymbol{p}$

Bibliography

A. Almansa, F. Cao, Y. Gousseau, and B. Rougé. Interpolation of digital elevation models using AMLE and related methods. *IEEE Transactions on Geoscience and Remote Sensing*, volume 40, no. 2, pages 314–325, 2002. http://dx.doi.org/10.1109/36.992791

E. D. Andersen, C. Roos, and T. Terlaky. On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming*, volume 95, no. 2, pages 249–277, 2003. http://dx.doi.org/10.1007/s10107-002-0349-3

E. P. Baltsavias and D. Stallmann. Metric information extraction from SPOT images and the role of polynomial mapping functions. In L. W. Fritz and J. R. Lucas, editors, *XVIIth ISPRS Congress Technical Commission IV: Cartographic and Data Base Applications of Photogrammetry and Machine Vision*, pages 358–364. Washington, 1992. http://dx.doi.org/10.3929/ethz-a-004336038

S. T. Barnard and M. A. Fischler. Computational Stereo. *ACM Computing Surveys*, volume 14, no. 4, pages 553–572, 1982. http://dx.doi.org/10.1145/356893.356896

E. Berthier, C. Vincent, E. Magnússon, A. P. Gunnlaugsson, P. Pitte, E. Le Meur, M. Masiokas, L. Ruiz, F. Palsson, J. Belart, and P. Wagnon. Glacier topography and elevation changes derived from Pléiades sub-meter stereo images. *The Cryosphere*, pages 2275–2291, 2014. http://dx.doi.org/10.5194/tc-8-2275-2014

F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz. PMBP: PatchMatch Belief Propagation for Correspondence Field Estimation. *International Journal of Computer Vision*, volume 110, no. 1, pages 2–13, 2014. http://dx.doi.org/10.1007/s11263-013-0653-9

S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 20, no. 4, pages 401–406, 1998. http://dx.doi.org/10.1109/34.677269

M. Bleyer and M. Gelautz. Simple but effective tree structures for dynamic programming-based stereo matching. In *Proceedings of the Third International Conference on Computer Vision Theory and Applications*, pages 415–422. SciTePress, 2008. http://dx.doi.org/10.5220/0001072904150422

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. http://dx.doi.org/10.1017/CBO9780511804441

Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 23, no. 11, pages 1222–1239, 2001. http://dx.doi.org/10.1109/34.969114

J. Braun and M. Sambridge. Modelling landscape evolution on geological time scales: a new method based on irregular spatial discretization. *Basin Research*, volume 9, no. 1, pages 27–52, 1997. http://dx.doi.org/10.1046/j.1365-2117.1997.00030.x

J. Braun and S. D. Willett. A very efficient O(n), implicit and parallel method to solve the stream power equation governing fluvial incision and landscape evolution. *Geomorphology*, volume 180–181, pages 170–179, 2013. http://dx.doi.org/10.1016/j.geomorph.2012.10.008

M. Brown, D. Burschka, and G. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 25, no. 8, pages 993–1008, 2003. http://dx.doi.org/10.1109/TPAMI.2003.1217603

S. Carretier and F. Lucazeau. How does alluvial sedimentation at range fronts modify the erosional dynamics of mountain catchments? *Basin Research*, volume 17, no. 3, pages 361–381, 2005. http://dx.doi.org/10.1111/j.1365-2117.2005.00270.x

J. B. Case. The analytical reduction of panoramic and strip photography. *Photogrammetria*, volume 22, no. 4, pages 127–141, 1967. http://dx.doi.org/10.1016/0031-8663(67)90026-9

J. Cech and R. Sara. Efficient Sampling of Disparity Space for Fast And Accurate Matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. 2007. http://dx.doi.org/10.1109/CVPR.2007.383355

A. Chen, J. Darbon, G. Buttazzo, F. Santambrogio, and J.-M. Morel. On the equations of landscape formation. *Interfaces and Free Boundaries*, volume 16, no. 1, pages 105–136, 2014a. http://dx.doi.org/10.4171/IFB/315

A. Chen, J. Darbon, and J.-M. Morel. Landscape evolution models: A review of their fundamental equations. *Geomorphology*, volume 219, pages 68–86, 2014b. http://dx.doi.org/10.1016/j.geomorph.2014.04.037

Q. Chen and V. Koltun. Fast MRF Optimization with Application to Depth Reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3914–3921. 2014. http://dx.doi.org/10.1109/CVPR.2014.500

Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, volume 10, no. 3, pages 145–155, 1992. http://dx.doi.org/10.1016/0262-8856(92)90066-C

E. Christophe, J. Inglada, and A. Giros. ORFEO Toolbox: a Complete Solution for Mapping from High Resolution Satellite Images. In C. Jun, J. Jie, and S. NAYAK, editors, *XXIst ISPRS Congress, Technical Commission IV*, pages 1263–1268. Beijing, 2008. http://www.isprs.org/proceedings/XXXVII/congress/4_pdf/221.pdf

L. Coeurdevey and C. Gabriel-Robez. Pléiades Imagery User Guide. Technical Report USRPHR-DT-125-SPOT-2.0, Astrium GEO-Information Services, 2012. http://www.geo-airbusds.com/pleiades/

A. Crave and P. Davy. A stochastic "precipiton" model for simulating erosion/sedimentation dynamics. *Computers & Geosciences*, volume 27, no. 7, pages 815–827, 2001. http://dx.doi.org/10.1016/S0098-3004(00)00167-9

M. Crespi, F. Fratarcangeli, F. Giannone, and F. Pieralice. SISAR: a rigorous orientation model for synchronous and asynchronous pushbroom sensors imagery. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences (1/W51)*. Hannover, 2007. http://www.isprs.org/proceedings/XXXVI/1-W51/

M. Crespi, F. Fratarcangeli, F. Giannone, and F. Pieralice. High Resolution Satellite Image Orientation Models. In *Geospatial Technology for Earth Observation*, chapter 4, pages 63–104. Springer US, Boston, MA, 2010. http://dx.doi.org/10.1007/978-1-4419-0050-0_4

W. E. H. Culling. Analytical Theory of Erosion. *The Journal of Geology*, volume 68, no. 3, pages 336–344, 1960. http://dx.doi.org/10.1086/626663

E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The Complexity of Multiway Cuts. In *Proceedings of the Twenty-fourth Annual ACM Symposium on Theory of Computing*, STOC '92, pages 241–251. ACM, New York, NY, USA, 1992. http://dx.doi.org/10.1145/129712.129736

P. D'Angelo and G. Kuschk. Dense multi-view stereo from satellite imagery. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 6944–6947. Munich, 2012. http://dx.doi.org/10.1109/igarss.2012.6352565

W. M. Davis. The convex profile of bad-land divides. *Science*, volume ns-20, no. 508, page 245, 1892. http://dx.doi.org/10.1126/science.ns-20.508.245

F. de Lussy, D. Greslou, C. Dechoz, V. Amberg, J.-M. Delvit, L. Lebegue, G. Blanchet, and S. Fourest. Pléiades HR in flight geometrical calibration: location and mapping of the focal plane. In *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XXXIX-B1, pages 519–523. 2012. http://dx.doi.org/10.5194/isprsarchives-XXXIX-B1-519-2012

U. Dhond and J. Aggarwal. Structure from stereo-a review. *IEEE Transactions on Systems, Man, and Cybernetics*, volume 19, no. 6, pages 1489–1510, 1989. http://dx.doi.org/10.1109/21.44067

W. E. Dietrich and J. T. Perron. The search for a topographic signature of life. *Nature*, volume 439, no. 7075, pages 411–418, 2006. http://dx.doi.org/10.1038/nature04452

J. Digne. An Analysis and Implementation of a Parallel Ball Pivoting Algorithm. *Image Processing On Line*, volume 4, pages 149–168, 2014. http://dx.doi.org/10.5201/ipol.2014.81

J. Digne, J. M. Morel, N. Audfray, and C. Lartigue. High fidelity scan merging. *Eurographics Symposium on Geometry Processing*, volume 29, no. 5, pages 1643–1651, 2010. http://dx.doi.org/10.1111/j.1467-8659.2010.01773.x

A. Drory, C. Haubold, S. Avidan, and F. A. Hamprecht. Semi-Global Matching: A Principled Derivation in Terms of Message Passing. In X. Jiang, J. Hornegger, and R. Koch, editors, *36th German Conference, GCPR 2014, Proceedings*, volume 8753 of *Lecture Notes in Computer Science*, pages 43–53. Springer International Publishing, Münster, Germany, 2014. http://dx.doi.org/10.1007/978-3-319-11752-2_4

C. E. Dutton and G. K. Gilbert. *Report on the geology of the Henry Mountains*. U.S. Government Printing Office, Washington, 1877. http://dx.doi.org/10.5962/bhl.title.51652

P. D'Angelo and P. Reinartz. DSM based Orientation of Large Stereo Satellite Image Blocks. In *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XXXIX-B1, pages 209–214. Melbourne, 2012. http://dx.doi.org/10.5194/isprsarchives-XXXIX-B1-209-2012

F. Espuny, P. Monasse, and L. Moisan. A New A Contrario Approach for the Robust Determination of the Fundamental Matrix. In F. Huang and A. Sugimoto, editors, *PSIVT*, pages 181–192. Springer, 8334, pp.181-192, 2013, Lecture Notes in Computer Science., Guanajuato, 2014. http://dx.doi.org/10.1007/978-3-642-53926-8_17

T. G. Farr, P. A. Rosen, E. Caro, R. Crippen, R. Duren, S. Hensley, M. Kobrick, M. Paller, E. Rodriguez, L. Roth, D. Seal, S. Shaffer, J. Shimada, J. Umland, M. Werner, M. Oskin, D. Burbank, and D. Alsdorf. The Shuttle Radar Topography Mission. *Reviews of Geophysics*, volume 45, no. RG2004, pages 1–33, 2007. http://dx.doi.org/10.1029/2005RG000183

P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Belief Propagation for Early Vision. *International Journal of Computer Vision*, volume 70, no. 1, pages 41–54, 2006. http://dx.doi.org/10.1007/s11263-006-7899-4

P. F. Felzenszwalb and D. P. Huttenlocher. Distance Transforms of Sampled Functions. *Theory of Computing*, volume 8, no. 1, pages 415–428, 2012. http://dx.doi.org/10.4086/toc.2012.v008a019

N. F. Fernandes and W. E. Dietrich. Hillslope evolution by diffusive processes: The timescale for equilibrium adjustments. *Water Resources Research*, volume 33, no. 6, pages 1307–1318, 1997. http://dx.doi.org/10.1029/97WR00534

C. S. Fraser, P. M. Dare, and T. Yamakawa. Digital Surface Modelling From Spot 5 HRS Imagery Using The Affine Projective Model. In *XXth ISPRS Congress*, volume XXXV-B1, pages 385–388. Istambul, 2004

C. S. Fraser and H. B. Hanley. Bias Compensation in Rational Functions for Ikonos Satellite Imagery. *Photogrammetric Engineering & Remote Sensing*, volume 69, no. 1, pages 53–57, 2003. http://dx.doi.org/10.14358/PERS.69.1.53

C. S. Fraser and H. B. Hanley. Bias-compensated RPCs for Sensor Orientation of High-resolution Satellite Imagery. *Photogrammetric Engineering & Remote Sensing*, volume 71, no. 8, pages 909–915, 2005. http://dx.doi.org/10.14358/PERS.71.8.909

D. Fritsch and D. Stallmann. Rigorous Photogrammetric Processing of High Resolution Satellite Imagery. In *ISPRS Archives*, volume XXXIII. Amsterdam, 2000

P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, volume 6, no. 1, pages 35–49, 1993. http://dx.doi.org/10.1007/BF01212430

Y. Furukawa and C. Hernández. Multi-View Stereo: A Tutorial. *Foundations and Trends in Computer Graphics and Vision*, volume 9, no. 1-2, pages 1–148, 2015. http://dx.doi.org/10.1561/0600000052

M. P. Gauchler. Etudes théoriques et pratiques sur l'écoulement et le mouvement des eaux. *Comptes rendus de l'Académie des Sciences, Paris*, volume 64, pages 818–822, 1867. http://gallica.bnf.fr/ark:/12148/bpt6k3021f/f820.image.langFR

A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*, pages 3354–3361. 2012. http://dx.doi.org/10.1109/CVPR.2012.6248074

A. Geiger, M. Roser, and R. Urtasun. Efficient Large-Scale Stereo Matching. In *Lecture Notes in Comput. Sci.*, volume 6492 LNCS, pages 25–38. 2011. http://dx.doi.org/10.1007/978-3-642-19315-6_3

G. K. Gilbert. The Convexity of Hilltops. *The Journal of Geology*, volume 17, no. 4, pages 344–350, 1909. http://dx.doi.org/10.1086/621620

A. H. Gordon. Satellite Meteorology. *Nature*, volume 195, no. 4847, pages 1161–1162, 1962. http://dx.doi.org/10.1038/1951161a0

J. Grodecki and G. Dial. Block Adjustment of High-Resolution Satellite Images Described by Rational Polynomials. *Photogrammetric Engineering & Remote Sensing*, volume 69, no. 1, pages 59–68, 2003. http://dx.doi.org/10.14358/PERS.69.1.59

A. W. Gruen. Adaptive least squares correlation: a powerful image matching technique. *South African Journal of Photogrammetry, Remote Sensing and Cartography*, volume 14, no. 3, pages 175–187, 1985. http://www.idb.arch.ethz.ch/files/alsm_awgruen.pdf

D. J. Gugan. Practical aspects of topographic mapping from SPOT imagery. *The Photogrammetric Record*, volume 12, no. 69, pages 349–355, 1987. http://dx.doi.org/10.1111/j.1477-9730.1987.tb00581.x

H. Guichard. Etude théorique de la précision dans l'exploitation cartographique d'un satellite à défilement. Application à SPOT. *Bulletin - Société Française de Photogrammétrie et de Télédétection*, , no. 90, pages 15–26, 1983. http://cat.inist.fr/?aModele=afficheN&cpsidt=9463022

T. Gumbricht. Mapping global tropical wetlands from earth observing satellite imagery. *Center for International Forestry Research*, 2012. http://dx.doi.org/10.17528/cifor/004014

R. Gupta and R. Hartley. Camera Estimation for Orbiting Pushbrooms. In *Asian Conference on Computer Vision*. 1995

R. Gupta and R. Hartley. Linear pushbroom cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19, no. 9, pages 963–975, 1997. http://dx.doi.org/10.1109/34.615446

A. F. Habib, M. Morgan, S. Jeong, and K.-O. Kim. Analysis of Epipolar Geometry in Linear Array Scanner Scenes. *The Photogrammetric Record*, volume 20, no. March, pages 27–47, 2005. http://dx.doi.org/10.1111/j.1477-9730.2005.00303.x

I. Haller and S. Nedevschi. Design of Interpolation Functions for Subpixel-Accuracy Stereo-Vision Systems. *IEEE Transactions on Image Processing*, volume 21, no. 2, pages 889–898, 2012. http://dx.doi.org/10.1109/TIP.2011.2163163

G. Hancock, T. Coulthard, C. Martinez, and J. Kalma. An evaluation of landscape evolution models to simulate decadal and centennial scale soil erosion in grassland catchments. *Journal of Hydrology*, volume 398, no. 3-4, pages 171–183, 2011. http://dx.doi.org/10.1016/j.jhydrol.2010.12.002

H. B. Hanley, T. Yamakawa, and C. S. Fraser. Sensor orientation for high-resolution satellite imagery. In *International Archives of Photogrammetry Remote Sensing and Spatial Information Science, vol. XXXIV, no. 1*, pages 69–75. 2002

R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. http://dx.doi.org/10.1017/CBO9780511811685

S. Hermann and R. Klette. Iterative Semi-Global Matching for Robust Driver Assistance Systems. In *Lecture Notes in Computer Science*, volume 7726 LNCS, pages 465–478. 2013. http://dx.doi.org/10.1007/978-3-642-37431-9_36

H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 807 – 814. IEEE, Washington, DC, USA, 2005. http://dx.doi.org/10.1109/CVPR.2005.56

H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 30, no. 2, pages 328–41, 2008. http://dx.doi.org/10.1109/TPAMI.2007.1166

H. Hirschmüller and D. Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 31, no. 9, pages 1582–99, 2009. http://dx.doi.org/10.1109/TPAMI.2008.221

H. Hirschmüller, F. Scholten, and G. Hirzinger. Stereo Vision Based Reconstruction of Huge Urban Areas from an Airborne Pushbroom Camera (HRSC). In W. Kropatsch, R. Sablatnig, and A. Hanbury, editors, *DAGM 2005*, volume 3663 of *Lecture Notes in Computer Science*, pages 58–66. Springer-Verlag Berlin Heidelberg, 2005. http://dx.doi.org/10.1007/11550518_8

A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz. Fast Cost-Volume Filtering for Visual Correspondence and Beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 35, no. 2, pages 504–511, 2013. http://dx.doi.org/10.1109/TPAMI.2012.156

A. D. Howard. A detachment-limited model of drainage basin evolution. *Water Resources Research*, volume 30, no. 7, pages 2261–2285, 1994. http://dx.doi.org/10.1029/94WR00757

K. Jacobsen. Geometric Calibration of Space Remote Sensing Cameras for Efficient Processing. In *ISPRS Archives*, pages 33–43. 1998

K. Jacobsen. Accuracy of Digital Orthophotos From High Resolution Space Imagery. In *Proceedings of ISPRS Workshop on High Resolution Mapping from Space*. Hannover, 2003

K. Jacobsen. DEM Generation by SPOT HRS. In O. Altan, editor, *ISPRS Archives – Volume XXXV Part B1*, pages 439–444. Istanbul, 2004. http://www.isprs.org/proceedings/XXXV/congress/comm1/comm1.aspx

M. Jama, C. Lewis, and D. E. Schinstock. Identifying degrees of freedom in pushbroom bundle adjustment. *ISPRS Journal of Photogrammetry and Remote Sensing*, volume 66, no. 4, pages 400–407, 2011. http://dx.doi.org/10.1016/j.isprsjprs.2011.02.001

R. A. Jarvis. A Perspective on Range Finding Techniques for Computer Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume PAMI-5, no. 2, pages 122–139, 1983. http://dx.doi.org/10.1109/TPAMI.1983.4767365

G. Joseph. *Building Earth Observation Cameras*. CRC Press, 2015. http://dx.doi.org/10.1201/b18022

L. F. Julià and P. Monasse. Bilaterally Weighted Patches for Disparity Map Computation. *Image Processing On Line*, volume 5, pages 73–89, 2015. http://dx.doi.org/10.5201/ipol.2015.123

T. Kim. A study on the epipolarity of linear pushbroom images. *Photogrammetric Engineering & Remote Sensing*, volume 66, no. 8, pages 961–966, 2000. http://eserv.asprs.org/PERS/2000journal/aug/2000_aug_961-966.pdf

V. Kolmogorov. Convergent Tree-Reweighted Message Passing for Energy Minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 28, no. 10, pages 1568–1583, 2006. http://dx.doi.org/10.1109/TPAMI.2006.200

V. Kolmogorov, P. Monasse, and P. Tan. Kolmogorov and Zabih's Graph Cuts Stereo Matching Algorithm. *Image Processing On Line*, volume 4, pages 220–251, 2014. http://dx.doi.org/10.5201/ipol.2014.97

V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proceedings Eighth IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 508–515. IEEE Comput. Soc, 2001. http://dx.doi.org/10.1109/ICCV.2001.937668

N. Komodakis, G. Tziritas, and N. Paragios. Fast, Approximately Optimal Solutions for Single and Dynamic MRFs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. http://dx.doi.org/10.1109/CVPR.2007.383095

G. Konecny. *Geoinformation*. Taylor & Francis, Abingdon, UK, second edition, 2003. http://dx.doi.org/10.4324/9780203469644

G. Konecny, P. Lohmann, H. Engel, and E. Kruck. Evaluation of SPOT Imagery on Analytical Photogrammetric Instruments. *Photogrammetric Engineering & Remote Sensing*, volume 53, no. 9, pages 1223–1230, 1987. http://eserv.asprs.org/PERS/1987journal/sep/1987_sep_1223-1230.pdf

V. Kratky. Universal Photogrammetric Approach to Geometric Processing of SPOT Images. In *XVIth ISPRS Congress Technical Commission IV: Cartographic and Data Bank Applications of Photogrammetry and Remote Sensing*, pages 180–189. ISPRS Archives, Kyoto, 1988. http://www.isprs.org/proceedings/xxvii/congress/part4/

V. Kratky. On-Line Aspects of Stereophotogrammetric Processing of SPOT Images. *Photogrammetric Engineering & Remote Sensing*, volume 55, no. 3, pages 311–316, 1989a. http://eserv.asprs.org/PERS/1989journal/mar/1989_mar_311-316.pdf

V. Kratky. Rigorous photogrammetric processing of SPOT images at CCM Canada. *ISPRS Journal of Photogrammetry and Remote Sensing*, volume 44, no. 2, pages 53–71, 1989b. http://dx.doi.org/10.1016/0924-2716(89)90007-5

G. Kuschk. Large scale urban reconstruction from remote sensing imagery. In *3D-ARCH 2013 - 3D Virtual Reconstruction and Visualization of Complex Architectures*, volume XL-5/W1, pages 139–146. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Trento, 2013. http://dx.doi.org/10.5194/isprsarchives-XL-5-W1-139-2013

H.-Y. Lee, T. Kim, W. Park, and H. K. Lee. Extraction of digital elevation models from satellite stereo images through stereo matching based on epipolarity and scene geometry. *Image and Vision Computing*, volume 21, no. 9, pages 789–796, 2003. http://dx.doi.org/10.1016/s0262-8856(03)00092-1

M. Levoy, J. Ginsberg, J. Shade, D. Fulk, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, and J. Davis. The digital Michelangelo project. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00*, pages 131–144. ACM Press, New York, New York, USA, 2000. http://dx.doi.org/10.1145/344779.344849

Z. Liu, P. Monasse, and R. Marlet. Match Selection and Refinement for Highly Accurate Two-View Structure from Motion. In *Proceedings of ECCV*, pages 818–833. Zurich, 2014. http://dx.doi.org/10.1007/978-3-319-10605-2_53

C. Loop and Z. Zhang. Computing rectifying homographies for stereo vision. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1, pages 125–131. IEEE Comput. Soc, 1999. http://dx.doi.org/10.1109/cvpr.1999.786928

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, volume 60, no. 2, pages 91–110, 2004. http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94

W. Luo. LANDSAP: a coupled surface and subsurface cellular automata model for landform simulation. *Computers & Geosciences*, volume 27, no. 3, pages 363–367, 2001. http://dx.doi.org/10.1016/S0098-3004(00)00104-7

D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, volume 194, no. 4262, pages 283–287, 1976. http://dx.doi.org/10.1126/science.968482

R. Marti, S. Gascoin, T. Houet, D. Laffly, and P. René. Evaluation du modèle numérique d'élévation d'un petit glacier de montagne généré à partir d'images stéréoscopiques Pléiades: cas du glacier d'Ossoue, Pyrenées françaises. *Revue Française de Photogrammétrie et de Télédétection*, volume 208, pages 57–62, 2014. http://www.sfpt.fr/rfpt/index.php/RFPT/article/view/107

P. Michalis and I. Dowman. A Generic Model for Along-track Stereo Sensors Using Rigorous Orbit Mechanics. *Photogrammetric Engineering & Remote Sensing*, volume 74, no. 3, pages 303–309, 2008. http://dx.doi.org/10.14358/PERS.74.3.303

P. Michel, C. Jean-Philippe, T. Claire, and F. Delphine. Potential of pleiades VHR data for mapping applications. In *2013 IEEE International Geoscience and Remote Sensing Symposium - IGARSS*, pages 4313–4316. IEEE, 2013. http://dx.doi.org/10.1109/IGARSS.2013.6723788

G. E. Moglen, E. A. B. Eltahir, and R. L. Bras. On the sensitivity of drainage density to climate change. *Water Resources Research*, volume 34, no. 4, pages 855–862, 1998. http://dx.doi.org/10.1029/97WR02709

P. Monasse. Quasi-Euclidean Epipolar Rectification. *Image Processing On Line*, volume 1, pages 1–13, 2011. http://dx.doi.org/http://dx.doi.org/10.5201/ipol.2011.mqer

P. Monasse, J.-M. Morel, and Z. Tang. Three-step image rectification. In *Procedings of the British Machine Vision Conference*, pages 89.1–89.10. 2010. http://dx.doi.org/10.5244/C.24.89

Z. Moratto, O. Alexandrov, S. McMichael, and R. Beyer. *The Ames Stereo Pipeline: NASA's Open Source Automated Stereogrammetry Software*. NASA, 2014. http://ti.arc.nasa.gov/tech/asr/intelligent-robotics/ngt/stereo

M. Morgan, K.-O. Kim, S. Jeong, and A. F. Habib. Epipolar Resampling of Space-borne Linear Array Scanner Scenes Using Parallel Projection. *Photogrammetric Engineering & Remote Sensing*, volume 72, no. 11, pages 1255–1263, 2006. http://cat.inist.fr/?aModele=afficheN&cpsidt=18251166

P. Moulon, P. Monasse, and R. Marlet. Adaptive Structure from Motion with a Contrario Model Estimation. In *Asian Conference on Computer Vision*, pages 257–270. Springer, Daejeon, 2013. http://dx.doi.org/10.1007/978-3-642-37447-0_20

J. D. Niemann, N. M. Gasparini, G. E. Tucker, and R. L. Bras. A quantitative evaluation of Playfair's law and its use in testing long-term stream erosion models. *Earth Surface Processes and Landforms*, volume 26, no. 12, pages 1317–1332, 2001. http://dx.doi.org/10.1002/esp.272

J. Oh, W. Hee Lee, C. K. Toth, D. A. Grejner-Brzezinska, and C. Lee. A Piecewise Approach to Epipolar Resampling of Pushbroom Satellite Images Based on RPC. *Photogrammetric Engineering & Remote Sensing*, volume 76, no. 12, pages 1353–1363, 2010. http://dx.doi.org/10.14358/PERS.76.12.1353

Y. Ohta and T. Kanade. Stereo by Intra- and Inter-scanline Search Using Dynamic Programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 7, no. 2, pages 139–154, 1985a. http://dx.doi.org/10.1109/tpami.1985.4767639

Y. Ohta and T. Kanade. Stereo by Two-level Dynamic Programming. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 1120–1126. 1985b. http://ijcai.org/PastProceedings/IJCAI-85-VOL2/PDF/093.pdf

A. Okamoto, S.-I. Akamatu, and H. Hasegawa. Orientation theory for satellite CCD line-scanner imageries of hilly terrains. *International Archives of Photogrammetry and Remote Sensing*, volume 29, pages 217–222, 1993

T. Ono. Epipolar resampling of high resolution satellite imagery. *International Archives of Photogrammetry and Remote Sensing*, 1988. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.40.9148

A. B. Orun and K. Natarajan. A modified bundle adjustment software for SPOT imagery and photography: tradeoff. *Photogrammetric Engineering & Remote Sensing*, volume 60, no. 12, pages 1431–1437, 1994. http://eserv.asprs.org/PERS/1994journal/dec/1994_dec_1431-1437.pdf

K. Paik. Simulation of landscape evolution using a global flow path search method. *Environmental Modelling & Software*, volume 33, pages 35–47, 2012. http://dx.doi.org/10.1016/j.envsoft.2012.01.005

N. Paparoditis and M. Pierrot Deseilligny. A multiresolution and optimization-based image matching approach: an application to surface reconstruction from SPOT5-HRS stereo imagery. In *ISPRS Archives – Volume XXXVI-1/W41*. Ankara, 2006

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Los Altos, 1988. http://dx.doi.org/10.1016/B978-0-08-051489-5.50002-3

R. Perrier, E. Arnaud, P. Sturm, and M. Ortner. Estimation of an Observation Satellite's Attitude using Multimodal pushbroom Cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 37, no. 5, pages 987–1000, 2015. http://dx.doi.org/10.1109/TPAMI.2014.2360394

M. Pierrot Deseilligny. MicMac, un logiciel pour la mise en correspondance automatique d'images dans le contexte géographique. *Bulletin d'Information Scientifique et Technique de l'IGN*, volume 77, 2007

M. Pierrot Deseilligny. MicMac, Apero, Pastis and Other Beverages in a Nutshell. *MicMac Documentation*, 2015. http://logiciels.ign.fr/?-Micmac,3-

M. Pierrot Deseilligny and I. Clery. Apero, an Open Source Bundle Adjusment Software for Automatic Calibration and Orientation of Set of Images. In *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XXXVIII-5, pages 269–276. 2012. http://dx.doi.org/10.5194/isprsarchives-XXXVIII-5-W16-269-2011

D. Poli. *Modelling of spaceborne linear array sensors*. Ph.D. thesis, Institute of Geodesy and Photogrammetry, Swiss Federal Institute of Technology, ETH, Zurich, 2005. http://dx.doi.org/10.3929/ethz-a-005051521

D. Poli. A Rigorous Model for Spaceborne Linear Array Sensors. *Photogrammetric Engineering & Remote Sensing*, volume 73, no. 2, pages 187–196, 2007. http://dx.doi.org/10.14358/PERS.73.2.187

D. Poli, F. Remondino, E. Angiuli, and G. Agugiaro. Radiometric and geometric evaluation of GeoEye-1, WorldView-2 and Pléiades-1A stereo images for 3D information extraction. *ISPRS Journal of Photogrammetry and Remote Sensing*, volume 100, pages 35–47, 2014. http://dx.doi.org/10.1016/j.isprsjprs.2014.04.007

D. Poli and T. Toutin. Review of developments in geometric modelling for high resolution satellite pushbroom sensors. *The Photogrammetric Record*, volume 27, pages 58–73, 2012. http://dx.doi.org/10.1111/j.1477-9730.2011.00665.x

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, third edition, 2007. http://dx.doi.org/10.2307/1269484

A. Refice, E. Giachetta, and D. Capolongo. SIGNUM: A Matlab, TIN-based landscape evolution model. *Computers & Geosciences*, volume 45, pages 293–303, 2012. http://dx.doi.org/10.1016/j.cageo.2011.11.013

I. Rey Otero and M. Delbracio. Anatomy of the SIFT Method. *Image Processing On Line*, volume 4, pages 370–396, 2014. http://dx.doi.org/10.5201/ipol.2014.82

A.-M. Rosu, M. Pierrot Deseilligny, A. Delorme, R. Binet, and Y. Klinger. Measurement of ground displacement from optical satellite image correlation using the free open-source software MicMac. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2014. http://dx.doi.org/10.1016/j.isprsjprs.2014.03.002

W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill Education, third edition, 1976. ISBN 007054235X. http://shop.mheducation.com/highered/product.M007054235X.html

D. Ruelland, A. Dezetter, C. Puech, and S. Ardoin-Bardin. Long-term monitoring of land cover changes based on Landsat imagery to improve hydrological modelling in West Africa. *International Journal of Remote Sensing*, volume 29, no. 12, pages 3533–3551, 2008. http://dx.doi.org/10.1080/01431160701758699

D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling. High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth. In *German Conference on Pattern Recognition*, 1, pages 31–42. Springer, Münster, Germany, 2014. http://dx.doi.org/10.1007/978-3-319-11752-2_3

D. Scharstein and R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, volume 47, no. 1/3, pages 7–42, 2002. http://dx.doi.org/10.1023/A:1014573219977

S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06)*, volume 1, pages 519–528. IEEE, 2006. http://dx.doi.org/10.1109/CVPR.2006.19

G. Simpson. Topographic evolution and morphology of surfaces evolving in response to coupled fluvial and hillslope sediment transport. *Journal of Geophysical Research*, volume 108, no. B6, page 2300, 2003. http://dx.doi.org/10.1029/2002JB002162

S. Sinha, D. Scharstein, and R. Szeliski. Efficient High-Resolution Stereo Matching using Local Plane Sweeps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1582–1589. 2014. http://dx.doi.org/10.1109/CVPR.2014.205

N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring Photo Collections in 3D. *ACM Transactions on Graphics*, volume 25, no. 3, pages 835–846, 2006. http://dx.doi.org/10.1145/1141911.1141964

R. Spangenberg, T. Langner, S. Adfeldt, and R. Rojas. Large scale Semi-Global Matching on the CPU. In *IEEE Intelligent Vehicles Symposium Proceedings*, pages 195–201. 2014. http://dx.doi.org/10.1109/IVS.2014.6856419

C. Strecha, W. Von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *IEEE Conference on Computer Vision and Pattern Recognition (2008)*, pages 1–8. 2008. http://dx.doi.org/10.1109/CVPR.2008.4587706

R. Szeliski. *Computer Vision*. Texts in Computer Science. Springer London, 2011. http://dx.doi.org/10.1007/978-1-84882-935-0

R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 30, no. 6, pages 1068–1080, 2008. http://dx.doi.org/10.1109/TPAMI.2007.70844

C. V. Tao and Y. Hu. A comprehensive study of the rational function model for photogrammetric processing. *Photogrammetric Engineering & Remote Sensing*, volume 67, no. 12, pages 1347–1358, 2001. http://eserv.asprs.org/PERS/2001journal/dec/2001_dec_1347-1357.pdf

M. F. Tappen and W. T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, pages 900–906 (volume 2). 2003. http://dx.doi.org/10.1109/ICCV.2003.1238444

T. Toutin. *Analyse mathématique des possibilités cartographiques du satellite SPOT*. Ph.D. thesis, Ecole Nationale des Sciences Géographiques (ENSG), Champs-sur-Marne, France, 1985

T. Toutin. Error tracking in IKONOS geometric processing using a 3D parametric model. *Photogrammetric Engineering & Remote Sensing*, pages 1–32, 2003. http://www.asprs.org/a/publications/pers/2003journal/january/2003_jan_43-51.pdf

T. Toutin. Comparison of stereo-extracted DTM from different high-resolution sensors: SPOT-5, EROS-A, IKONOS-II, and QuickBird. *IEEE Transactions on Geoscience and Remote Sensing*, volume 42, no. 10, pages 2121–2129, 2004a. http://dx.doi.org/10.1109/TGRS.2004.834641

T. Toutin. Review article: Geometric processing of remote sensing images: models, algorithms and methods. *International Journal of Remote Sensing*, volume 25, no. 10, pages 1893–1924, 2004b. http://dx.doi.org/10.1080/0143116031000101611

T. Toutin. Generation of DSMs from SPOT-5 in-track HRS and across-track HRG stereo data using spatiotriangulation and autocalibration. *ISPRS Journal of Photogrammetry and Remote Sensing*, volume 60, no. 3, pages 170–181, 2006a. http://dx.doi.org/10.1016/j.isprsjprs.2006.02.003

T. Toutin. Spatiotriangulation with multisensor HR stereo-images. *IEEE Transactions on Geoscience and Remote Sensing*, volume 44, no. 2, pages 456–462, 2006b. http://dx.doi.org/10.1109/TGRS.2005.861005

G. E. Tucker and R. L. Slingerland. Erosional dynamics, flexural isostasy, and long-lived escarpments: A numerical modeling study. *Journal of Geophysical Research*, volume 99, no. B6, page 12 229, 1994. http://dx.doi.org/10.1029/94JB00320

L. Vandenberghe. The CVXOPT linear and quadratic cone program solvers. Technical report, UCLA, 2010. http://www.seas.ucla.edu/~vandenbe/publications/coneprog.pdf

O. Veksler. Stereo Correspondence by Dynamic Programming on a Tree. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 384–390. 2005. http://dx.doi.org/10.1109/CVPR.2005.334

M. Wang, F. Hu, and J. Li. Epipolar resampling of linear pushbroom satellite imagery by a new epipolarity model. *ISPRS Journal of Photogrammetry and Remote Sensing*, volume 66, no. 3, pages 347–355, 2011. http://dx.doi.org/10.1016/j.isprsjprs.2011.01.002

E. Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal, First Series*, volume 43, pages 355–386, 1937. https://www.jstage.jst.go.jp/article/tmj1911/43/0/43_0_355/_article

T. Westin. Precision Rectification of SPOT Imagery. *Photogrammetric Engineering & Remote Sensing*, volume 56, no. 2, pages 247–253, 1990. http://eserv.asprs.org/PERS/1990journal/feb/1990_feb_247-253.pdf

G. Willgoose, R. L. Bras, and I. Rodriguez-Iturbe. A coupled channel network growth and hillslope evolution model: 2. Nondimensionalization and applications. *Water Resources Research*, volume 27, no. 7, pages 1685–1696, 1991. http://dx.doi.org/10.1029/91WR00936

J. Wohlfeil, H. Hirschmüller, B. Piltz, A. Börner, and M. Suppa. Fully automated generation of accurate digital surface models with sub-meter resolution from satellite imagery. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XXXIX-B3, no. September, pages 75–80, 2012. http://dx.doi.org/10.5194/isprsarchives-XXXIX-B3-75-2012

H. Yésou, A. Escudier, S. Battiston, J.-Y. Dardillac, S. Clandillon, C. Uribe, M. Caspard, H. Giraud, J. Maxant, A. Durand, K. Fellah, M. Studer, C. Huber, L. Philippoteaux, P. de Fraipont, and D. Fontannaz. Exploitation de l'imagerie Pléiades en cartographie réactive suite à des catastrophes naturelles ayant affecté le territoire français en 2013. *Revue Française de Photogrammétrie et de Télédétection*, volume 209, pages 39–45, 2015. http://www.sfpt.fr/rfpt/index.php/RFPT/article/view/210

K. J. Yoon and I. S. Kweon. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 28, no. 4, pages 650–656, 2006. http://dx.doi.org/10.1109/TPAMI.2006.70

R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proceedings of the European Conference on Computer Vision (ECCV '94)*, volume 2, pages 151–158. Springer-Verlag, Berlin/Heidelberg, 1994. http://dx.doi.org/10.1007/BFb0028345

K. Zhu, P. D'Angelo, and M. Butenuth. Evaluation of stereo matching costs on close range, aerial and satellite images. In *Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods*, pages 379–385. SciTePress, 2012. http://dx.doi.org/10.5220/0003764203790385