ISSN 1688-2784

Universidad de la República
Facultad de Ingeniería

Challenges of multi-view satellite stereo reconstruction pipelines and some contributions on key stages

Tesis presentada a la Facultad de Ingeniería de la Universidad de la República por

Alvaro Gómez

en cumplimiento parcial de los requerimientos
para la obtención del título de
Doctor en Ingeniería Eléctrica.

Directores de Tesis

Gregory Randall ..... Universidad de la República
Rafael Grompone ..... École Normale Supérieure Paris-Saclay
Gabriele Facciolo ..... École Normale Supérieure Paris-Saclay

Tribunal

Pablo Musé ..... Universidad de la República
Jean-Michel Morel ..... École Normale Supérieure Paris-Saclay
Matías Di Martino (Revisor) ..... Duke University
Enric Meinhardt (Revisor) .. École Normale Supérieure Paris-Saclay

Director Académico

Gregory Randall ..... Universidad de la República

Montevideo
Febrero 2023

Challenges of multi-view satellite stereo reconstruction pipelines and some contributions on key stages, Alvaro Gómez.

ISSN 1688-2784

Esta tesis fue preparada en LaTeX usando la clase iietesis (v1.1).
Contiene un total de 161 páginas.
Compilada el martes 18 abril, 2023.
http://iie.fing.edu.uy/

Agradecimientos

A Gregory, Rafael y Gabriele por el apoyo y la paciencia.
A Sandra, Lucía y Martín por estar bien cerca y por soportar este proceso.

This page intentionally left blank.

Resumen

Los satélites que toman imágenes de la Tierra son cada vez más numerosos, produciendo imágenes diarias de todos los puntos del globo, tanto gratuitas como de pago. En esta tesis nos concentramos en la reconstrucción de superficies a partir de imágenes de satélite de luz visible a través de estereovisión. Dadas dos imágenes de una escena desde diferentes puntos de vista conocidos, el objetivo del estéreo es estimar la forma o profundidad 3D más probable que explica esas imágenes. Cuando hay más de dos imágenes disponibles, se puede aplicar el estéreo multivista (MVS) trabajando por pares e integrando las reconstrucciones (MVS por pares) o derivando una reconstrucción de todas las imágenes a la vez (MVS "real"). En el caso de las imágenes de satélite, el MVS se ha realizado tradicionalmente con enfoques por pares, en los que las múltiples vistas se tratan por pares realizando estéreo tradicional de dos vistas y luego fusionando los modelos digitales de superficie (DSM) de las reconstrucciones por pares para obtener el resultado final. Varias soluciones comerciales y de código abierto bien establecidas organizan sus pipelines de trabajo de este modo. Estas soluciones se basan principalmente en algoritmos de estereo clásicos, mientras que las alternativas de aprendizaje profundo (AP) se están adaptando poco a poco para funcionar en los pipelines. Pero los resultados de los métodos basados en AP no han superado claramente a los de los pipelines tradicionales y queda mucho por hacer en este campo aún abierto. Una cuestión crucial que complica el avance en este campo es la escasez de conjuntos de datos públicos con altura conocida.

En la tesis se evaluaron y compararon un conjunto de métodos de diferentes enfoques de MVS por pares y real. Para la comparación, se adaptaron métodos clásicos y de aprendizaje profundo para trabajar con imágenes de satélite y para interactuar correctamente con S2P, un pipeline modular de estereo satelital. Los resultados obtenidos con los métodos de aprendizaje profundo mostraron el potencial del uso de este tipo de algoritmos en imágenes de satélite como un paso en un pipeline estéreo clásico o como una solución MVS de extremo a extremo.

Si se considera el MVS por pares, además del matching estéreo, hay otros dos pasos cruciales para lograr una buena reconstrucción: (a) la selección de los pares más apropiados, y (b) la fusión de los DSMs reconstruidos a partir de los pares. Para la selección de pares, se concibió una estrategia novedosa basada en la simulación de imágenes de satélite que puede ordenar los pares de forma más consistente que las heurísticas utilizadas habitualmente. Para la simulación de imágenes, se desarrolló una herramienta que puede generar vistas a partir de una escena 3D artificial. En cuanto a la fusión de DSMs, se desarrolló un esquema iterativo basado en el filtrado bilateral que demostró ser un método robusto. Las mejoras en otras etapas del pipeline estéreo satelital y el procesamiento de nubes de puntos también formaron parte de los temas abordados durante la tesis.

This page intentionally left blank.

Abstract

Satellite imagery is quickly gaining in importance, with Earth observation satellites producing daily images from all the points of the globe, both commercially and freely available. In this thesis we concentrate on surface reconstruction from visible light satellite images through stereo-vision. Given two images of a scene from different known viewpoints, the objective of stereo is to estimate the most likely 3D shape or depth that explains those images. When more than two images are available, multi-view stereo (MVS) can be applied working by pairs and integrating the reconstructions (pair-wise MVS) or deriving a reconstruction from all the images at a time (true MVS). In the case of satellite images, MVS has traditionally been performed with pair-wise approaches where the multiple views are treated by pairs doing traditional two-view stereo and then aggregating the digital surface models (DSM) from the pair-wise reconstructions to get the final result. Several well established commercial and open-source solutions organize their working pipelines in this way. This solutions mostly rely on classic stereo algorithms while deep learning (DL) alternatives are slowly being adapted to work in the pipelines. But the DL based approaches have not still clearly outperformed the traditional pipelines and there is room for much more work in this yet open area. A crucial issue that complicates the advance in this field is the scarce public datasets with well curated ground-truth.

In this thesis a set of methods from different approaches of pair-wise and true MVS were evaluated and compared. For the comparison, classic and deep learning methods were adapted to work with satellite images and to correctly interface with S2P, a modular satellite stereo pipeline. The results obtained with deep learning methods showed the potential of using this kind of algorithms on satellite images as a step in a classic pipeline or as an end-to-end MVS solution.

Considering pair-wise MVS, besides the stereo matching, two other steps are crucial to achieve a good reconstruction: (a) the selection of the most appropriate pairs, and (b) the fusion of the DSMs reconstructed from the pairs. For pair selection, a novel strategy based on the simulation of satellite images was devised and can order the pairs in a more consistent way than commonly used heuristics. For the simulation of images, a tool that can generate views from an artificial 3D scene was developed. Regarding the fusion of DSMs, an iterative scheme based on the bilateral filtering was conceived showing to be a robust and performant method. Improvements in other stages of the baseline stereo pipeline and the processing and analysis of point clouds were also part of the topics addressed during the thesis.

This page intentionally left blank.

Table of contents

This page intentionally left blank.

Chapter 1

Overview of the thesis

This chapter presents an overview of the thesis with an abridged version of the subjects that are exposed on the following chapters.

1.1. Satellite imagery

The number of satellites that observe the earth is continuously growing since the 1950s. The first satellites were intended for military purposes but in the 1970s the commercial earth observation began with the launch of Landsat-1 by NASA in 1972.

Stereo reconstruction was still not possible at that time. In order to reconstruct a 3D shape by stereo vision, images from different viewpoints of a scene are necessary. Back in the 1970s, satellites carried 2D imaging sensors and could only take nadir images.

In 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. It was also the first civilian satellite to provide off-nadir viewing capabilities, enabling the acquisition of stereo images.

Following SPOT-1, other satellites for stereo acquisition were constructed and launched. They are classified in [66] as: (1) standard across-track systems, (2) standard simultaneous multi-view along-track systems, and (3) agile single-lens systems.

In the first case, the CCD lines and one optical system are generally combined with a mirror or similar mechanism that rotates from one side of the sensor to the other across the flight direction. With this configuration, the stereo-images are collected from different orbits at different dates, with the overlapping area across the flight direction. Some examples of this class are the SPOT 1-4, the IRS-1C/1D from India and the Kompsat-1 from Korea.

In the second case, two or more strips are taken simultaneously from the same orbit at different angles along the flight direction. For each viewing direction, there is one lens and one set of CCD lines placed on the focal plane. Examples of this class are the ALOS-PRISM from Japan and Cartosat-1 from India.

The third is the most recent class of satellites which scan the Earth flying along sun-synchronous and quasi-polar orbits. They have the ability to rotate about the axes of their sensor in order to point at off-nadir targets up to 30°, or to view the target from different directions during a single orbit. Two subclasses can be distinguished within this class. Some agile sensors have a synchronous acquisition mode, thus the satellite speed and the scanning speed are equal, and the viewing angle is constant during one image acquisition. Other agile sensors can perform rapid movements and scan the earth in asynchronous mode where the scanning speed can be faster than the satellite speed. IKONOS-2 from USA, launched in 1999, was the first satellite with an agile system that could work in synchronous acquisition mode. Other examples of this kind are Kompsat-2 from Korea and Formosat-2 from Taiwan. Examples of rapid agile systems working in asynchronous acquisition mode are the WorldView 1-3 from USA and Pleiades 1-2 from France.

Satellite systems are evolving from the single sensor model to the cooperative

approach working in constellations [81], allowing for shorter revisit times. While single satellites have a revisit time of one or more days, a constellation such as the SkySat can observe any point in Earth several times per day.

Spatial resolution of the images acquired by satellites (known as Ground Sampling Distance or GSD) is progressively getting better with the new generations of sensors. While SPOT resolution was of 10 meters, several new satellites are acquiring with GSD below the meter like WorldView, QuickBird and Pléiades. In particular, WorldView3 has reached a GSD of 0.3m and this trend will continue in the future as new sensors are developed.

The enhanced GSD, the short revisit times and the stereo capability of the current constellations of satellites enables the generation of high quality data to perform Multi-View Stereo and reconstruct 3D models of any desired location in Earth.

1.1.1. Example satellite images

Figure 1.2 shows some images acquired by the WorldView-3 satellite from the city of Buenos Aires. This images are part of the Multiple View Stereo Benchmark for Satellite Imagery (MVS3D) public dataset [10]. The region of the city covered by the images, an altitude map and a plot of the orientation of the views can be seen in Figure 1.1.

Figure 1.4 shows some images acquired by the WorldView-3 satellite from the city of Jacksonville (JAX). This images are part of the US3D [9] public dataset. Figure 1.3 shows the region of the city, an altitude map and a plot of the orientation of the views.

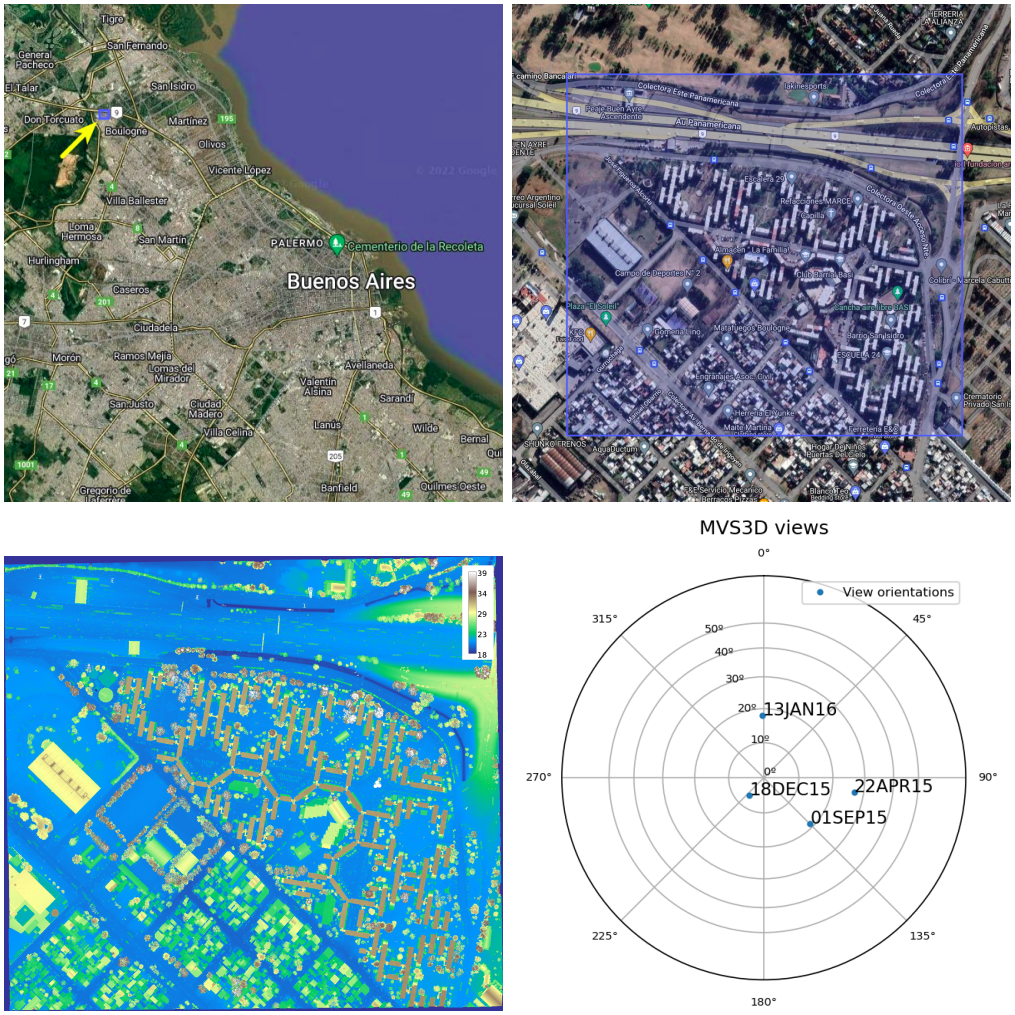

Figure 1.1: Region of the city of Buenos Aires and orientations of the views corresponding to the images in Figure 1.2. Top: region of the city. Bottom left: altitude map of the region from an airborne Lidar. Bottom right: orientation of the views.

Figure 1.2: Images of Buenos Aires city from the MVS3D public dataset.

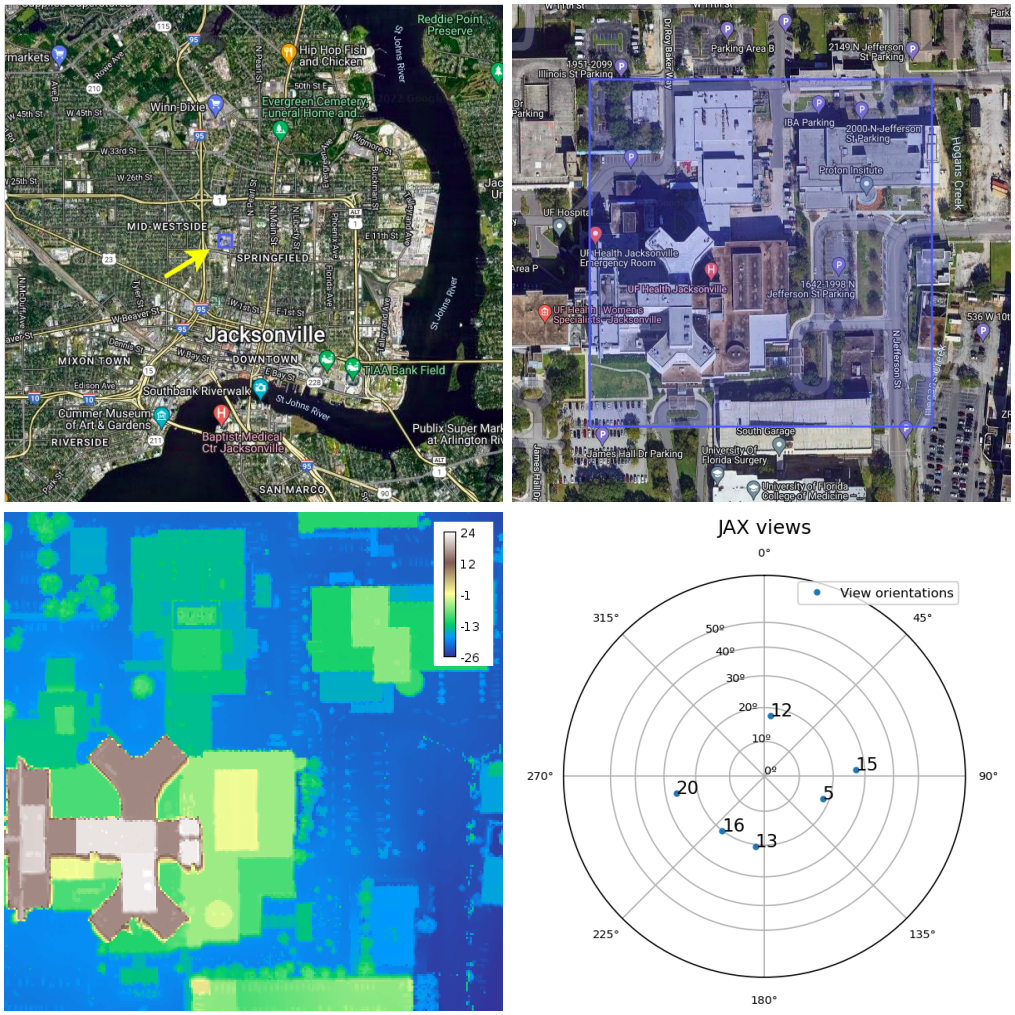

Figure 1.3: Region of the city of Jacksonville and orientations of the views corresponding to the images in Figure 1.4. Top: region of the city. Bottom left: altitude map of the region from an airborne Lidar. Bottom right: orientation of the views.

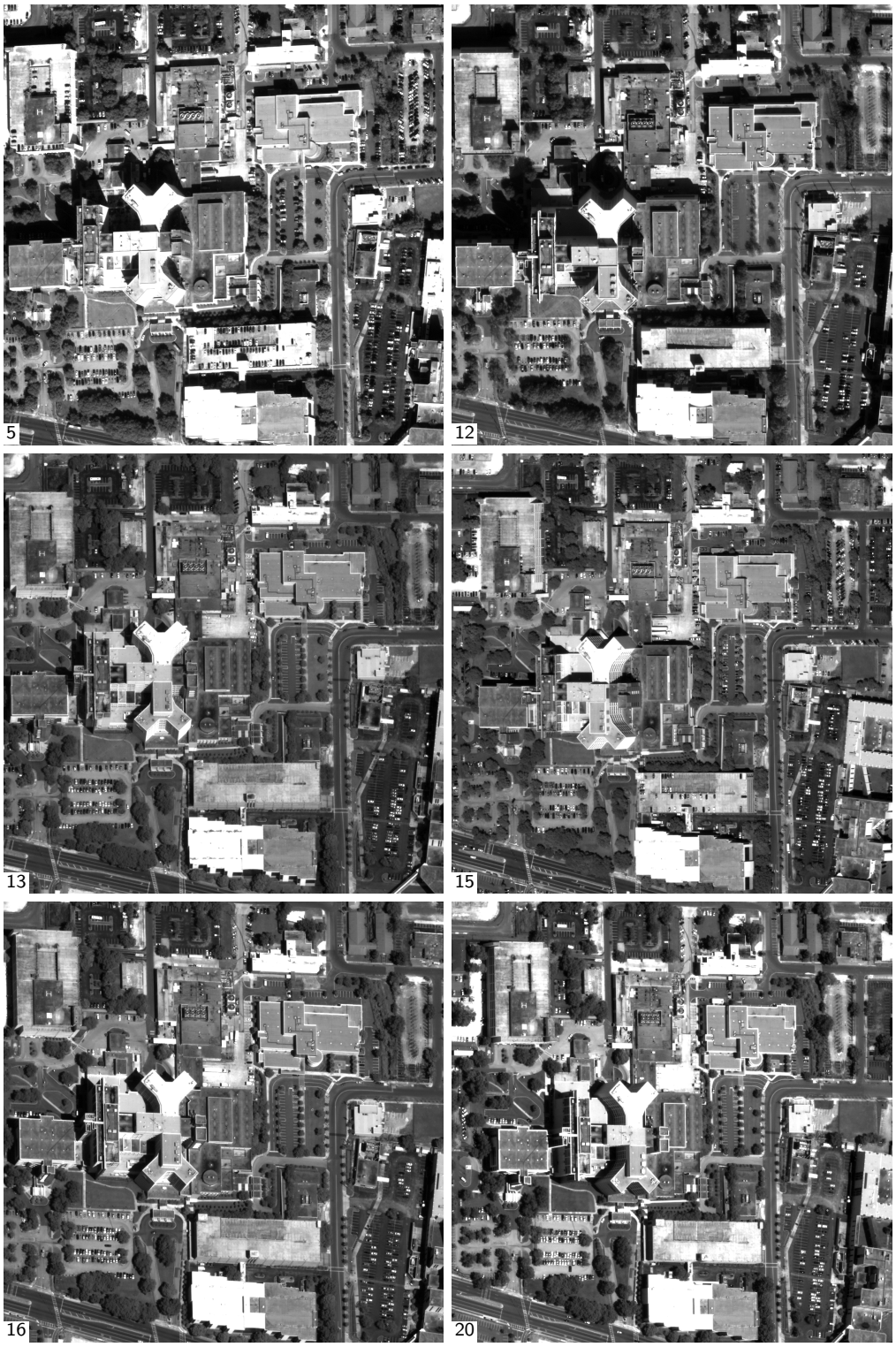

Figure 1.4: Images from Jacksonville city from the US3D public dataset. Views 5, 12, 13, 15, 16 and 20 from the region 68 of the city in the US3D dataset.

1.2. Multi-view stereo

In multi-view stereo (MVS), given a set of images of an object or a scene, the objective is to estimate the most likely 3D shape that explains those images [31]. The main cue exploited in MVS as in binocular stereovision is the photometric consistency. The appearance of the scene in corresponding regions across the images is expected to be similar. The image values in corresponding regions or patches should be similar and/or have a similar structure.

Stereo vision is an area that has been extensively researched and multiple algorithms have been proposed along the last decades [40, 49, 75]. Initial methods worked on one stereo pair. Then, Multi-View Stereo (MVS) was first approached as an extension of the stereo algorithms by aggregating the information of multiple stereo pairs. True MVS algorithms considering directly all the images of the scene arrived some time afterwards [13]. True MVS algorithms were mostly devised, by the computer vision community, for the reconstruction of objects, buildings and interiors with images taken with standard pinhole-like cameras at close distance [31, 77]. Deep Learning (DL) MVS methods [49] flourished in the last years and have taken over the top rankings of the main benchmarks of the area [46, 78] but classic methods are still a valid option when dealing with many and/or large size images since DL methods struggle to accommodate large 3D structures in GPU memories.

In the case of satellite images, MVS has traditionally been performed with pair-wise approaches where the multiple views are treated by pairs doing traditional two-view stereo and then aggregating the pair-wise reconstructions (elevation models or point clouds, for example) to get the final result [19, 47, 60].

It was only recently in 2019 that Zhang et al. [89] showed that classic true MVS algorithms used in computer vision could be adapted to satellite images for the benefit of the remote sensing field. In the last three years, there has been an important thrust in the research for the adaptation of true MVS to satellite imagery and the use of DL methods as part of pair-wise stereo pipelines or as end-to-end MVS solutions [11,33,39,57]. But results of this impulse have not still clearly outperformed the traditional pipelines and there is room for much more work in classic and DL methods in this yet open area. A crucial issue that complicates the advance in this field is the scarce public datasets with well curated ground-truth.

The present thesis is framed in this active point in the field and seeks to analyze different stages of satellite MVS in the light of recent techniques, proposing improvements to some of its stages. Chapter 2 contains a brief introduction to some concepts related to multi-view stereo. The next sections in the current chapter present an overview of the methods and experiments addressed throughout the thesis.

1.3. Pair-wise and true multi-view stereo

A set of methods from different approaches of pair-wise and true multi-view stereo were evaluated and compared. For the comparison, the methods were adap-

ted to work with satellite images and to correctly interface with the S2P pipeline. We built upon the work in [89] and extended the concept to include stereo and MVS methods based on DL with interesting results.

The methods used for comparison are shown in Table 1.1. They are representative of different approaches that can be applied to satellite images in order to derive an altitude map from a set of images.

Table 1.1: Tested methods

Method	Type	DL	Notes
S2P [19]	Pair-wise	No	MGM [28] in disparity computation
S2P-GANet	Pair-wise	Partially	GANet [88] in disparity computation
COLMAP [76, 77]	Multi-view	No	Adapted for satellite images by [89]
CasMVSNet [38]	Multi-view	Yes	Adapted for satellite images in this work

The methods were applied to the datasets following the scheme depicted in Figure 1.5. For pair-wise methods, a Digital Surface Model (DSM) is computed for every possible pair of images of a region; these DSMs are then aggregated to get an enhanced multi-pair DSM. On the other hand, true multi-view methods are fed with all the images of a region.

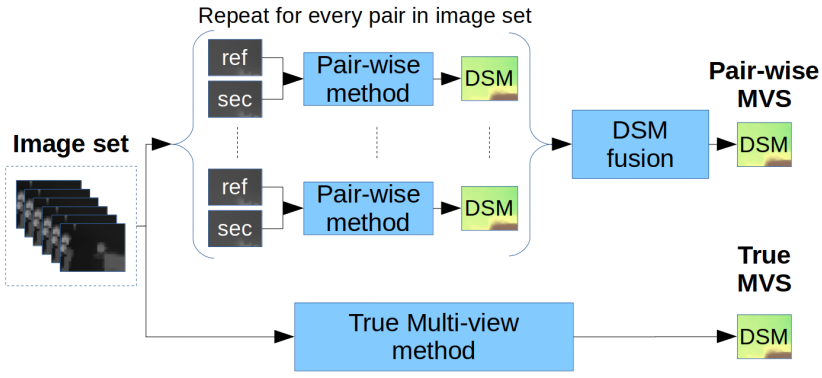

Figure 1.5: Scheme of multi-pair and true multi-view methods.

In order to assess the performance of the different approaches, the computed DSMs are compared against the ground-truth DSM on all the datasets. The comparison is done as shown in Figure 1.6.

Most satellite pipelines in use are based on pair-wise approaches with classic methods that are known to achieve accurate results. Our study confirms this fact

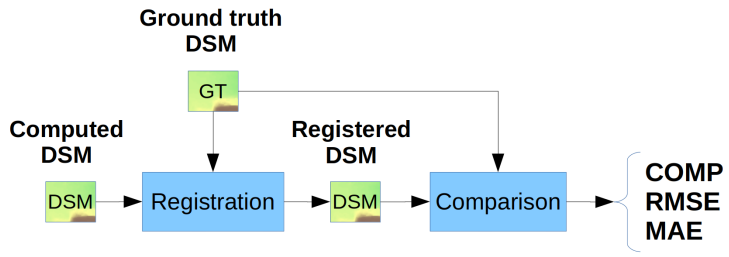

Figure 1.6: Flow diagram for the comparison of computed DSM with respect to the ground truth altitude.

showing that it is hard to beat the baseline pipeline. On the other hand, results also expose that other valuable methods from the computer vision field can be adapted to work on satellite images since they get results comparable and in some cases slightly better than the baseline, even if they have not been trained on satellite images.

Stereo methods can be adapted to work as a stage of an existing satellite stereo pipeline. In this work we adapted and tested the GANet method as an alternative stereo matching step in the S2P pipeline. An interesting finding is that the results for the S2P-GANet variant were similar and in some cases better than the S2P baseline pipeline without any specific training. The fine-tuning of GANet in more appropriate datasets, such as WHU and RVL, showed a slight but consistent improvement in mean on all datasets used in the experiments.

Regarding true MVS methods, Zhang et al. proposed in [89] an adaptation strategy and implemented it for classic methods such as COLMAP and Plane-sweep. The approach, as reported in their article, does not achieve better results than pair-wise MVS based on S2P. Nevertheless, the approach is very interesting and can be applied to other methods, not intended initially for satellite imagery. We use that strategy to adapt the CasMVSNet method in this work.

In summary, a deep learning (DL) stereo matcher (GANet) was adapted and integrated into the S2P pipeline. This enabled to compare the end-to-end reconstruction performance of this matcher against MGM, its classic counterpart currently in use in the pipeline. An interesting result of this adaptation was that the DL matcher achieved a similar performance to the current classic matcher without any specific training on satellite images, presenting a great generalization power. This showed that the GANet algorithm and eventually other DL stereo algorithms could be a valid alternative to be used in the pipeline as the stereo matching step. A fine-tuning of GANet yielded slightly better results in completeness and in accuracy. Regarding MVS, a state-of-the-art method as CasMVSNet was adapted to work on satellite images. The results were encouraging and showed the potential of using DL-based algorithms in satellite stereo pipelines. The comparison of methods is addressed in Chapter 3.

1.4. Disparity map post-processing

A method that filters the disparity map guided by the images of the pair was analyzed and tested. The method is based on [26] and we used, for the experiments, a code developed by Sébastien Drouyer. For the experiments, the disparity refinement was inserted in the S2P pipeline as an additional step that modifies the disparity map before the triangulation step is performed.

The disparity map resulting from a stereo matching algorithm may have missing values associated to low matching in regions lacking good texture cues. Other artifacts may also occur due to repetitive texture giving place to small regions with disparity values far from the correct values.

Several algorithms for the post-processing of depth and/or disparity maps have been proposed in the literature. The survey in [4] classifies the approaches in: a) Filtering, interpolation, extrapolation methods, b) Methods based in reconstruction, and c) Inpainting based methods. According to that classification, the tested algorithm can be considered a filtering of the disparity map guided by the images of the stereo pair.

Consider a rectified stereo pair and a left-to-right (and right-to-left if available) disparity map computed by any stereo matching algorithm. The objective is to fill-in two kind of pixels in the disparity map:

- the pixels where the disparity is undefined;
- the pixels belonging to small regions with disparity values far away from their surrounding disparities (we will call these regions "speckles").

Missing disparities are usually due to occlusions and regions with poor texture cues in the stereo pair. Speckles can be originated by incorrect matching of repetitive structures and also by poor texture in the images.

The tested algorithm is organized in two main parts. The first part comprises the identification of speckles. In the second part, the undefined disparity pixels are filled-in by a diffusion process. In this process, guided by the reference image of the stereo pair, pixels in undefined disparity regions are filled with a weighted average of the disparities of neighboring regions.

In satellite imagery, the disparity range of a stereo pair is dependent on the position and attitude of the satellite in each acquisition. This makes difficult the comparison in disparity error across different stereo pairs. The comparison on altitude can give a better idea of the performance of the algorithm in these kind of images. In the experiments, the altitudes triangulated from the baseline and from the processed disparity maps were compared against the altitude data from the ground-truth airborne Lidar on images of the MVS3D dataset. Results as the ones depicted in Figure 1.7 were obtained.

The results for the tests show that the diffusion of the disparity map guided by the images of the stereo pair is a plausible filling strategy. It enhances the definition of structures that have contrasted values in the images of the stereo pair. However, the completeness of the reconstruction does not necessarily raise

with the diffusion and this depends on the initial condition of the disparity map. Anyway, the algorithm seems an interesting option to be applied in controlled conditions.

An IPOL demo of the method was implemented and can be tried at:
`https://ipolcore.ipol.im/demo/clientApp/demo.html?id=77777000089`

A complete description of the method and the experiments are presented in Chapter 4.

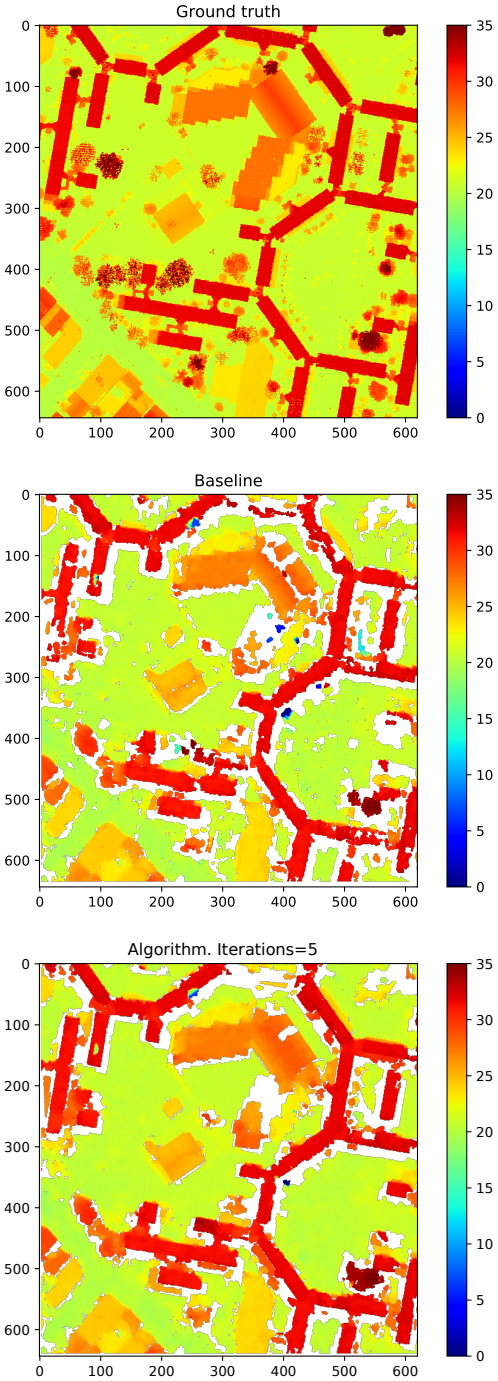

Figure 1.7: The ground truth altitude and altitudes computed from the baseline disparity map and for the processed disparity map. Pixels in white are the ones that do not have information and are filled-in by the diffusion process. Boundaries of buildings get better defined after five iterations of the algorithm.

1.5. DSM fusion

In pair-wise MVS, the DSMs reconstructed from several stereo pairs need to be aggregated to obtain a final DSM. In this work an iterative approach based on the bilateral filter [80] is proposed for the fusion of the DSMs. The bilateral filter framework allows to robustly integrate the spatial information along with other available sources of information that can regularize the final integrated DSM. Typically, the framework can integrate not only the height of the DSMs and the gray level or color of a reference image, but also other features such as a semantic segmentation or confidence maps, if available. The iterative scheme with shrinking height ranges allows to gradually refine the solution, taking into account, in each step, height samples that are closer to the previous estimation.

The most common way of integrating a set of DSMs is to apply a per-pixel median of the heights in the set of DSMs. This usually yields a robust estimation and removes most outliers in the DSMs. However, this pixel-wise approach does not introduce spatial coherence.

Figure 1.8 illustrates the performance change on one region of the dataset when the proposed fusion is introduced in the pipeline. The graphs show the behavior of the two metrics—completeness and median of the absolute error—for the region as the number of integrated DSMs is increased according to different ordering criteria. The proposed fusion method outperforms the per-pixel median and the completeness of the reconstruction does not degrade as more DSMs with decreasing quality are included in the fusion.

In summary, the method proposed in this thesis fuses the DSMs following an iterative approach based on the bilateral filtering. Contrary to the most commonly used per-pixel median, the approach allows to better integrate DSMs considering the spatial coherence of different properties of the data such as height and gray level. The method produces a spatial regularization effect without affecting the borders of the structures and outperforms the classic per-pixel median. Another positive attribute is that the bilateral filter framework is applied iterativelly, allowing to gradually refine the solution. Using progressively more restrictive ranges for the height allows to focus on the height samples that are close to the previous estimation and are then probably more accurate. This makes the reconstruction robust, tolerating the potential inclusion of DSMs from bad pairs without noticeable degradation in the reconstruction performance. The method is presented as part of Chapter 6.

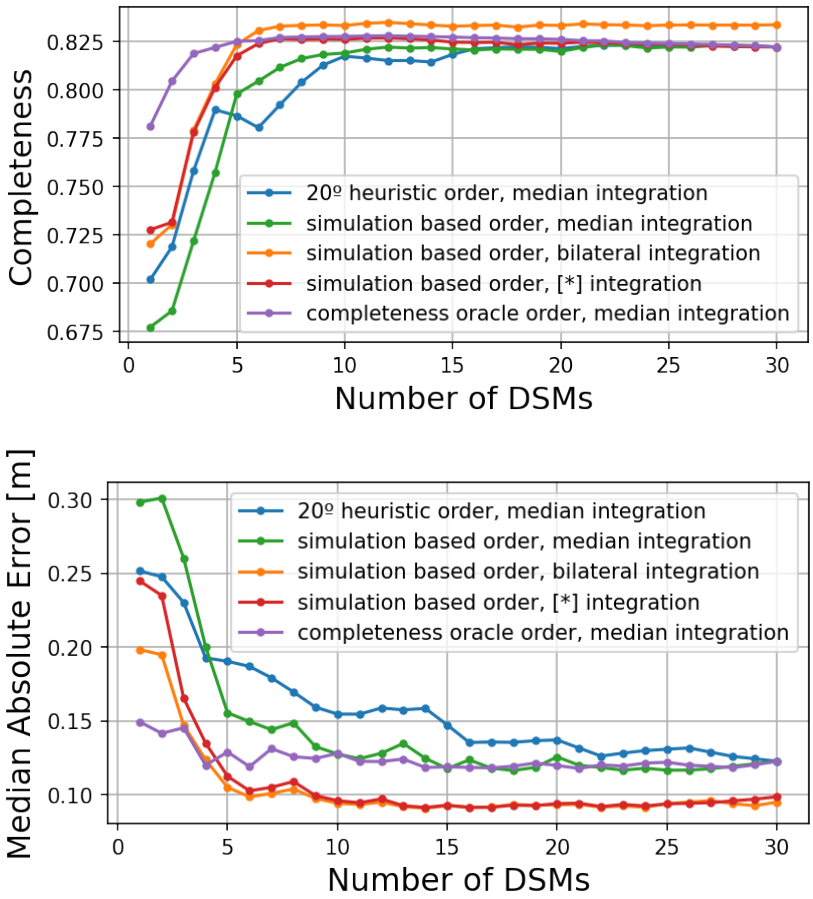

Figure 1.8: Progressive integration of DSMs for region JAX_156. Completeness (top) and MAE (bottom) evolution when integrating a growing number of DSMs. The purple curve corresponds to an integration by the median and an oracle ordering. The curve in orange corresponds to the integration with the iterative bilateral filtering. The red curve (marked with [*]) corresponds to an implementation of the method in [70]. Note how the proposed method (in orange) outperforms the others and the completeness of the reconstruction does not degrade as more DSMs with decreasing quality are included in the fusion.

1.6. Simulation of image and RPC

A simulation tool was implemented that allows to produce views of an artificial 3D scene generating images and RPC models suitable to be used in a satellite pipeline. The tool uses an affine camera model that is a sensible approximation of a real satellite projection for a small area of interest [20]. The generated images and RPCs can be used to train or test algorithms for satellite images.

In this thesis, the tool was used to devise a novel pair selection strategy, a problem that has been traditionally tackled with heuristics. The simulator tool allows to draw any pair of views in the hemisphere surrounding a 3D scene that can then be reconstructed with a satellite stereo pipeline. This enables to study

and map the relation between the orientation of the views and the quality of the 3D reconstruction of a pair. The map can then be used for the ordering of real pairs in a more consistent way than with heuristics.

Public satellite datasets with ground truth altitude are scant and this is a problem for the development of algorithms by the community. The simulation tool could also be used to complement the few available real datasets.

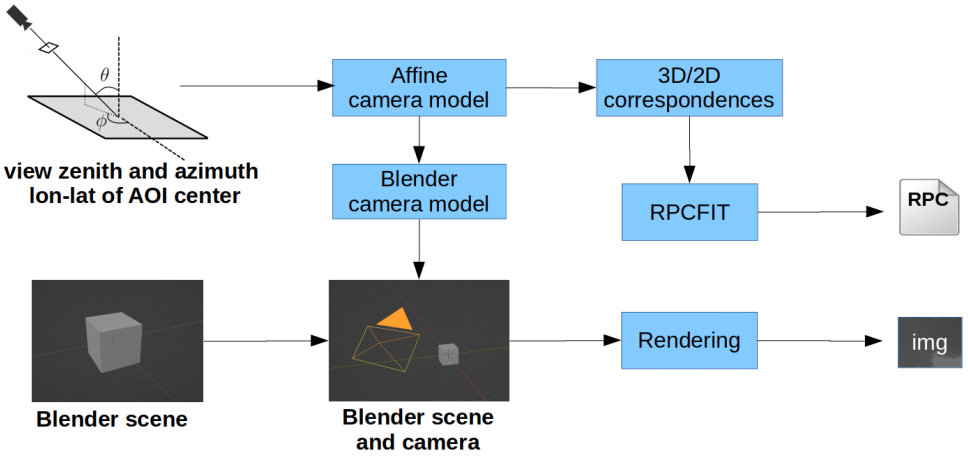

Figure 1.9: Block diagram of the simulator. Please refer to the text for the description of the blocks and the flow of data.

Figure 1.9 presents a block diagram of the simulation tool. Given a scene and a view direction, an affine camera model is determined. The affine camera model is a sensible approximation of a real satellite projection for a small area of interest (AOI) [20]. This model gives corresponding 3D/2D coordinates between the volume of interest (AOI plus height range) and the image. The correspondences are then used to adjust an RPC camera model using the RPCFIT tool [2], which fits an RPC model to the 3D/2D correspondences through a regularized least squares minimization. The simulator uses Blender [15] as the 3D engine to render the views. Blender is launched and configured automatically through Python scripts.

The tool can also generate configurations to run a stereo reconstruction with S2P on a simulated pair. Figure 1.10 shows an example of a stereo pair generated with the tool and the DSM reconstructed with S2P.

The simulation tool is available for public use at:
`https://github.com/zemogoravla/simsatool`.

In this thesis it was used for the development of a pair selection strategy but it could be also useful test or train other algorithms for stereo and multi-view stereo in satellite images. The details of the tool are presented in Chapter 5. The use of the tool for pair selection is part of Chapter 6.

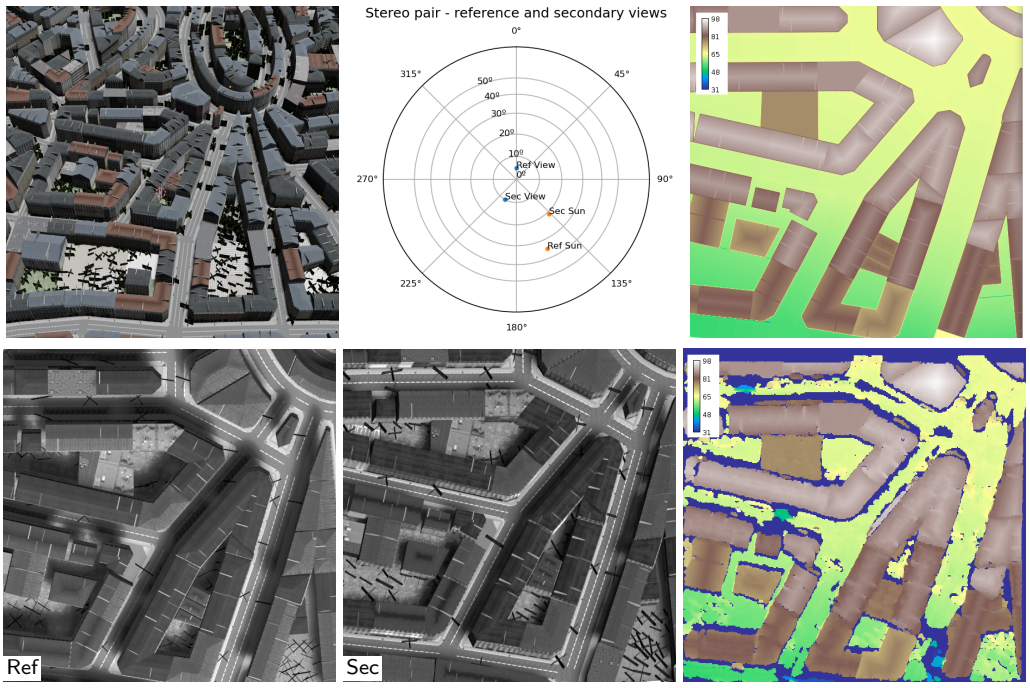

Figure 1.10: Generation of an image pair for the stereo reconstruction with the S2P pipeline. Top: the 3D scene, the orientations of views and sun in the stereo pair, and the ground truth DSM. Bottom: Generated reference and secondary images of the pair and the DSM reconstructed with the S2P pipeline.

1.7. Pair selection

In pair-wise MVS, it is well known that the fusion of DSMs computed from multiple pairs improves in general the completeness [29, 64] of the reconstruction. However, if bad pairs are included in the fusion, the result may get degraded. The selection of the most appropriate pairs has been traditionally tackled by designing heuristics based on the metadata of the pairs.

An alternative selection method based on the simulation of images was devised and evaluated. The method maps the relation between the orientation of the views and the reconstruction quality through simulation. The simulation tool described in 1.6 allows to produce views of a 3D scene from multiple orientations. The stereo reconstruction from a pair of simulated images can be assessed by comparing to the known altitude of the scene. This enables to compute a map that encodes the reconstruction quality in relation to the orientation of any pair of views sampled from the hemisphere surrounding the scene. The map acts as a proxy for the quality of real pairs and can be used to sort the pairs in a more consistent way than the traditional heuristics.

In pair-wise MVS, given a set of N images taken from a scene, $N(N-1)$ ordered pairs can be considered for stereo reconstruction. For each pair, a Digital

Surface Model (DSM) of the scene is determined. The final MVS reconstruction of the scene can then be obtained by the integration of all the computed DSMs. The quality of the final reconstruction is determined by the quality of the pair-wise DSMs, which depend on several factors such as the orientations of the views of a pair and changes in the acquisition conditions between the images, among others. It is hard to identify all of the factors and tell their relative importance. This difficult task has been traditionally tackled by designing heuristics that take into account the metadata of the images [17, 29].

With the aid of the simulation tool, we sampled the hemisphere over a 3D scene and generated stereo pairs from multiple orientations. The DSMs computed from the sampled pairs were compared against the known altitude of the scene to assess the reconstruction performance. Figure 1.11 shows the reconstruction errors of simulated pairs for different reference-secondary orientations in the hemisphere.

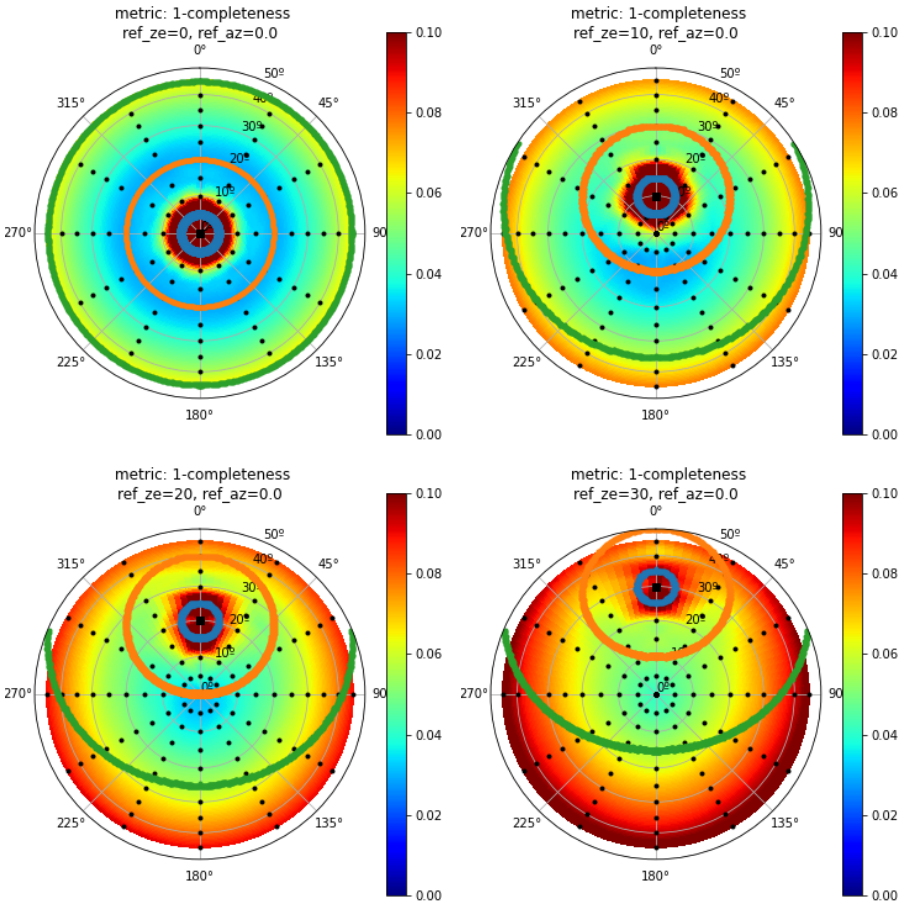

Figure 1.11: Reconstruction errors of simulations for different reference-secondary image orientations. The square represents the reference view and circular points represent the tested secondary views. Metric 1-COMP is shown for increasing zenith angle of the reference view. Blue corresponds to small errors while red indicates large errors. The blue, orange and green curves indicate the positions in the hemisphere for views 5º, 20º and 45º apart from the reference, respectively.

Given a set of N real satellite images, there are $N \times (N-1)$ possible ordered pairs. For each candidate pair, we can estimate the reconstruction error (as $1 -$ COMP) by querying the orientations of the real images in the pre-computed error maps of Figure 1.11. This provides an ordering for the integration of the DSMs reconstructed from the pairs. This ordering based on the completeness obtained from the simulation acts as a proxy for the true completeness, which cannot be computed in a real scene where the ground-truth is not available.

The study of the selection criteria led to the evaluation of different performance metrics for the stereo reconstruction. The traditional performance metrics used in satellite images are the ones proposed in [10] (COMP, RMSE, MAE). Usually, a single threshold is used to evaluate the completeness that achieves a method or a pipeline [10, 19]. Instead of evaluating the COMP with a single threshold or a set of few thresholds, it is possible to tests it in a more dense set of thresholds. Figure 1.12 shows the COMP achieved for the stereo reconstruction of different pairs of images of the JAX dataset. In that figure, the COMP is calculated for a set of thresholds. The curves of COMP as a function of the threshold have different evolutions for the set of reconstructions. The curves hold valuable information of the accuracy of the reconstruction; this information is lost when one considers only discrete thresholds. We can consider the Area Under the Completeness Curve (AUCC) as a single number that summarizes the information of completeness and accuracy of the reconstruction. The tests on pair selection of real images showed that using the AUCC instead of the COMP of the corresponding simulation pairs led to a better ordering of the stereo pairs.

In summary, an alternative selection method based on the simulation of images was developed and is proposed as a more consistent strategy than the traditional heuristics. A comprehensive metric (AUCC) that carries the joint information of completeness and accuracy was evaluated and results indicate that is worth considering for the task of pair selection. The development of these topics is carried out in Chapter 6.

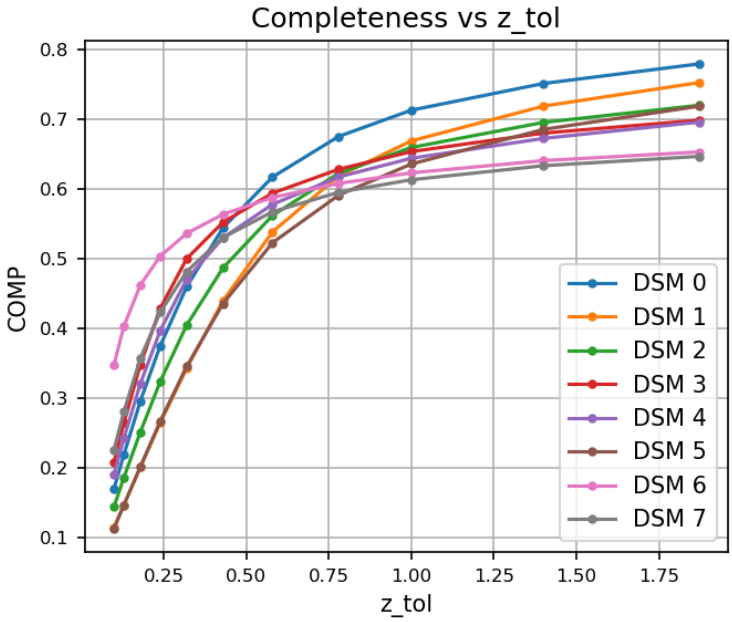

Figure 1.12: Completeness metric as a function of the altitude tolerance z_tol. Pixels are considered correct if their altitude difference to the ground-truth is less than z_tol. Although several of the DSMs have a similar completeness at 1m, the completeness curves show that certain reconstructions are more accurate than others.

1.8. Point cloud analysis

Not directly connected to stereo pipelines, other works on point clouds were done during the thesis. Based on the *a contrario* methodology, Lezama et al. proposed in [51] a point alignment detector. This method, originally in 2D was extended to 3D.

The *a contrario* methodology proposed by Desolneux, Moisan and Morel [22,23] is a mathematical formalization of the *non-accidentalness principle* proposed for perception [3,85,86]. In this approach, an observed structure is considered relevant if it rarely occurs by chance. This is implemented assuming a null-hypothesis H_0 for the data where no detections should occur (the a contrario model). The rarity or non-accidentalness of a structure is quantified as the probability of observing that structure under the H_0 hypothesis.

For the detection of point alignments, given a data set of 3D points, a candidate alignment consists of a thin cylinder in space defined by two points of the data set. The idea is to evaluate if the point density inside the cylinder is significantly high with respect to the local background. A meaningful alignment should also have regularly spaced points inside the candidate cylinder. The local point density is evaluated in the local cylinder surrounding the candidate cylinder. The candidate alignment cylinder is divided into pill-boxes and will be validated if the number of

occupied boxes is statistically significant given the local point density. Figure 1.13 shows a schematic representation of the candidate and local cylinders determined by a pair of points in the domain.

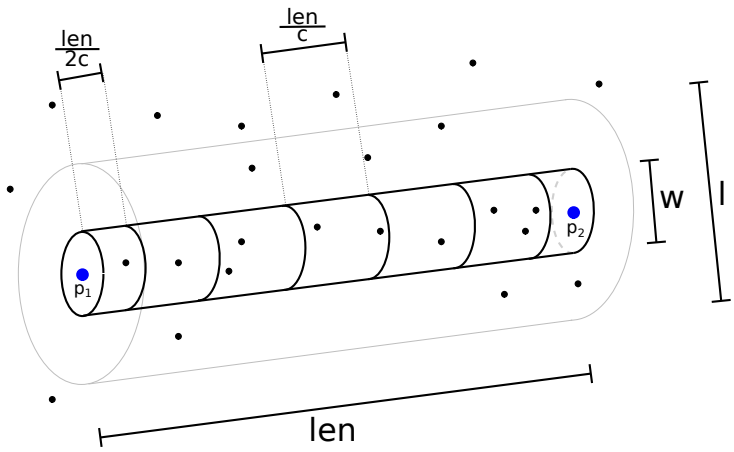

Figure 1.13: A schematic representation of the evaluated alignment determined by a pair of points in the domain. For each candidate, a set of alignment cylinders (with diameter w) and local cylinders (with diameter l) are considered. The candidate alignment cylinder is divided into c pill-boxes. The local point density is evaluated in the local cylinder surrounding the candidate cylinder. A candidate alignment will be validated if the number of occupied boxes is statistically significant given the local point density.

Figure 1.14 shows the detection of an alignment in noise.

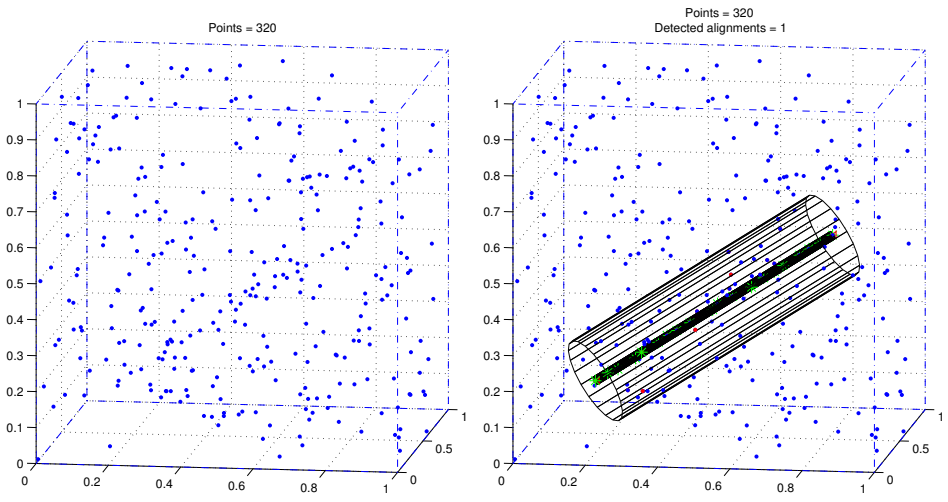

Figure 1.14: Detection of an alignment in noise. Left: An alignment in uniform noise, Right: Detection showing the alignment and the local cylinders. Green points are part of the alignment and red points are in the local cylinder but not in the alignment cylinder.

The full description of the 3D point alignment detector is presented in Chapter 7.

1.9. Summary of the main contributions

- The deep learning stereo method GANet was interfaced with the S2P pipeline to work as the stereo matching step. Experiments showed that it is a valid alternative to the classic counterpart currently in use in the pipeline.
- The deep learning true MVS method CasMVSNet was adapted to work with satellite images following a recent framework. The method achieved results close to the pair-wise MVS baseline without a specific fine-tuning on satellite images.
- An existing method for the post-processing of the disparity map was tested. Inserted in the S2P pipeline before the triangulation step is performed, the method can help achieve reconstructions with better defined structures.
- For pair selection in pair-wise MVS, a novel strategy based on the simulation of satellite images was developed.
- A comprehensive metric (Area under the completeness curve, AUCC) that carries the joint information of completeness and accuracy was evaluated and results indicate that is worth considering for the task of pair selection.
- A simulation tool was implemented that allows to produce views of an artificial 3D scene generating images and RPC models suitable to be used to test or train a satellite pipeline.
- An iterative scheme based on the bilateral filter is proposed as a robust method for the fusion of pair-wise DSMs.
- The point alignment detector based on the a contrario methodology originally for 2D was extended to 3D.

1.10. Summary of publications

- Alvaro Gómez, Gabriele Facciolo, Rafael Grompone von Gioi and Gregory Randall. “Improving the Pair Selection and the Model Fusion Steps of Satellite Multi-View Stereo Pipelines.” In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023.
- Alvaro Gómez, Gabriele Facciolo, Rafael Grompone von Gioi and Gregory Randall. “An experimental comparison of multi-view stereo approaches on satellite images.” In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 844-853. 2022.

- Alvaro Gómez. "An overview of GANet - Guided Aggregation Net for End-to-end Stereo Matching" Image Processing On Line, preprint (2022). https://www.ipol.im/pub/pre/441/
- Alvaro Gómez, Gregory Randall, and Rafael Grompone von Gioi."A contrario 3D point alignment detection algorithm." Image Processing On Line 7 (2017): 399-417. https://doi.org/10.5201/ipol.2017.214

1.11. Summary of demos and code

- "An overview of GANet - Guided Aggregation Net for End-to-end Stereo Matching" Image Processing On Line, preprint
  Demo: https://ipolcore.ipol.im/demo/clientApp/demo.html?id=441
  Code: https://github.com/mlbriefs/441
- "Simsatool: Tool for the simulation of satellite images"
  Code: https://github.com/zemogoravla/simsatool
- "A contrario 3D point alignment detection algorithm." Image Processing On Line.
  Demo: http://demo.ipol.im/demo/214/
- "Good continuation in 3D" Image Processing On Line, workshop demo.
  Demo: http://dev.ipol.im/~agomez/ipol_demo/GoodContinuation3D/
  User:demo, Password:demo
- "Disparity post-processing by guided diffusion" Image Processing On Line, workshop demo.
  https://ipolcore.ipol.im/demo/clientApp/demo.html?id=77777000089

This page intentionally left blank.

Chapter 2

Multi-view stereo in satellite imagery

This chapter presents a brief introduction to some concepts related to multi-view stereo. The satellite stereo pipeline that is used as the baseline for all the experiments and the datasets used for the tests are introduced.

2.1. Stereo

Binocular stereo vision is an area that has been active for many years and is still in constant evolution [40, 49, 75]. Given two images of a scene from different known viewpoints, the objective of stereo is to estimate the most likely 3D shape or depth that explains those images. The change in viewpoint induces a relative displacement of the objects in the scene causing that closer objects move more than far ones in the images of the pair. This apparent motion between the two views (disparity) is inversely proportional to the depth.

In [75], the authors point out that most stereo algorithms perform these four steps: (1) matching cost computation, (2) cost aggregation, (3) disparity computation, (4) disparity refinement.

The first step implies finding sparse or dense correspondences between the images. In the sparse case, characteristic points along with their local features are extracted and compared. In the dense approach, image patches in both images are compared computing the cost of matching the patches for different possible disparities. The search of corresponding patches is simplified by the geometric constraints of the stereo pair (epipolar constraints). Instead of a 2D search for correspondences, the epipolar constraints restrict the search for corresponding image points from the entire image plane to a single line. Moreover, the images can be resampled (stereo-rectification) in such a way that corresponding points are located on the same row.

The matching information is organized usually in a cost volume that stores the costs $C_p(d)$ of matching the position p of the reference image with $p + d$ in the second image for all the considered possible disparity values d.

Matching at the correct disparity is challenging in real life due to the photometric and geometric distorsions introduced by the change of viewpoint and by ambiguities due to occlusions, low texture and repetitive patterns in the scene. The step of cost aggregation tries to overcome this difficulty by imposing spatial coherence to the matching. This can be done by a simple local filtering of the cost volume or in a more comprehensive approach by formulating a global energy minimization problem with a regularization term that enforces the regularity of the disparity map.

Once the cost volume has been regularized, the disparity values can be estimated by processing the volume and applying optimization techniques.

The resulting disparity map may still have erroneous and missing values and several algorithms (filtering, interpolation, inpainting and others) for the post-processing of depth and/or disparity maps have been proposed in the literature [4].

2.2. Multi-view stereo

Multi-View Stereo (MVS) vision aims at reconstructing a 3D scene from multiple 2D views. The goal in MVS is defined in [31] as: “given a set of photographs of an object or a scene, estimate the most likely 3D shape that explains those photographs, under the assumptions of known materials, viewpoints, and lighting

conditions". In practice, in most applications, the assumptions of known materials and lighting conditions are not feasible to satisfy and this already hard problem becomes actually ill-conditioned. Despite these tough conditions, numerous algorithms have been devised in the last decades achieving detailed reconstructions. The main cue exploited in multiview stereo as in stereo is the photometric consistency. The appearance of the scene in corresponding regions across the images is expected to be similar. That is, if the regions in the images "see" the same part of the scene, the image values in those regions or patches should be similar and have a similar structure.

The reconstruction of an object or a scene from a stereo pair may have undefined regions and also regions with erroneous altitude values. This issue originates typically in the stereo matching process. The disparity map resulting from a stereo matching algorithm may have missing values associated with feeble matching in regions lacking good texture cues. Other artifacts may also occur due to repetitive texture giving place to small regions with disparity values far from the correct values. At the borders of structures it can happen that some occluded regions are observed only by one camera and no disparity can be computed. When multiple views of the same scene are available, the mentioned issues can be alleviated and a more complete and accurate reconstruction can be obtained. Basically, two approaches can be taken for the integration of the views:

(a) Pair-wise multi-view approach. Use the images by pairs, compute the stereo reconstruction for several pairs and integrate afterwards into a single final reconstruction.

(b) "True" multi-view approach. Use all the views at a time to compute the reconstruction.

The development of MVS in the computer vision field has concentrated mainly in the reconstruction of objects, buildings and interiors with images taken with standard pinhole-like cameras at ground-level. The state of the art was dominated until recently by pipelines such as COLMAP [76] that can handle thousands of views with images taken from diverse viewpoints and conditions. Although deep learning MVS methods for ground-level images have flourished in the last years [48], classic pipelines as COLMAP are still in the ranks of the main benchmarks of the area [46, 78] and are the preferred option when dealing with many images and/or large size images since deep learning methods struggle to accommodate large 3D structures in GPU memories.

As explained in [89], MVS on satellite imagery has evolved in the remote sensing field almost independently from the computer vision field advances. While true MVS methods are popular for close range imaging [49, 77] they are still seldom used for satellite images as they have not shown significantly better results [39, 89] or are too computationally expensive to be applied to large scale images [21, 57].

In satellite imaging, MVS has been traditionally performed by a pair-wise MVS approach: the views are grouped into pairs and each pair is processed by two-view stereo matching method, producing an elevation model or point cloud; then all the pair-wise reconstructions are fused to obtain a final result.

State-of-the-art pipelines in actual use by academia and in production in space

centers are mostly based on multi-pair approaches [19, 47, 60, 69]. One reason for preferring multi-pair approaches is that satellite imagery has specific characteristics that make the problem tough for multi-view strategies. Multiple images for a certain location can only be acquired through several sweeps sometimes days, weeks or even months or years apart. This may cause great variability in illumination, shadows, reflections, seasonal changes and man-made changes. The variability poses important challenges for the matching of correspondent regions across the images.

2.3. Structure from motion - Bundle adjustment

In order to produce highly detailed reconstructions, MVS requires knowing accurately the viewpoints of all the images in use. When starting with a set of images with unknown or not accurate viewpoints information, a previous step called Structure from Motion (SFM) is required. SFM allows recovering and refining the camera parameters for all the images. Most SFM algorithms [31, 76] rely on a similar pipeline that starts with the detection of keypoints and their associated local features in the images. The features are then matched across the images. The SFM model is determined as the best camera parameters for all the images and set of 3D points in such a way that the 3D points project to corresponding keypoints/-features and the sum of squared distance errors between actual detected keypoints and projected points is minimized across all the images. The minimization that leads to the best cameras and 3D points is known as Bundle Adjustment (BA). An analysis of BA in satellite imagery is done in [56].

2.4. RPC - Rational Polynomial Camera

In satellite images, the projection parameters are usually known for all the images. Satellite images are typically provided along with a Rational Polynomial Coefficients (RPC) camera model [24], and other metadata such as the acquisition timestamp or the pixel size. The RPC camera model is composed of two rational polynomial functions that approximate the mapping from 3D space points to 2D image pixels (the projection function) and its inverse (the localization function). The RPC camera model [37] has become a standard in satellite imagery across image vendors and saves the complications of dealing with the complex specificities of rigorous models of the different sensors.

Although RPCs are expected to be precise enough, the complex system they encode is subject to measurement errors in the satellite geopositioning equipment, mainly due to the attitude angles. Such inaccuracies, also referred to as pointing errors [64], can be of the order of tens of pixels in the image domain. Bundle adjustment is one among different approaches that have been used in the field in order to correct the pointing errors [56]. Another approach is the relative pointing error correction presented in [18].

2.5. The satellite stereo pipeline

In this thesis, the S2P[1] [19] satellite stereo pipeline is used as the baseline for all the experiments. S2P is a simple and modular pipeline developed by the group that won the IARPA Multi-View Stereo 3D Mapping Challenge in 2016 [29].

Figure 2.1 shows an overview of the S2P pipeline. The input is a pair of images with their respective camera models approximated by rational polynomial coefficients (RPC). Input images are cut into small tiles. Tiling allows to locally approximate the pushbroom sensor by an affine camera model with a small error, which enables the use of well established stereo rectification and matching methods [19]. Tiles can also be processed in parallel.

Each tile image pair undergoes a pointing correction (necessary refinement of the RPCs affected by a bias due to usual error in the sensor attitude estimation) and a rectification. On the rectified images, the disparity is computed with the MGM stereo matching algorithm [28]. Finally the computed disparity map gives correspondences in the original (non rectified) images of the tile that can be triangulated (via the refined RPCs) to give a geo-referenced 3D point cloud. The point clouds computed for the tiles can be merged to get the complete point cloud for the stereo pair.

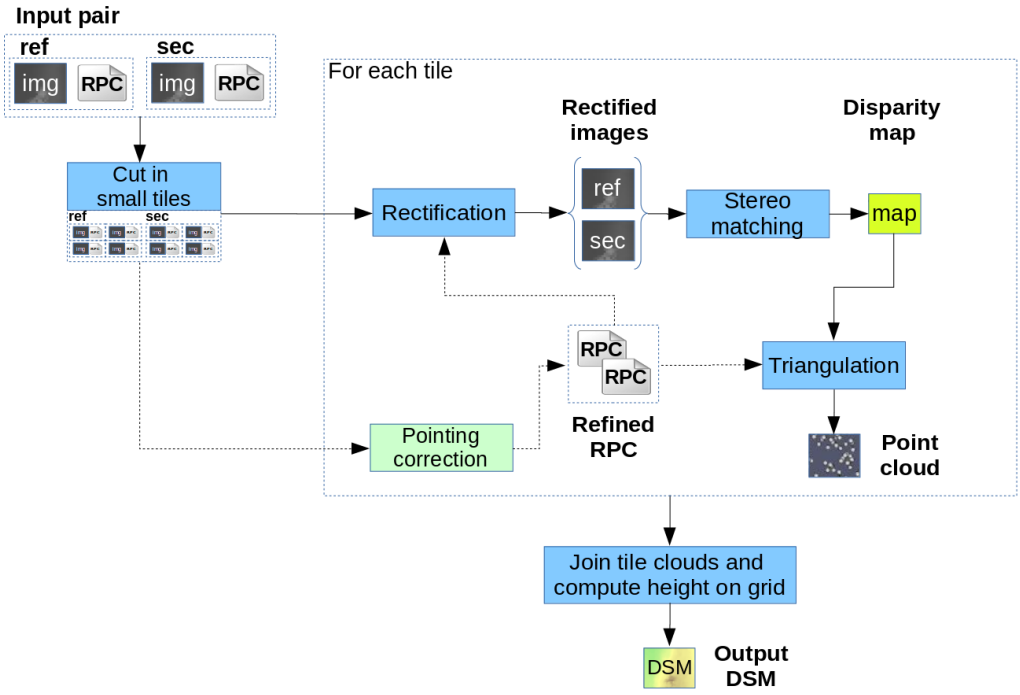

Figure 2.1: S2P [19] overview. The input is a pair of images with their respective rational polynomial camera models, and the output is a DSM.

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

The output point cloud is geo-referenced (X,Y in UTM coordinates and altitude Z in meters). In order to extract the DSM, the S2P pipeline uses a vertical projection approach with these steps:

1. A UTM grid of the desired resolution is initialized.
2. Each cell of the grid is filled with the mean altitude of the points that fall into the cell. The mean altitude can be taken on the points directly above the cell (radius 0) or considering also the points above neighboring cells (radius 1 or above). The former, while more accurate, can result in numerous empty cells. The latter gives place to more complete but also smoother maps.
3. The final DSM is stored as a TIF floating point image.

With the S2P pipeline, a pair-wise MVS scheme can be mounted as depicted in Figure 2.2.

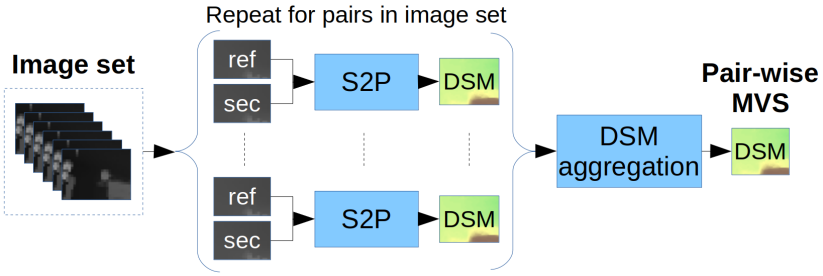

Figure 2.2: The pair-wise MVS scheme.

2.6. Satellite datasets

In order to test the methods in the thesis, satellite images from public datasets are used. We used three datasets, consisting on satellite images from the Multiple View Stereo Benchmark for Satellite Imagery (MVS3D) [10] and the US3D dataset [9].

The MVS3D is a set of 47 satellite images of Buenos Aires (MVS), Argentina. The corresponding GT DSMs are derived from an airborne Lidar acquisition (from a different date than the satellite images) of the same region. The US3D dataset consists of 26 WorldView-3 target-mode panchromatic images collected between 2014 and 2016 over Jacksonville (JAX), Florida and 43 WorldView-3 target-mode panchromatic images collected between 2014 and 2015 over Omaha (OMA), Nebraska. Semantic labels and an airborne Lidar are also available. The Lidar, acquired by the USGS at a different date than the satellite images, is used to derive the GT DSMs.

Figure 2.3 shows ground-truth altitude maps of some regions from Jacksonville (JAX), Omaha (OMA) and Buenos Aires (MVS) used to test the methods presented in the thesis.

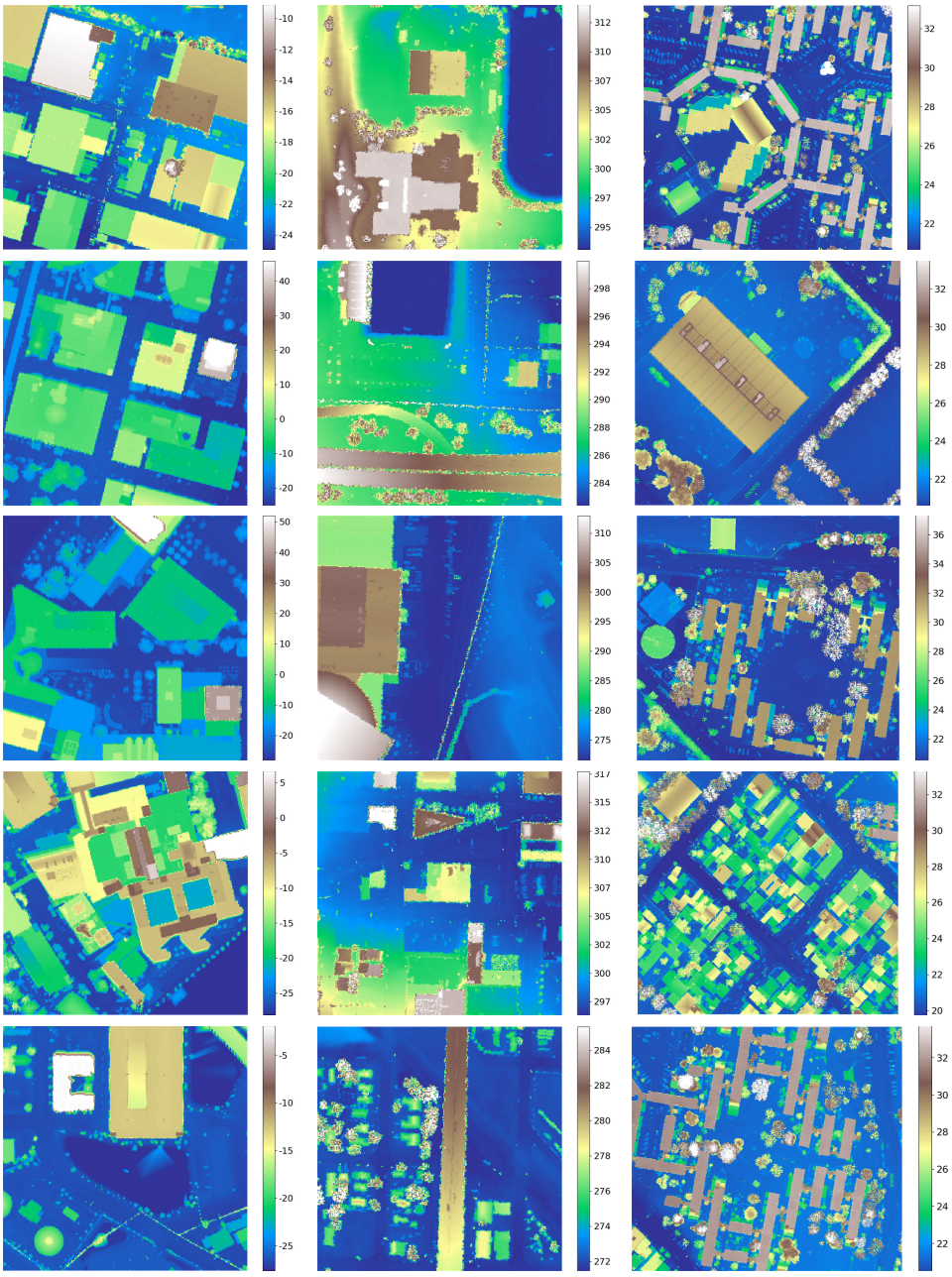

Figure 2.3: Ground-truth altitude maps from datasets used in this thesis. Left: JAX dataset (subregions 156, 165, 214, 251, 264). Center: OMA dataset (subregions 203, 247, 251, 287, 353). Right: Site 1 of MVS3D dataset (subregions 001, 002, 003, 004, 005). Altitudes are in meters.

2.7. Benchmark of reconstructions

The methods presented along the thesis change stages of the baseline pipeline with the objective of improving the stereo or MVS reconstruction. In order to evaluate the methods, images from the datasets in Sec. 2.6 are fed to the pipeline (baseline and modified) and the reconstructed altitude maps are compared to their respective ground-truth altitude map. The following metrics are considered [10]:

Evaluated pixels Proportion of the pixels that have ground-truth information

Bad Z pixels - BAD Proportion of the evaluated pixels where the computed map has an altitude that differs more than z_tol from the ground-truth. For the used datasets, $z\_tol = 1m$

Invalid pixels - INV Proportion of the evaluated pixels where the computed map has an undefined altitude.

Completeness - COMP Proportion of the evaluated pixels where the computed map has an altitude that differs less or equal than z_tol from the ground-truth. Note that $Completeness = 1 - (Bad + Invalid)$.

Average absolute error - AAE Average of the absolute difference between the computed altitude map and the ground truth calculated over the pixels that have valid information both in the ground truth and in the computed map.

Median absolute error - MAE Median of the absolute difference between the computed altitude map and the ground truth calculated over the pixels that have valid information both in the ground truth and in the computed map.

Root mean squared error - RMSE RMSE of the difference between the computed altitude map and the ground truth calculated over the pixels that have valid information both in the ground truth and in the computed map.

Chapter 3

Comparison of multi-view stereo methods

Different methods can be applied to satellite images to derive an altitude map from a set of images. In this chapter we evaluate a set of representative methods from different approaches. We consider true multi-view stereo methods as well as pair-wise ones, classic methods and deep learning based ones, methods already in use on satellite images and others that were originally devised for close range imaging and are adapted to satellite imagery. While deep learning (DL) methods have taken over multi-view stereo reconstruction in the last years, this tendency has not fully reached satellite stereo pipelines, that still largely rely on pair-wise classic algorithms. For the comparison, we set-up a framework that allows to interface a DL-based stereo method taken from the computer vision literature with a satellite stereo pipeline. For multi-view stereo algorithms, we build on a recently proposed framework originally devised to apply the Colmap method to satellite images. Methods are compared on several datasets that include sets of images taken within a few days and sets of images taken months apart. Results show that DL methods have, in general, a good generalization power. In particular, the use of the GANet DL method as the matching step in a pair-wise stereo pipeline is promising as it already performs better than the classic counterpart, even without a specific training.

3.1. Introduction

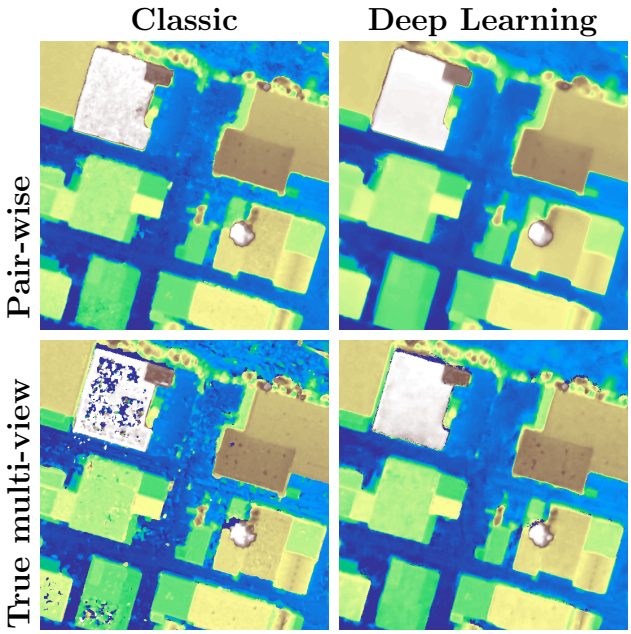

Figure 3.1: Digital Surface Models (DSM) computed by the methods analyzed in this work on the subregion 156 of JAX_NIT dataset. Methods from top to bottom and left to right: S2P, S2P-GANet, COLMAP, and CasMVSNet.

Stereo vision is one of the most active fields in computer vision, with constant advances along the years [40, 49, 75]. Initial methods worked on one stereo pair. Then, Multi-View Stereo (MVS) was first approached as an extension of the stereo algorithms by aggregating the information of multiple stereo pairs. True MVS algorithms considering directly all the images of the scene arrived some time afterwards [13]. True MVS algorithms were mostly devised, by the computer vision community, for the reconstruction of objects, buildings and interiors with images taken with standard pinhole-like cameras at close distance [31, 77]. Deep Learning (DL) MVS methods [49] flourished in the last years and have taken over the top rankings of the main benchmarks of the area [46, 78], but classic methods are still a valid option when dealing with many and/or large size images since DL methods struggle to accommodate large 3D structures in GPU memories.

In the case of satellite images, MVS has traditionally been performed with pair-wise approaches where the multiple views are treated by pairs doing traditional two-view stereo and then aggregating the pair-wise reconstructions (elevation models or point clouds, for example) to get the final result [19, 47, 60]. Satellite images have specific characteristics that have historically discouraged the use of true MVS methods, for example: (a) the extremely small ratio between the depth range and the distance from the camera to the scene implies working with a camera model that deviates from the standard pinhole and deals with structures that occupy few

pixels in the images; (b) the images for a certain location can only be acquired through several sweeps which may be days, months or, even years apart, introducing variability in illumination, seasonal changes and man-made changes, among others. The variability poses important challenges for the matching of correspondent regions across the images. This variability problem has usually been tackled with a heuristic selection of best pairs that tend to minimize separation in time of the images and prefer viewing angles that ensure less error in triangulation [29].

A recent work [89] in 2019 showed that classic true MVS algorithms used in computer vision could be adapted to satellite images for the benefit of the remote sensing field. We build upon that work and extend the concept to include stereo and MVS methods based on DL.

This chapter is a concise evaluation of a set of methods which are representative of different approaches that can be applied to satellite images in order to derive an altitude map from a set of images. It is not an extensive benchmark attempt. The selected methods span different interesting aspects: methods already in use on satellite images and others originally devised for close range imaging and adapted to satellite imagery, classic and DL approaches, pair-wise and multi-view reconstruction methods.

A simple and modular satellite image processing pipeline (S2P) [19] is considered as a baseline. Experiments in this section explore if modifications in the pipeline or the use of other methods adapted to satellite imagery can give promising or better results in comparison to the already established pipeline. A framework is built in order to compare different methods on several datasets that include sets of images from different sites and that consider acquisitions over short and long periods of time. The comparison shows that the analyzed methods attain comparable and sometimes better performance than the classical ones. Figure 3.1 shows typical results, where rows compare pair-wise vs. multi-view approaches and columns compare classic vs. DL methods.

3.2. Framework for the comparison

The different methods are applied to the datasets following the scheme depicted in Figure 3.2. For pair-wise methods, a Digital Surface Model (DSM) is computed for every possible pair of images of a subregion; these DSMs are then aggregated to get an enhanced multi-pair DSM. On the other hand, true multi-view methods are fed with all the images of a subregion.

In order to assess the performance of the different approaches, the computed DSMs are compared against the ground-truth DSM on all the datasets. The comparison is done as shown in Figure 3.3. First each altitude map is registered to the ground-truth map and then compared to it. A normalized cross correlation approach is used in order to obtain integer X,Y translations that register the DSMs and the altitude is adjusted by the median of the difference to the ground-truth. Once registered, the following metrics [10] are computed: COMP, MAE and RMSE.

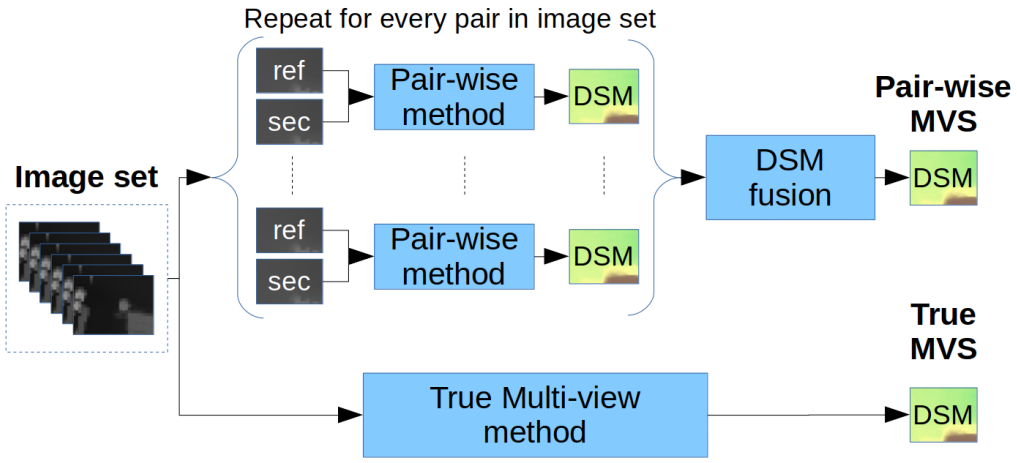

Figure 3.2: Scheme of multi-pair and true multi-view methods.

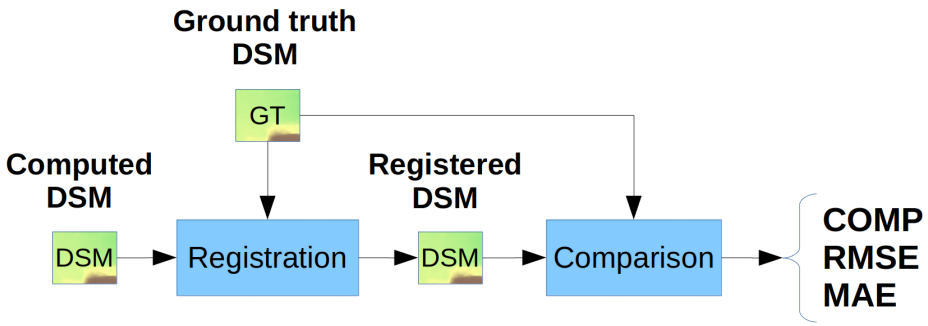

Figure 3.3: Flow diagram for the comparison of the computed DSM with respect to the ground truth altitude.

3.3. Methods

Table 3.1 summarizes the evaluated methods. A brief description of each one and the necessary adaptations to satellite imagery are presented hereafter.

3.3.1. S2P

Figure 3.4 shows an overview of the S2P [19] pipeline[1], already presented in Chapter 2. The input is a stereo pair of images with their respective camera models expressed by rational polynomial coefficients (RPC). Each image pair undergoes a pointing correction and is rectified. Disparity is computed on rectified images using the MGM stereo matching algorithm [28]. Computed correspondences are then triangulated to produce a geo-referenced 3D point cloud and an altitude map. The pipeline is configured to use only one tile as the images considered in

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

Table 3.1: Tested methods

Method	Type	DL	Notes
S2P [19]	Pair-wise	No	MGM [28] in disparity computation
S2P-GANet	Pair-wise	Partially	GANet [88] in disparity computation
COLMAP [76, 77]	Multi-view	No	Adapted for satellite images by [89]
CasMVSNet [38]	Multi-view	Yes	Adapted for satellite images in this work

this chapter are small enough to be processed without being cut in more tiles.

3.3.2. GANet

GANet [88] uses Deep Neural Networks (DNN) to compute a disparity map. As other DNN methods [49], it follows the traditional stereo steps: dense features are extracted for both images, the cost of matching the features at different disparities is organized in a Cost Volume (CV), which is regularized by aggregation and/or filtering and finally a map with minimal cost is derived from the CV.

In most DNN based stereo methods, cost aggregation is done by 3D convolutions, usually in an hourglass configuration [49]. 3D convolutions imply large memory requirements; the computational burden restricts the size of images that can be processed. GANet takes a different approach by introducing a Semi-global Guided Aggregation layer (SGA) which implements a differentiable approximation of Semi-Global Matching (SGM) [42]. SGA is followed by a Local Guided Aggregation layer (LGA) that performs a local filtering. SGA and LGA weights are generated by an auxiliary "guidance subnet" fed with original images and the extracted features.

A more detailed overview of GANet is presented in appendix A.

S2P-GANet: Adaptation to satellite images In this work, GANet is used as an alternative "stereo matching" step in the S2P pipeline, see Figure 3.4. The stereo matching step receives a rectified stereo pair of images and computes disparity maps in both directions: left-to-right and right-to-left. A consistency check is performed to filter out pixels with non congruent disparities [12, 30]. In order to use GANet in the S2P pipeline, some adaptations have to be considered:

a) Negative disparities: In most stereo algorithms, a CV is computed and regularized, and a disparity map is derived from it. The CV is computed for a certain range of possible disparity values, which must be known *a priori* or estimated. In the S2P pipeline, the disparity range is traditionally estimated by the sparse matching of interest points (e.g. SIFT keypoints [55]), but other strategies are allowed such as specifying a fixed known disparity range or estimating the disparity range from a known altitude range. In several stereo matching algorithms, including [28]

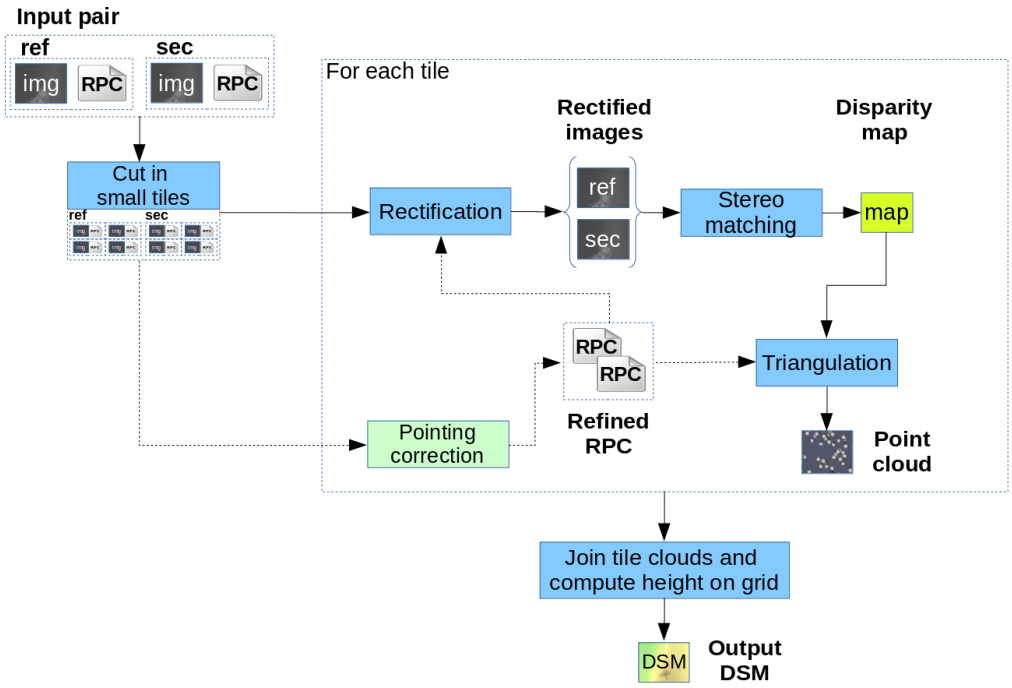

Figure 3.4: S2P overview. The input is a pair of images with their respective RPC camera models, the output is a DSM given as a georeferenced 3D point cloud and as an altitude image.

used by default in S2P, disparity admits positive and negative values. GANet, however, accept only negative disparities. That is, all pixels in the secondary rectified image must "move" to the left relative to the rectified reference image. In this work, a fixed known altitude range is used, based on the ground truth plus an additional tolerance. The S2P pipeline was adapted to get a rectification compatible with negative disparities.

b) GPU memory restrictions: The size of the rectified stereo images that can be handled by GANet is bounded by the available memory in the GPU. Also, images' width and height must be multiple of 48. A tiling strategy is thus implemented to process large images. The disparity estimation is more error prone at tile borders. So tiles are chosen as large as possible and the overlaps are merged considering the distance to the border as a weight. The experiments reported in this work use tiles of 1872×480 pixels and were run on a Nvidia Tesla P100 GPU with 12Gb of RAM.

3.3.3. COLMAP

COLMAP [77] is part of a family of methods [31,32,35] that focus on large-scale dense reconstruction and fusion. These methods aim to integrate the information of diverse multiple images of a scene such as crowd-sourced image datasets. COLMAP is closely related to [90] and considers, as other MVS methods, a variety

of photometric and geometric priors ensuring consistency among different views. The method follows a generalized Expectation-Maximization (EM) scheme with alternating and interleaved estimation of occlusions in the E-step and depths in the M-step. The depth estimation M-step is based on PatchMatch [8] where the tested hypothesis are based on the depths, the normals and their perturbations.

Zhang et al. [89] adapted COLMAP to the peculiarities of satellite images. For that, they work on local scene coordinates, approximate the RPC camera model with a perspective camera and reparameterize the depths as heights over a horizontal reference plane that lies below the scene. In the adaptation, the hypothesis depth planes become the horizontal planes in the scene.

To approximate the camera, the scene volume is sampled in a regular grid and projected by the RPC model. This yields a set of 3D-2D correspondences that are used to fit a 3×4 projection matrix.

Being $(0, 0, 1, -d)$ the coefficients of the horizontal reference plane (plane equation $n^t x - d = 0$ with $n = (0, 0, 1)^t$), the reparameterization of the depths is handled extending the 3×4 projection matrix with a fourth row

$$\begin{pmatrix} u \\ v \\ 1 \\ m \end{pmatrix} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \mathbf{P}_{13} & \mathbf{P}_{14} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \mathbf{P}_{23} & \mathbf{P}_{24} \\ \mathbf{P}_{31} & \mathbf{P}_{32} & \mathbf{P}_{33} & \mathbf{P}_{34} \\ 0 & 0 & \overline{Z} & -\overline{Z}d \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}, \tag{3.1}$$

where Z is the conventional depth (distance from the camera center) and m the reparameterized depth (the height over the reference plane). $\overline{Z}$ is computed as the average conventional depth of all the sparse scene points in the Structure From Motion (SFM) step. In order to check the photometric consistency between two views at a certain depth, COLMAP, as other MVS methods, computes an homography between the first and second view as:

$$\mathbf{H} = \mathbf{K}_2 \left(\mathbf{R}_{12} - \frac{n^T t_{12}}{f} \right) \mathbf{K}_1^{-1}, \tag{3.2}$$

where $(n, -f)$ are the coefficients of the hypothesis plane that induces the homography, $\mathbf{R}_{12}$ and t_{12} indicate the pose of camera 2 w.r.t camera 1; and $\mathbf{K}_1$ and $\mathbf{K}_2$ are the intrinsic parameters of the cameras. In the adaptation, Zhang et al. [89] propose a more stable homography computation based on the 4×4 projection matrices instead of operating with the intrinsic and extrinsic parameters.

3.3.4. CasMVSNet

True multi-view stereo methods based on DNN construct a CV in a similar manner as DNN stereo methods [49]. MVSNet [87] is a well known representative of this category. Given a reference image, fronto-parallel hypothesis planes at different depths are considered. Features are then extracted for each image and differentiable homographies are used to warp the 2D feature maps into the hypothesis planes of the reference camera to form feature volumes (one feature volume per image). The CV is computed as the variance of all the feature volumes. The cost regularization

is done by 3D convolutions. A raw depth map is regressed from the regularized CV and finally refined taking into account the information of the reference image.

As mentioned above, aggregation with 3D convolutions struggles with memory and computational costs growing cubically. The Cascade Cost Volume for High-Resolution Multi-View Stereo (CasMVSNet) [38] is a true multi-view stereo method that tries to overcome this issue with a coarse-to-fine approach. Features are extracted at multiple scales. At the early stages of the cascade the CV is built taking into account large scale features and sparse plane hypothesis. This low resolution CV leads to a raw depth that is subsequently used to adjust the depth sampling. Progressing through the stages, features at finer scales are considered but on the other hand the considered depths hypothesis are narrower bands surrounding the previous estimated depths. This keeps bounded the size of the CV along the stages.

Adaptation to satellite images The adaptation of the algorithm in order to use it with satellite images is based on the COLMAP adaptation [89] summarized in Section 3.3.3. The perspective approximation of the RPC cameras into 4×4 extended projection matrices is borrowed from the output of the SFM step of a run of the adapted COLMAP.

In the code of the adapted CasMVSNet, projections are handled with Equation 3.1 and homographies are computed as proposed by [89]. In order to preserve the ordering of the planes as trained in CasMVSNet (from near to far relative to the camera), the horizontal planes are traversed in decreasing height (from near to far).

3.3.5. DSM aggregation criteria

In the case of pair-wise MVS, it is well known that DSM aggregation improves in general the completeness [29, 64]. However, if the DSM computed from a bad pair is included, the result degrades. For this reason it is essential to pre-select the pairs to be aggregated. In [29] a very simple heuristic based on two conditions on the metadata of the images was proposed: **a) filtering:** both images in the pair must have an incidence angle smaller than 40^{o} and the angle between views should be in the $[5^{\text{o}}, 45^{\text{o}}]$ range, preferably around 20^{o}; **b) ordering:** pairs are ordered by the increasing absolute difference between the dates of the images. This is the *a priori* selection criterion used in our experiments. But, this is not enough to filter out all bad matches. Indeed, we observed that strong seasonal changes can lead to very bad results independently of the metadata.

For this reason we apply an additional selection criterion on the pairs, which determines *a posteriori*, after computing the DSM, whether it should be aggregated or not. A DSM is aggregated if it has more that a certain number of valid (not undefined) pixels. In our experiments a minimum of 70 % of valid pixels was considered.

3.4. Datasets

The methods described above were tested on three datasets, consisting on stereo satellite images from the Multiple View Stereo Benchmark for Satellite Imagery (MVS3D) [10] and the US3D dataset [9] that were introduced in Chapter 2.

For our evaluation, 5 subregions from each of the datasets are considered. The selection is representative of the datasets but arbitrary in any other aspect, see Figure 2.3. In each subregion, a limited set of images is considered in order to allow a tractable pairwise analysis: 6 images acquired in a small time interval (same day or some days apart) and 6 images acquired in a longer time interval (spanning months) are considered. These sets of images are given the suffixes NIT and FIT, which stand for near-in-time and far-in-time, respectively. Table 3.2 summarizes the set of images used for the experiments.

Table 3.2: Datasets used for evaluation. For each location, five subregions are considered. For each subregion, six images acquired in a small time interval (same day or some days apart) and six acquired in a longer time interval (months apart) are considered.

Dataset	**Subregions**	**Time span (days)**	**Alias**
MVS3D	MVS_ {001, 002, 003, 004, 005}	1	MVS_NIT
US3D_JAX	JAX_ {151, 165, 214, 251, 264}	44	JAX_NIT
US3D_OMA	OMA_ {203, 247, 251, 287, 353}	24	OMA_NIT
US3D_JAX	JAX_ {151, 165, 214, 251, 264}	386	JAX_FIT
US3D_OMA	OMA_ {203, 247, 251, 287, 353}	405	OMA_FIT

3.5. Experiments

3.5.1. Methods on stereo pairs

The S2P, S2P-GANet and COLMAP methods were tested on all the possible pairs for each subregion. With 6 images per subregion, there are 30 possible pairs considering the order of the images. Methods based on the S2P pipeline already work on stereo pairs, while COLMAP is set to work on the minimal set of two images. Table 3.3 presents the results, averaging only the DSMs that have at least 70 % of valid pixels.

As expected, and consistently with what was reported in [89], S2P-based methods outperform the adapted COLMAP, which is not intended to work just on image pairs. Among the two variants of S2P, Figure 3.5 shows that results are quite similar in completeness and MAE for both methods. S2P-GANet achieves better values of RMSE accuracy which denotes a lower dispersion in the altitude error values of the DSMs.

The S2P-GANet variant achieves comparable or better results than the S2P pipeline, even without having been trained on satellite images, as illustrated in Table 3.3.

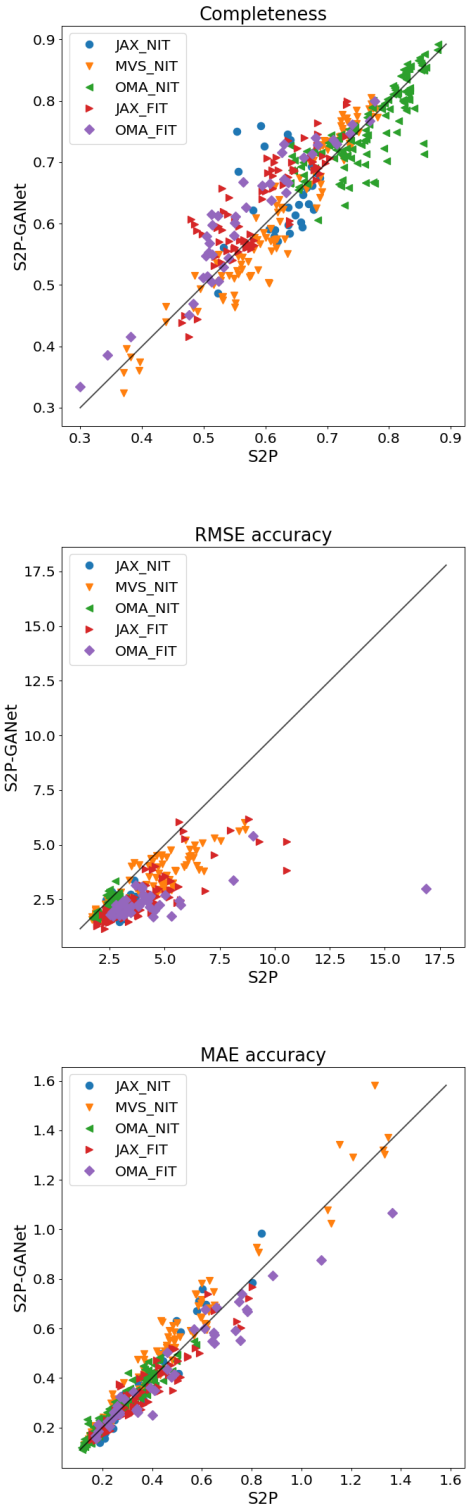

Figure 3.5: Results on stereo pairs for S2P and S2P-GANet. Each marker in the plots corresponds to the evaluation of a stereo pair reconstruction against the ground truth with both methods. The black line in the plots corresponds to the locus of equal metric values. Note how, overall the considered regions, results for both methods are similar in completeness and MAE. RMSE values for S2P-GANet are lower than the values for S2P which denotes less dispersion in the altitude errors for the former.

Table 3.3: Results on stereo pairs for S2P, S2P-GANet and COLMAP methods. The results are the average of the metrics on each of the datasets. Only DSMs with at least 70 % of valid pixels are considered. The best metrics on each row are depicted in bold.

	S2P				S2P-GANet				COLMAP			
	# DSMs	COMP	RMSE	MAE	# DSMs	COMP	RMSE	MAE	DSMs	COMP	RMSE	MAE
JAX_NIT	139	**0.61**	4.25	**0.42**	127	**0.61**	**3.22**	0.48	119	0.53	18.03	0.58
MVS_NIT	146	**0.62**	2.89	0.51	131	**0.62**	**2.57**	**0.47**	111	0.56	7.41	0.55
OMA_NIT	136	**0.76**	2.53	**0.27**	143	**0.76**	**2.16**	0.28	123	0.57	17.18	0.53
JAX_FIT	114	0.57	4.85	0.35	90	**0.62**	**2.80**	**0.34**	99	0.38	30.37	1.25
OMA_FIT	71	0.54	4.94	0.50	58	**0.60**	**2.54**	**0.47**	63	0.16	39.25	8.48

3.5.2. Aggregated pair-wise DSM

Pair-wise DSMs computed for S2P and S2P-GANet methods are aggregated after being selected and ordered with the criteria explained in Section 3.3.5. Among the criteria that hasve been used, the *a posteriori* filtering becomes relevant in the case of sets of images with important seasonal changes as seen in Figure 3.6. Note that in OMA_FIT dataset, less than half of pair-wise DSMs pass the criterion and are considered for the aggregation (see column #DSMs in Table 3.3), however the heuristics with *a posteriori* filtering achieves similar results as the oracle guided integration.

For the aggregation, the selected DSMs are registered to the ground truth DSM and then reduced by the median. Note that this procedure is used for comparing the methods but cannot be used in real cases when ground truth is not available. In a realistic setting a surrogate DSM must be selected as reference for the registration [29].

Figure 3.7 illustrates the progression of DSM integration for S2P and S2P-GANet on two subregions. Second and fourth rows show the error reduction induced by DSM aggregation. Note the better performance of S2P-GANet, even with less DSMs.

3.5.3. Pair-wise vs true multi-view methods

Table 3.4 presents the results for aggregated pair-wise and true multi-view methods. As can be seen, pair-wise methods outperform true multi-view methods in almost all datasets and metrics.

3.5.4. Fine-tuning

Although there are multiple datasets for training stereo and multi-view stereo algorithms, these datasets are mostly comprised of close range scenes [46,75,78]. In the last years some datasets were deployed that allow training in long range scenes such as the WHU MVS/Stereo dataset (WHU dataset) [53] and the RVL Purdue SatStereo (RVL dataset) [65]. WHU is a synthetic aerial dataset produced out of thousands of real aerial images covering an area over the Guizhou Province in China. RVL Purdue SatStereo dataset provides a set of stereo-rectified images and associated ground truth disparities for areas of interest drawn from two sources: IARPA's MVS Challenge dataset [10] and the CORE3D-Public dataset [9].

Table 3.4: Results of the tested pair-wise and true multi-view methods on each subregion of the datasets.

	Pair-wise								True multi-view					
	S2P				S2P-GANet				COLMAP			CasMVSNet		
	# DSMs	COMP	RMSE	MAE	# DSMs	COMP	RMSE	MAE	COMP	RMSE	MAE	COMP	RMSE	MAE
JAX_156	30	0.822	1.803	**0.123**	30	0.813	1.864	0.144	0.763	3.322	0.175	**0.830**	**1.423**	0.147
JAX_165	25	**0.690**	6.042	**0.311**	20	0.676	4.260	0.431	0.570	6.645	0.515	0.624	**3.741**	0.381
JAX_214	24	**0.699**	5.868	**0.271**	22	0.671	4.847	0.417	0.593	8.480	0.455	0.612	**4.331**	0.485
JAX_251	30	**0.675**	4.747	**0.313**	25	0.661	3.946	0.396	0.561	12.990	0.621	0.654	**2.968**	0.351
JAX_264	30	0.792	2.562	**0.160**	30	**0.833**	**2.147**	0.183	0.697	7.341	0.210	0.755	2.231	0.214
MVS_001	30	**0.744**	2.570	**0.282**	25	0.730	**2.388**	0.367	0.675	2.666	0.389	0.682	2.917	0.329
MVS_002	30	**0.828**	2.024	**0.158**	30	0.823	**1.945**	0.180	0.787	1.978	0.223	0.827	2.587	0.164
MVS_003	28	**0.688**	3.873	**0.338**	23	0.667	**3.851**	0.415	0.645	3.915	0.413	0.642	3.913	0.359
MVS_004	30	**0.707**	**2.218**	**0.358**	27	0.654	2.263	0.544	0.681	2.324	0.392	0.665	2.461	0.412
MVS_005	28	**0.670**	2.983	0.440	26	0.663	**2.683**	0.433	0.632	2.867	0.488	0.633	2.760	**0.407**
OMA_203	29	0.867	1.728	**0.105**	30	**0.871**	**1.678**	0.127	0.817	1.779	0.151	0.800	1.923	0.236
OMA_247	28	0.841	2.458	**0.146**	27	**0.845**	**2.345**	0.149	0.778	2.973	0.217	0.803	2.544	0.221
OMA_251	25	0.903	2.769	**0.095**	30	**0.915**	**2.240**	0.124	0.856	3.000	0.167	0.835	2.360	0.219
OMA_287	24	0.832	2.561	0.155	28	**0.853**	**2.025**	**0.138**	0.743	3.801	0.220	0.763	2.858	0.285
OMA_353	30	0.836	2.455	**0.126**	28	**0.837**	**2.387**	0.138	0.796	2.420	0.192	0.792	2.738	0.193
JAX_156	29	0.791	2.031	**0.123**	30	**0.828**	1.739	0.124	0.693	7.715	0.188	0.800	**1.579**	0.171
JAX_165	15	0.692	5.985	**0.274**	6	**0.706**	**4.195**	0.310	0.579	10.845	0.384	0.529	5.267	0.465
JAX_214	19	0.678	6.615	**0.244**	12	**0.701**	**3.951**	0.285	0.586	12.693	0.334	0.444	4.257	0.655
JAX_251	21	0.681	4.461	0.285	18	**0.703**	**3.624**	**0.284**	0.611	15.230	0.370	0.610	4.447	0.371
JAX_264	30	0.728	3.074	0.187	24	**0.794**	**2.407**	**0.181**	0.621	8.564	0.263	0.687	2.807	0.286
OMA_203	17	**0.729**	2.650	**0.229**	7	0.707	2.381	0.392	0.500	20.750	0.537	0.602	**2.336**	0.504
OMA_247	13	0.792	2.796	0.229	10	**0.825**	2.501	**0.198**	0.574	13.987	0.383	0.725	**2.419**	0.305
OMA_251	14	0.859	3.224	**0.156**	17	**0.891**	**2.318**	0.208	0.623	17.756	0.312	0.679	3.049	0.388
OMA_287	15	0.772	3.853	0.243	12	**0.822**	**2.557**	**0.211**	0.537	16.973	0.534	0.610	4.245	0.462
OMA_353	12	0.740	2.652	**0.315**	12	**0.744**	**2.511**	0.375	0.528	16.477	0.651	0.693	2.694	0.459

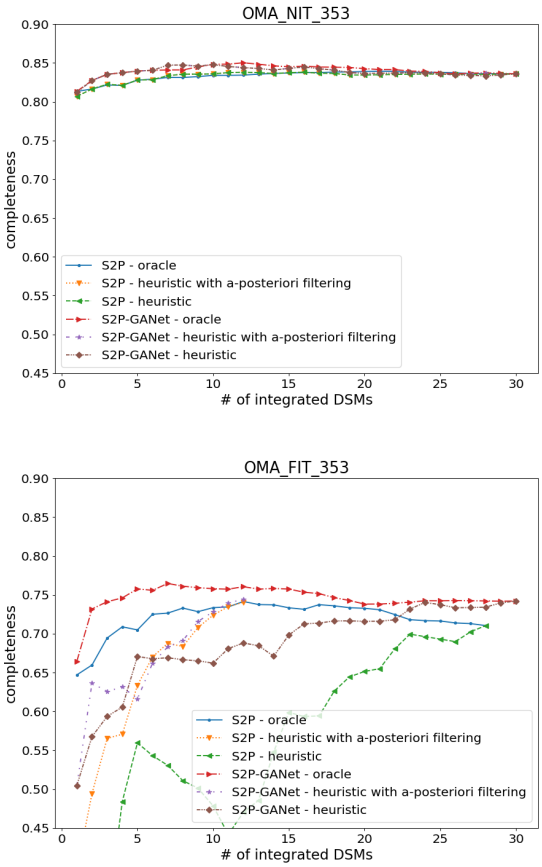

Figure 3.6: Evaluation of the *a posteriori* pair selection criterion. Comparison of the progressive integration of pair-wise DSMs using the completeness *oracle* (DSM are ordered by completeness) and the heuristic rules based on metadata with and without the *a posteriori* filtering. Left: subregion 353 of OMA_NIT. Right: subregion 353 of OMA_FIT. The use of the *a posteriori* criterium becomes relevant in the case of the OMA_FIT dataset which presents important seasonal changes between images.

GANet developers made available three pretrained models. The results reported in previous sections use the very basic model trained on the Sceneflow [58] dataset for only 10 epochs. This model is intended, in principle, as a starting point for fine tuning, but it was used as-is in our work. Tests were conducted on the other two available models which are fine-tuned for the Kitti2012 and Kitti2015 [34, 59] benchmarks. Best results were attained with the basic model.

A fine-tuning of GANet, starting from the basic model and using the WHU and RVL datasets, was performed. For the training on WHU, over 8000 stereo pairs from the training set of the dataset were used and results are reported after 9 epochs. Figure 3.8 shows an example of a stereo pair of the WHU-stereo dataset. The pairs in this dataset are of size 768×384 and with negative disparities complying with the requirements of GANet. On the other hand, the RVL SatStereo dataset is comprised of large images with disparities not restricted to

Figure 3.7: Two examples of the progressive integration of pair-wise DSMs for the S2P and S2P-GANet methods. Second and fourth rows show the difference with the respective ground truth DSM. Colorbars are in meters.

negative values. Images and ground truth disparities were re-rectified in order to have negative disparities and a reduced disparity range. Crops of size 768×384 were extracted from the RVL dataset (sites Master Provisional MP1 and MP2, and Master Sequestered MS1 and MS2 from the IARPA MVS sets of RVL) and 440 stereo pairs were created for the training. Figure 3.9 shows an example of a stereo pair and disparity map from the MP1 subset of the RVL dataset. Note how the disparity has opposite sign in the top and bottom of the images with a shear effect between the rectified images.

In the rectification step, two planar transformations S_1 and S_2 are computed. These transformations map the images of the pair in such a way that the epipolar lines become horizontal. More precisely, since the RPC projection for each view can be approximated by an affine camera model if the images are small or the images are processed by tiles. In the affine camera model, the epipolar lines are bundles of parallel lines and the rectifying transformations S_1 and S_2 are similarities (a composition of translations, rotations and scalings).

It can be seen that each rectified image can undergo an arbitrary horizontal shear and still be correctly rectified. Most rectification methods [54] deal with this degree of freedom by limiting the distortion of the images (thus limiting the shear). However, we note that this shear would allow to register the ground plane.

The shear compensation is implemented as a horizontal shear on the second image. The rectification similarity S_2 becomes an affinity S_2'

$$S_2' = \begin{bmatrix} a & b & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} S_2,$$

where a, b, c are computed by least squares as to minimize the horizontal distance between a set of corresponding keypoint matches $(x_1^i, y_1^i), (x_2^i, y_2^i)$ in the rectified images:

$$(\hat{a}, \hat{b}, \hat{c}) = \arg \min_{(a,b,c)} \sum_i \|a x_2^i + b y_2^i + c - x_1^i\|_2^2.$$

This correction has two advantages. It reduces the disparity range and also reduces the distortion that may affect the surfaces parallel to the ground plane which is, in general, the dominant plane in urban scenes.

Figure 3.10 shows the crops extracted from the images in Figure 3.9 with a standard size and where the shear effect has been removed. Results on this dataset are reported after a training of 65 epochs.

The results on stereo pairs for the basic model and the two fine-tuned models are presented in Table 3.5. In general, similar or better results are obtained in completeness and in the accuracy with the fine-tuned models.

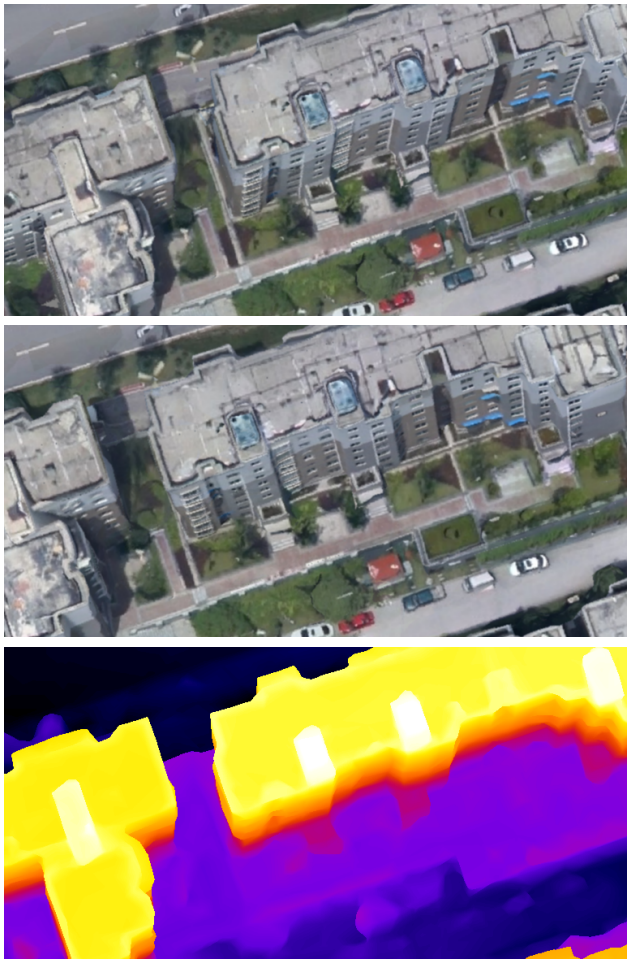

Figure 3.8: An example of stereo pair and disparity from the WHU-stereo dataset. Top: left and right images, Bottom: Disparity map.

Table 3.5: Results of S2P-GANet on the stereo pairs trained with SceneFlow, WHU and RVL datasets. Results are the average of the metrics on each of the datasets. Only DSMs with at least 70 % of valid pixels are considered. Best metrics on each row are in bold.

	SceneFlow (basic model)				WHU (9 epochs)				RVL (65 epochs)			
	# DSMs	COMP	RMSE	MAE	# DSMs	COMP	RMSE	MAE	DSMs	COMP	RMSE	MAE
JAX_NIT	127	0.61	3.22	0.48	127	**0.62**	3.37	**0.43**	116	0.60	**2.80**	0.55
MVS_NIT	131	0.62	2.57	0.47	134	**0.64**	2.55	**0.42**	140	0.62	**2.36**	0.47
OMA_NIT	143	**0.76**	2.16	0.28	149	**0.76**	2.20	**0.24**	138	**0.76**	**2.14**	0.28
JAX_FIT	90	0.62	2.80	0.34	70	**0.64**	2.32	**0.27**	92	**0.64**	**2.35**	0.35
OMA_FIT	58	0.60	2.54	0.47	26	0.63	2.24	**0.34**	66	**0.64**	**2.03**	0.38

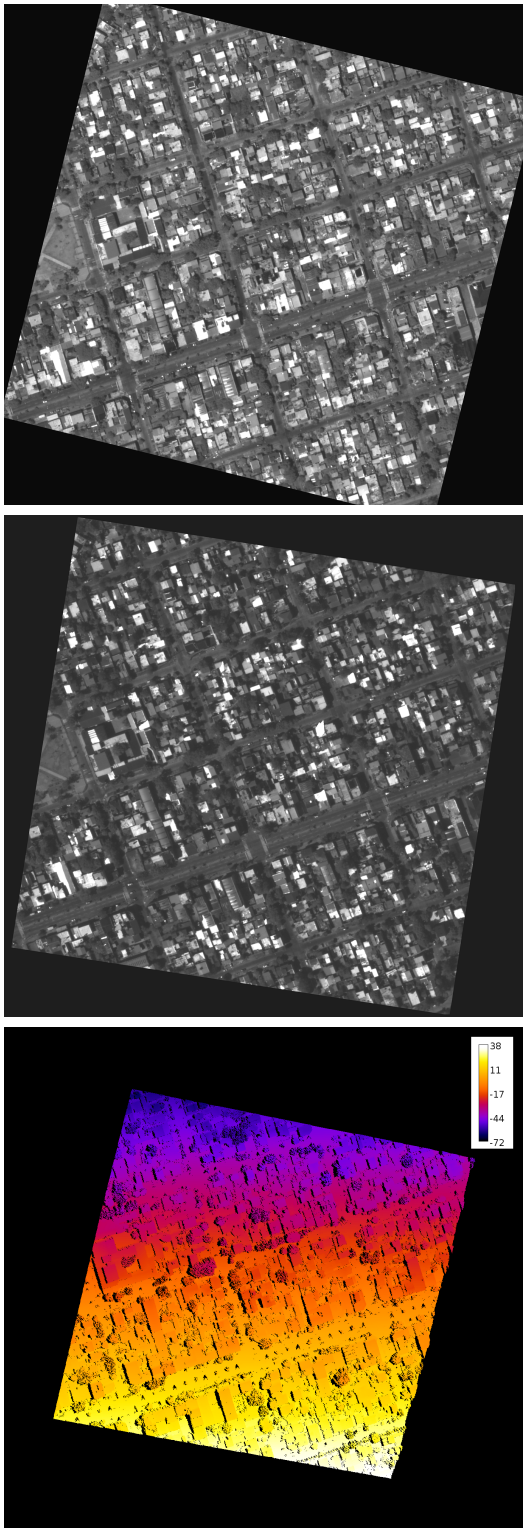

Figure 3.9: An example of stereo pair and disparity from the MP1 subset of the RLV SatStereo dataset. Top: left and right images, Bottom: Disparity map. Note how the disparity has opposite sign in the top and bottom of the images with a shear effect between the rectified images.

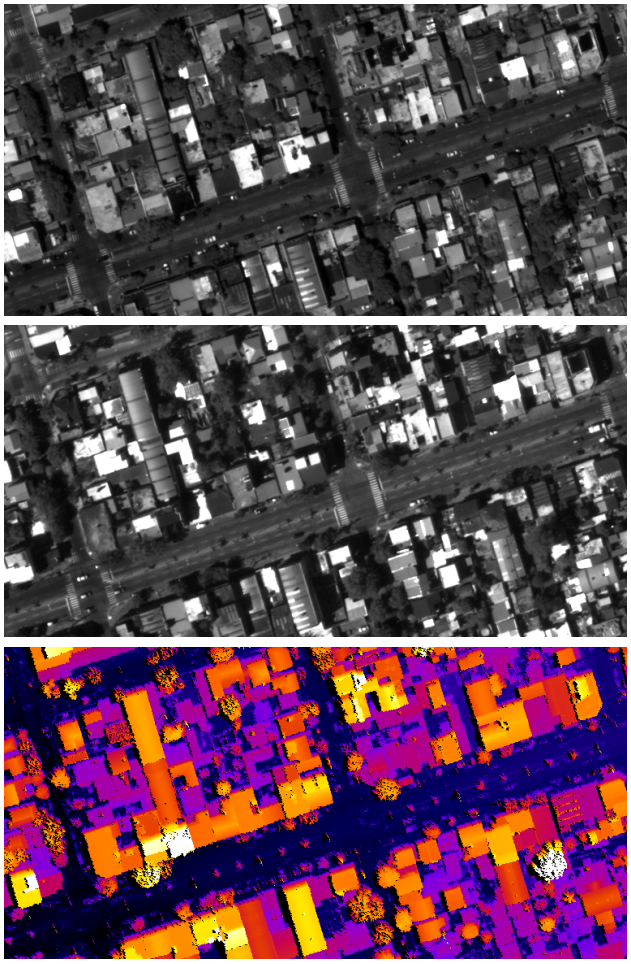

Figure 3.10: Stereo pair and disparity map crops extracted from the RVL sample shown in Figure 3.9 . Top: left and right images, Bottom: Disparity map.

3.6. Discussion

Satellite images have specific characteristics that hinder the adaptation of well established methods used on close range images. Most satellite pipelines in use are based on pair-wise approaches with classic methods that are known to achieve accurate results. This study confirms this fact showing that it is hard to beat the baseline pipeline. On the other hand, results also expose that other valuable methods from the computer vision field can be adapted to work on satellite images since they get results comparable and in some cases slightly better than the baseline even if they have not been trained on satellite images.

Stereo methods can be adapted to work as a stage of an existing satellite stereo pipeline. In this work we tested the GANet method as an alternative stereo matching step in the S2P pipeline. An interesting finding is that the results for the S2P-GANet variant were similar and in some cases better than the S2P baseline pipeline without specific training. The fine-tuning of GANet in more appropriate

datasets, such as WHU and RVL, showed a slight but consistent improvement in mean on all datasets used in the experiments. The improvement are more notorious in the (more challenging) far in time acquired images (JAX_FIT, OMA_FIT).

Regarding true MVS methods, Zhang et al. proposed in [89], an adaptation strategy and implemented it for classic methods such as COLMAP and Planesweep. The approach, as reported in their article, does not achieve better results than a pair-wise MVS based on S2P. Nevertheless, the approach is very interesting and can be applied to other methods, not intended initially for satellite imagery. We use that strategy to adapt the CasMVSNet method in this work. Table 3.4 shows that, although not outstanding, the metrics are close to the ones of pair-wise MVS. Results for CasMVSNet were obtained using a model pretrained on the DTU [44] dataset comprised of close range scenes.

Although the popularity of DL methods, they are still not the preferred option in satellite stereo pipelines. The results obtained with DL methods in this study show the potential of using this kind of algorithms on satellite images as a step in a classic pipeline or as an end-to-end MVS solution. Both tested DL methods exhibited a great generalization power in particular GANet. It is interesting to note that part of the internal structure of the GANet mimics SGM [42], which has been extensively used as the main aggregation strategy in several classic satellite stereo pipelines [50, 60, 73]. This fact, along with its generalization ability, points to this method as a really attractive option to include it in an existing satellite pipeline. The adaptation and enhancement of this and other methods to satellite images depends largely on the existence of aerial and satellite training datasets that are still scarce. The availability of more datasets such as WHU and RVL will surely trigger the adaptation of existing methods and the development of new methods for the benefit of the remote sensing community.

This page intentionally left blank.

Chapter 4

Enhancement of the disparity map

In a stereo pair of images, the change in viewpoint induces a relative displacement of the objects in the scene causing that closer objects move more than far ones in the images of the pair. This apparent motion between the two views called disparity is inversely proportional to the depth of the objects in the scene. As already mentioned, in order to reconstruct a scene, most stereo algorithms perform four steps: (1) matching cost computation, (2) cost aggregation, (3) disparity computation, (4) disparity refinement.

This chapter is related to the fourth step of post-processing or refinement of the disparity map. A method based on a filtering process guided by the values of the reference image of the stereo pair is analyzed and tested.

4.1. Introduction

The enhancenment of the disparity map is an optional step in a stereo matching algorithm that aims to fill-in missing values and filter-out erroneous values.

A similar problem is the completion of depth maps generated from sensors such as RGB-D structured light or time of flight cameras, Lidar, etc. In this modalities, missing values can be related to the characteristics of the imaged surfaces (for example specular surfaces and transparent surfaces usually pose difficulties to these kind of sensors).

Several algorithms for the post-processing of depth and disparity maps have been proposed in literature. An extensive survey is presented in [4], where the authors classify the approaches for still images in: a) filtering, interpolation, extrapolation methods, b) methods based on reconstruction, and c) inpainting based methods. Filtering methods are the most common approaches and are based on propagating local characteristics to interpolate or extrapolate the missing information. Examples of these approaches use anisotropic diffusion, bilateral filtering and other edge preserving filtering techniques. Inpainting methods are variants of the techniques traditionally used on color images that were adapted to the problem of disparity and depth filling. Methods based on reconstruction tackle the problem in a global fashion using typically a variational energy minimization approach or a model to recover the complete map.

Transversal to the multiple different approaches, [4] also distinguishes between the methods that fill the holes directly on the map from those that use the color or gray-level image (from the stereo pair or acquired by an RGB-D sensor) to guide the hole filling of the map.

4.2. Disparity map diffusion

The method analyzed and tested in this section is based on [26]. According to the above classification, this method can be considered a filtering of the disparity map guided by one of the images of the stereo pair. The code of the method used in the tests was developed by Sébastien Drouyer, author of [26]. The code was analyzed and documented, and a demo was developed in IPOL as part of this thesis. For the experiments, the method was inserted in the S2P pipeline as an additional step that modifies the disparity map before the triangulation step is performed.

4.2.1. Method Description

Consider a rectified stereo pair and a left-to-right (and right-to-left if available) disparity map computed by any stereo matching algorithm. The objective is to fill-in two kind of pixels in the disparity map:

- a) the pixels where the disparity is undefined;

- b) the pixels belonging to small regions with disparity values far away from their surrounding disparities (we will call these regions "speckles").

Missing disparities are usually due to occlusions and regions with poor texture cues in the stereo pair. Speckles can be originated by incorrect matching of repetitive structures and also by poor texture in the images. While missing disparities are objective targets to fill in, speckles are not. Speckles are merely "small" regions in the disparity "too different to be true" so a set of parameters quantifying these characteristics must be selected according to the type of stereo images. For example, in satellite stereo images of urban environments, these parameters could be selected taking into account the known scale of the images and maximum height of buildings.

The diffusion algorithm is organized in two main parts: The first part comprises the identification and removal of speckles. The disparity of the pixels identified as speckles is set to "undefined". In the second part, the undefined disparity pixels are filled-in by a diffusion process. In this process, guided by the reference image of the stereo pair, pixels in undefined disparity regions are filled with a weighted average of the disparities of neighboring regions.

The algorithm can work on the left stereo image and the left-to-right disparity map or, if available, also the right stereo image and the right-to-left disparity map. The latter enables a left-to-right consistency check useful to invalidate occlusions and eventual inconsistent matches introduced by the diffusion process.

Speckle identification

The speckles can be identified as the regions with an area less than a certain threshold $area\_th$ and a difference with respect to neighboring regions greater than another threshold $discontinuity\_th$. Pixels identified as belonging to a speckle are set to "undefined".

Disparity map diffusion

In this stage, all the disparity pixels that are "undefined" get filled.

First, a partition of the left (or right) stereo image is computed giving a set of approximately uniform regions. In order to compute a partition, a local gradient of the reference stereo image is computed. Local minima of the gradient are used as seeds for a watershed segmentation to finally get a partition of the stereo image. The regions in the partition guide the diffusion of the disparity map ensuring a map consistent with the gray values of the stereo image. In a diffusion step, the disparity pixels corresponding to a region get a value computed as a weighted average of the disparity values of neighboring regions. The weights are determined by the difference between neighboring regions in the stereo image.

4.2.2. Left-right consistency check

When a right-to-left disparity map is available, the algorithm can be applied on both disparity maps. Then a consistency check can be performed to ensure

that correspondent disparities in both maps differ in less than a desired threshold $left\_right\_consistency\_th$.

4.3. Algorithm

4.3.1. Main Body

Algorithm 1: Main body of the algorithm

input : Stereo pair: **l_img**[, **r_img**]
Disparity maps: **lr_map**[, **rl_map**]
output: Processed disparity maps: **out_lr_map**[, **out_rl_map**]

1 **out_lr_map**= $remove\_speckles$(**lr_map**)
2 **out_lr_map**= $diffuse$(**out_lr_map**)

Algorithm 1 presents the main body of the algorithm. The algorithm requires as inputs the left image of the rectified stereo pair and the left-to-right disparity map plus parameters. If the right-to-left disparity map is available, the algorithm can accept it along with the right image of the rectified stereo pair.

In the first step, speckles are removed from the left-to-right disparity map. Then the undefined regions in the disparity map are filled by a diffusion process.

If the right image and right-to-left disparity map are supplied, steps 3 to 5 are executed. Steps 3 and for 4 are equivalent to the first two steps but applied on the right image and right-to-left disparity map. Step 5 enforces the consistency between the two processed disparity maps.

4.3.2. Speckle Removal

Algorithm 2: Speckle removal

input : Disparity map: **d_map**
Parameters: **discontinuity_th**, **area_th**
output : Processed disparity map: **out_d_map**

1 **d_map_gradient** = $compute\_morphological\_image\_gradient$ (**d_map**)
2 **candidate_speckle_boundaries** = $d\_map\_gradient$[**d_map_gradient** > **discontinuity_th**]
3 **candidate_speckle_regions** = $fill\_from\_boundaries$ (**candidate_speckle_boundaries**)
4 **speckle_regions** = $filter\_regions\_by\_area$ (**candidate_speckle_regions**, **area_th**)
5 **out_d_map**= $set\_to\_undefined$(**d_map**, **speckle_regions**)

Algorithm 2 presents the pseudocode for the removal of speckle regions in the disparity map.

- Line 1: A morphological image gradient is computed on the original disparity map.
- Line 2: Candidate boundaries for the speckles are computed according to the gradient. The **discontinuity_th** controls how much change in disparity is considered a speckle.

4.3.3. Diffusion of the disparity map

Algorithm 3 presents the pseudocode for the diffusion of the disparity map. The procedure fills all the undefined pixels in the disparity map.

Algorithm 3: Disparity map diffusion

input : Disparity map
and stereo image: **d_map**, **img** (**lr_map**, **l_img** *or* **rl_map**, **r_img**)
Parameters: **local_gradient_window**, **h_minima_th**, **min_absortion_weight**,
gaussian_alpha, **gaussian_beta**,
defined_disparity_ratio
output: Processed disparity map: **out_d_map**

1 **img_gradient**= *compute_local_gradient* (**img**, **local_gradient_window**)
2 **img_partition**= *watershed_segmentation* (**img**, **img_gradient**, **h_minima_th**)
3 **disparity_means**, **undefined_disparity_ratio**= *compute_disparity_stats* (**img_partition**, **d_map**)
4 **adjacency_graph**= *build_adjacency_graph* (**img_partition**, **img_gradient**, **min_absortion_weight**, **gaussian_alpha**, **gaussian_beta**)
5 **diffused_img_partition**= $diffuse\_disparity\_values$ (**adjacency_graph**)
6 **out_d_map**= *build_map*(**diffused_img_partition**, **img_partition**)

- Line 1: For each location in the input image, a local gradient is computed in a neighborhood determined by the parameter **local_gradient_window**.
- Line 2: The reference image from the stereo pair is segmented by a watershed algorithm [6]. In the watershed algorithm, the gray-level image is viewed as a topographic surface and the watershed lines are the contours that divide water basins in the surface terrain. The surface is flooded from its minima or from a predefined set of markers and the waters from different sources are prevented to merge by barriers. The water barriers can be determined, for

example, by computing the gradient of the image. The final disjoint water regions should correspond to approximately homogeneous gray level regions of the image.

In the implementation, the markers to start the flooding are determined by the h_minima [79] of the precomputed local gradient. These are connected sets of pixels with gradient level strictly smaller (minimum difference controlled by the parameter **h_minima_th**) than the gradient levels of all pixels in direct neighborhood of the set. The watershed algorithm is run on the gradient image starting from the set of markers.

- Line 3: On each region of the image partition two quantities are computed: a) the mean disparity of the region without considering the undefined values (**disparity_means**), and b) the proportion [number of pixels with undefined disparity over the number of pixels of the region] (**undefined_disparity_ratio**).

- Line 4: The regions from the image partition are arranged in a bidirectional weighted graph where the edges join neighboring regions. The weight of an edge from a region A to a region B reflects the "diffusiveness" between the regions. This quantity is computed as the minimum value of the gradient (computed in step 1) along the piece of contour shared by the two regions. Hence, the diffusion process will be favored between regions with small gray value differences and disfavored between regions separated by steep borders. The outgoing edges of each region are normalized to $[0, 1]$. The pseudocode of the adjacency graph is presented in Algorithm 4.

- Line 5: A region that has a **undefined_disparity_ratio** $<$ **defined_disparity_ratio** is considered "fixed". All the pixels get the mean disparity value of the region. A region with **undefined_disparity_ratio** $<$ **defined_disparity_ratio** has a proportion of undefined disparity pixels above the threshold. The disparity value of the region is computed as the weighted average sum of mean disparities of its adjacent regions.

Algorithm 4: Adjacency graph

input : Disparity map
and partition: **d_map**, **img_partition**,
Parameters: **min_absortion_weight**,
gaussian_alpha, **gaussian_beta**,
defined_disparity_ratio
output: Graph: **edges**, **weights**

```
1  edges= get_adjancent_regions (img_partition)
2  min_gradients= compute_min_gradient_between_regions(edges,
     img_partition, img_gradient)
3  w = min_gradients
4  for each region A in img_partition do
5  |  w_min_A = mín_{B adjacent to A} w(A,B)
6  |  for each region B adjacent to A do
7  |  |  w(A,B) = w(A,B) - w_min_A
8  |  |  w(A,B) = exp^(−0,5∗(w(A,B)/sigma)^2)
9  |  |           with sigma = (w_min_A + gaussian_alpha) /
   |  |  gaussian_beta
10 |  |  if  w(A,B) < min_absortion_weight then
11 |  |  |  w(A,B) = min_absortion_weight
12 |  |  end
13 |  end
14 |  weights(A,B) = w(A,B) / ∑_B(w(A,B))
15 end
```

4.3.4. Consistency check

Algorithm 5: Check left right consistency

input : Disparity maps: **lr_map**, **rl_map**
Parameters: **consistency_th**
output : Checked disparity maps: **out_lr_map**, **out_rl_map**

```
1  out_lr_map= lr_map
2  out_rl_map= rl_map
3  for each pixel [x_left,y_left] in lr_map do
4  |   x_right= x_left+ lr_map[x_left,y_left]
5  |   y_right= y_left
6  |   x_left'= x_right+ rl_map[x_right,y_right]
7  |   if abs( x_left'-x_left) > consistency_th) then
8  |   |   set_to_undefined (out_lr_map[x_left,y_left])
9  |   end
10 end
11 for each pixel [x_right,y_right] in rl_map do
12 |   x_left= x_right+ rl_map[x_right,y_right]
13 |   y_left= y_right
14 |   x_right'= x_left+ lr_map[x_left,y_left]
15 |   if abs( x_left'-x_left) > consistency_th) then
16 |   |   set_to_undefined (out_rl_map[x_right,y_right])
17 |   end
18 end
```

Algorithm 5 presents the pseudocode for the consistency check between the left-to-right and right-to-left disparity map. The disparity is consistent in a certain location if, going from left to right via the **lr_map** (right to left via the **rl_map**) and back from right to left (left to right), the resulting location column differs from the original location column in less than the parameter **consistency_th**.

4.3.5. Parameters

The speckle removal 2 depends on two parameters as shown in table 4.1. These parameters are problem-specific and must be selected according to the scene and the characteristics of the disparity images.

The diffusion in algorithm 3 is controlled by the parameters listed in table 4.2. A local gradient (of the reference or secondary stereo image) is computed in a region of radius *gradient_size* and the local minima of depth *h_minima* are computed on the gradient.

These local minima are used as seeds for a watershed segmentation. A local minimum M of depth h is a local minimum for which there is at least one path joining M with a deeper minimum on which the maximal value of the path is $(f(M)+h)$ [79]. The default values imply a 3×3 neighborhood for computing the local gradient and that every local minimum of the gradient is considered.

Table 4.1: Parameters for the speckle removal

Parameter	Default value	Units	Description
discontinuity_th	10	image levels	How much change in disparity is considered a discontinuity
area_th	200	pixels	Maximum area of a connected component to be considered a speckle

The watershed segmentation gives a partition of the stereo image. This partition is used to analyze the disparity map. In each region of the partition there are pixels where the disparity map has a value and also pixels where the disparity is not defined. A region is considered "fixed" if the proportion of pixels with defined disparity is above **undefined_disparity_ratio**. A fixed region is not updated during the diffusion. The default value for **undefined_disparity_ratio** is 0,5.

Table 4.2: Parameters for the diffusion

Parameter	Default value	Units	Description
gradient_size	1	pixels	Radius of the region where the local gradient is computed
h_minima	1	pixels	Depth for the local minima
defined_threshold_ratio	0.5	-	For each region, what is the minimum area ratio that must be defined in the disparity map so that the region is fixed

The diffusion is implemented via an adjacency graph. In each iteration of the diffusion process, the disparity of non-fixed regions is updated as the weighted sum of the disparities of the adjacent fixed regions. The weights of the graph are computed according to algorithm 4. The parameters for computing the weights are listed in table 4.3.

gaussian_alpha, gaussian_beta The weight of an edge $A \longrightarrow B$ is related to the minimum of the gradient (gradient of the reference or secondary image) along the contour separating regions A and B. The weights are scaled by a gaussian where the sigma is controlled by the parameters **gaussian_alpha** and **gaussian_beta** (see line 9 in algorithm 4). Given a region A, if one of the neighboring regions is similar to A then $w\_min\_A \sim 0$ and $sigma \sim$

Table 4.3: Parameters for the adjacency weighted graph

Parameter	Default value	Units	Description
min_absortion	1E-5	-	Minimum weight affected to a connection between neighboring regions
weights_alpha	10	-	Controls the *sigma* of the gaussian bell that scales the weights
weights_beta	5	-	Controls the *sigma* of the gaussian bell that scales the weights

gaussian_alpha/*gaussian_beta*. Then with the default values, $sigma \sim 2$ (narrow bell) and the update of A will be mostly influenced by similar regions. On the other hand, if A is surrounded by regions that are not similar to A, the value of w_min_A will be higher and *sigma* will raise accordingly giving a wider bell. In brief, *gaussian_alpha*/*gaussian_beta* controls the minimum *sigma* while *gaussian_beta* controls the maximum *sigma*. Higher values of *gaussian_beta* imply that the update takes into account only the most similar regions. Lower values of *gaussian_beta* entail an update with the contribution of more regions.

min_absortion_weight A non-null value implies that every edge has a minimum weight and hence, all the adjacent regions have a base contribution in the weighted sum.

4.3.6. Experiments

Experiments were conducted on the stereo satellite images from the MVS3D dataset [10]. The baseline for the disparity computation is given by the S2P pipeline [19].

For each considered stereo pair, the disparity is computed with the MGM stereo matching algorithm [28] from the S2P pipeline (we will call it the baseline disparity) and then processed with the presented algorithm (we will call it the processed disparity).

In satellite imagery, the disparity range of a stereo pair is dependent on the position and attitude of the satellite in each acquisition. This makes difficult the comparison in disparity error across different stereo pairs. The comparison on altitude can give a better idea of the performance of the algorithm in these kind of images. In these experiments, the altitudes triangulated from the baseline and processed disparity maps are compared against the altitude data from the airborne Lidar that is considered the ground-truth.

In order to compute the disparity of a stereo pair, the S2P pipeline partitions the reference and secondary images of the pair in $M \times N$ tiles. For the case of these experiments, $M = N = 4$ and then there will be 16 tiles per stereo pair and

the corresponding 16 disparity and altitude images.

Figure 4.1 shows a typical stereo pair from a tile that covers part of the scene.

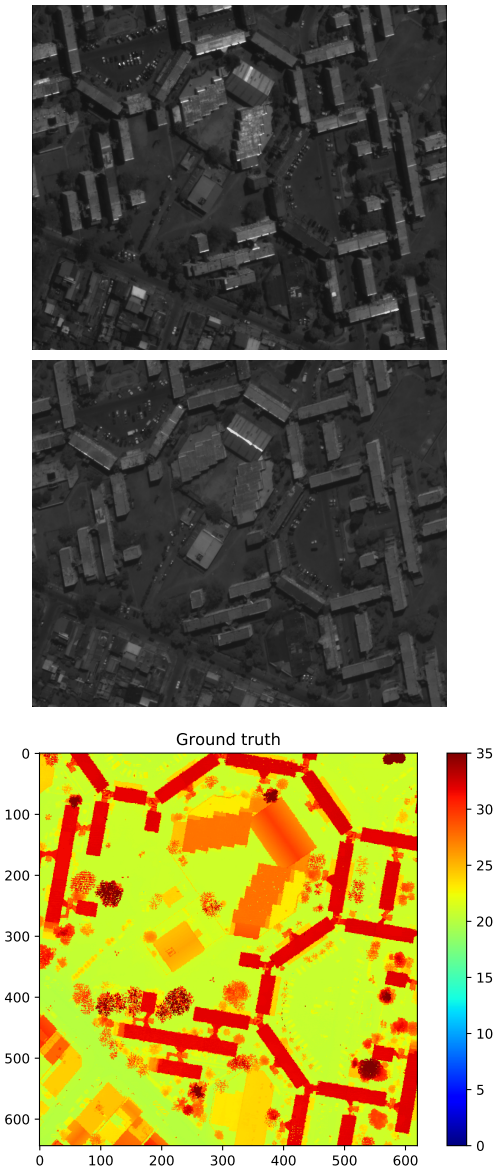

Figure 4.1: A stereo pair tile from pair of satellite images number 33 and 32 of the MVS3D dataset and the ground truth of the same region.

Benchmark

In order to measure the results of the algorithm, the altitudes computed from the baseline disparity map and from the processed disparity map are compared against the ground truth. Each altitude map is registered to the ground truth Lidar data and the following metrics are computed: EVA, BAD, INV, COMP, AAE.

Figure 4.2 presents the altitude computed from the baseline and the processed disparity maps for four pairs on the same region of tile 5. The metrics for this cases can be found in table 4.4. The metrics show that a proportion of the pixels with undefined altitude for the baseline get filled-in with the algorithm (Invalid decreases) but not all the newly created altitudes are correct (Bad-Z increases). In pairs 29-28 and 33-32, the completeness metric has an increment that shows that, in these cases, a significant proportion of the undefined pixels get a good altitude value. However, bad values can be propagated by the diffusion as seen in Figure 4.2 for the pair 29-28. Figure 4.3 graphically shows for pair 33-32 this results comparing the pixel altitude categories (invalid, bad, good) for the baseline and the diffusion algorithm.

The completeness metric does not increase for pairs 39-38 and 40-39 after the diffusion. Note that these two pairs have a low initial proportion of invalid pixels different from the other two pairs.

Table 4.5 presents the difference in metrics (algorithm-baseline) averaged over all the tiles of the satellite image pairs.

Table 4.4: Benchmark for the different pairs on tile 5.

Pair	Method	Iterations	BAD	INV	COMP	AAE(m)
29-28	Baseline	-	0.212	0.265	0.524	1.771
29-28	Algorithm	1	0.221	0.224	0.555	1.780
29-28	Algorithm	3	0.256	0.172	0.572	1.853
29-28	Algorithm	5	0.256	0.170	0.574	1.851
33-32	Baseline	-	0.172	0.272	0.557	1.347
33-32	Algorithm	1	0.189	0.221	0.590	1.353
33-32	Algorithm	3	0.204	0.189	0.607	1.413
33-32	Algorithm	5	0.205	0.188	0.607	1.413
39-38	Baseline	-	0.436	0.060	0.504	1.544
39-38	Algorithm	1	0.472	0.044	0.484	1.615
39-38	Algorithm	3	0.473	0.041	0.486	1.623
39-38	Algorithm	5	0.473	0.041	0.486	1.623
40-39	Baseline	-	0.504	0.069	0.428	1.703
40-39	Algorithm	1	0.557	0.049	0.394	1.799
40-39	Algorithm	3	0.558	0.047	0.396	1.804
40-39	Algorithm	5	0.558	0.047	0.396	1.804

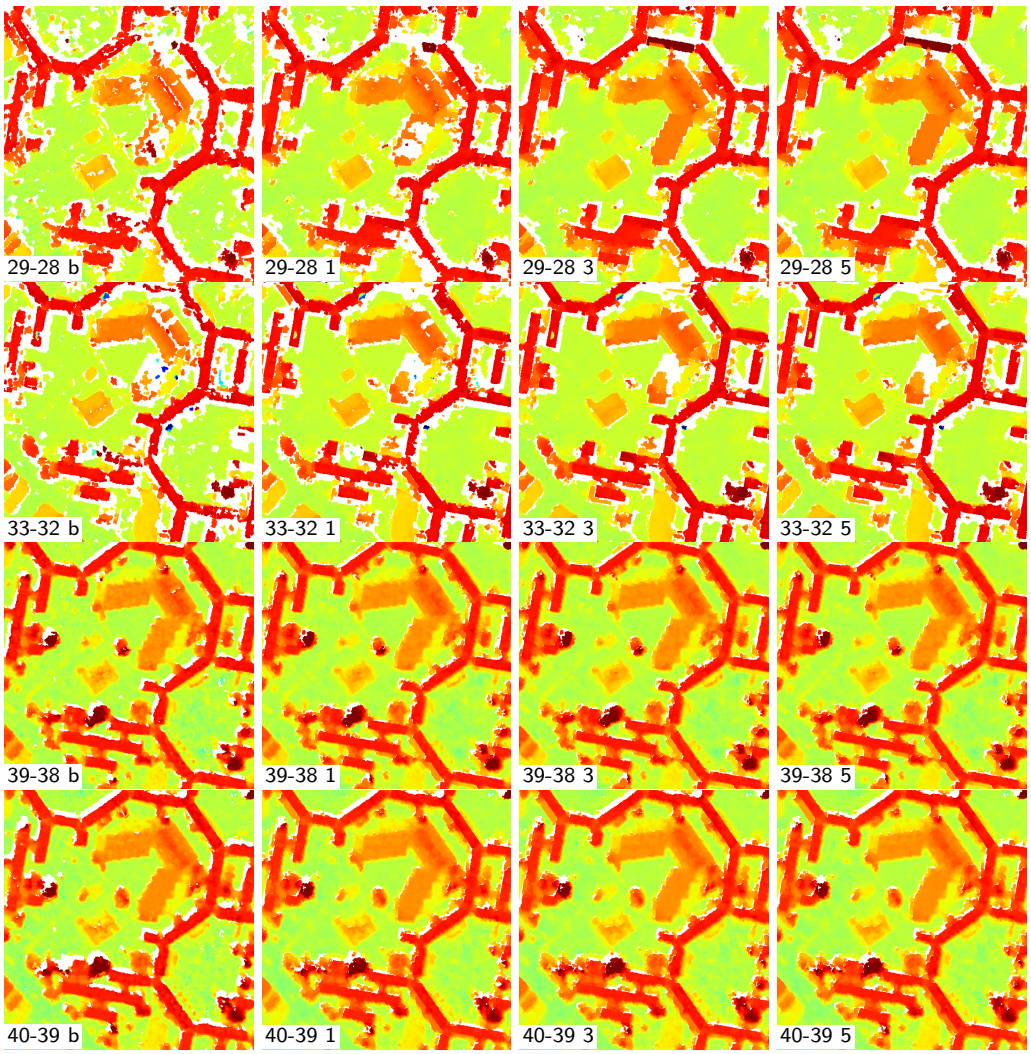

Figure 4.2: Result for the different pairs on the same region of tile 5. From top to bottom, rows correspond to pairs 28-29, 33-32, 39-38 and 40-39. From left to right, columns correspond to baseline and iterations 1, 3 and 5 of the algorithm.

Table 4.5: Difference in metrics(algorithm-baseline) averaged over all the tiles of the satellite image pairs.

		Metric Difference: Algorithm – Baseline (mean ± std)			
Pair	Iter.	BAD	INV	COMP	AAE(m)
29-28	1	0.020 ± 0.020	-0.035 ± 0.021	0.014 ± 0.017	0.041 ± 0.051
29-28	3	0.033 ± 0.026	-0.054 ± 0.032	0.021 ± 0.020	0.063 ± 0.061
29-28	5	0.034 ± 0.027	-0.054 ± 0.033	0.021 ± 0.020	0.065 ± 0.062
33-32	1	0.024 ± 0.016	-0.044 ± 0.012	0.020 ± 0.015	0.091 ± 0.100
33-32	3	0.036 ± 0.021	-0.063 ± 0.019	0.027 ± 0.017	0.125 ± 0.109
33-32	5	0.037 ± 0.022	-0.064 ± 0.019	0.027 ± 0.017	0.128 ± 0.110
39-38	1	0.014 ± 0.018	-0.014 ± 0.007	0.000 ± 0.018	0.015 ± 0.037
39-38	3	0.016 ± 0.020	-0.016 ± 0.007	0.000 ± 0.019	0.020 ± 0.041
39-38	5	0.016 ± 0.020	-0.016 ± 0.007	0.000 ± 0.019	0.020 ± 0.042
40-39	1	0.015 ± 0.018	-0.014 ± 0.007	-0.002 ± 0.020	0.019 ± 0.039
40-39	3	0.018 ± 0.019	-0.016 ± 0.007	-0.002 ± 0.019	0.027 ± 0.042
40-39	5	0.018 ± 0.019	-0.016 ± 0.007	-0.002 ± 0.019	0.027 ± 0.042

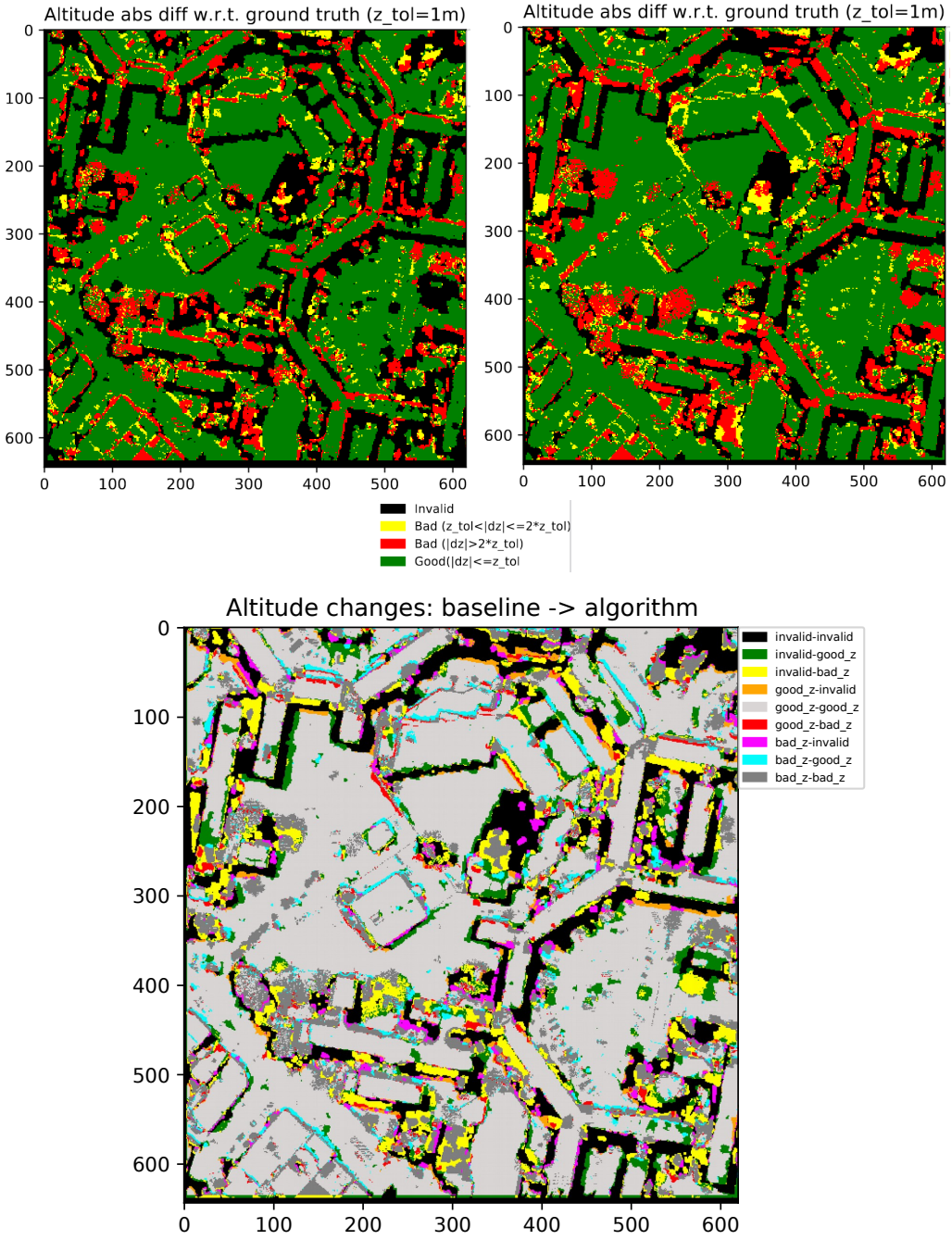

Figure 4.3: Comparison of the altitudes computed from the baseline disparity map and from the processed disparity map. Top: Altitude absolute difference with respect to ground truth. Top left: Baseline, Top right: Diffusion algorithm with five iteration steps.
Bottom: Altitude change between the altitude computed from the baseline disparity map and the altitude computed from the processed disparity map. For example, the pixels in green are the ones with undefined altitude for the baseline that have been filled-in with a good altitude ($|dz| <= z\_tol$) with the diffusion algorithm while yellow pixels have been filled-in with a bad altitude ($|dz| > z\_tol$).

4.4. Conclusion

The diffusion of the disparity map guided by the images of the stereo pair is a plausible filling strategy. It enhances the definition of structures that have contrasted values in the images of the stereo pair. On the downside, the method seems dependent of the initial conditions of the disparity map and cannot ensure to always perform an enhancement in terms of the completeness. Anyway, the algorithm seems an interesting option to be applied in controlled conditions.

Chapter 5

Simulation of images and RPCs

This chapter presents a simulation tool that allows to produce views of a 3D scene from multiple orientations generating images along with RPC models suitable for a stereo pipeline. A stereo reconstruction with those images can be assessed by comparing to the known altitude of the scene.

In this thesis, the tool is used to sample pairs from many geometric configurations on the hemisphere surrounding an artificial scene. The reconstruction quality can be assessed for each pair and can be used as a measure for pair selection as will be seen in the next chapter.

One of the main problems that bedevils the advance in MVS for satellite imagery is the scarce public datasets of satellite images with well curated ground-truth altitude. That causes that the adaptation of recent algorithms that could be beneficial to the remote sensing community is slow as the necessary data for training and fine-tuning is still insufficient. The simulation tool could be used to mitigate this issue as it can help generate images from realistic scenes that can complement the few available real satellite image datasets.

5.1. Introduction

The development and testing of MVS algorithms requires the availability of extensive datasets with ground truth information. While there are massive datasets for close range stereo processing and benchmarking, there are not many satellite image datasets with ground truth elevation. Such satellite datasets are hard to produce as it requires the coordination and effort of many actors; it has been done, for example, in the case of some challenges [9,10]. The simulation of satellite images can be an alternative or a complement to generate more data with ground truth. This can facilitate the development and testing of algorithms of a classic pipeline, or the training and testing of new machine learning methods for MVS.

In this work we use simulated optical satellite images to analyze and evaluate the performance of a pair-wise MVS pipeline. For this aim, a tool was developed that allows to simulate images along with rational polynomial coefficients (RPC) camera models.

5.2. Image and RPC simulation tool

Starting from a pre-built 3D scene, the longitude and latitude coordinates of the scene center and the orientation of the view, the simulation tool generates an image and a corresponding RPC camera model suitable to be used in a satellite stereo pipeline. The simulator uses Blender [15] as the 3D engine to render the views. Blender is launched and configured automatically through Python scripts.

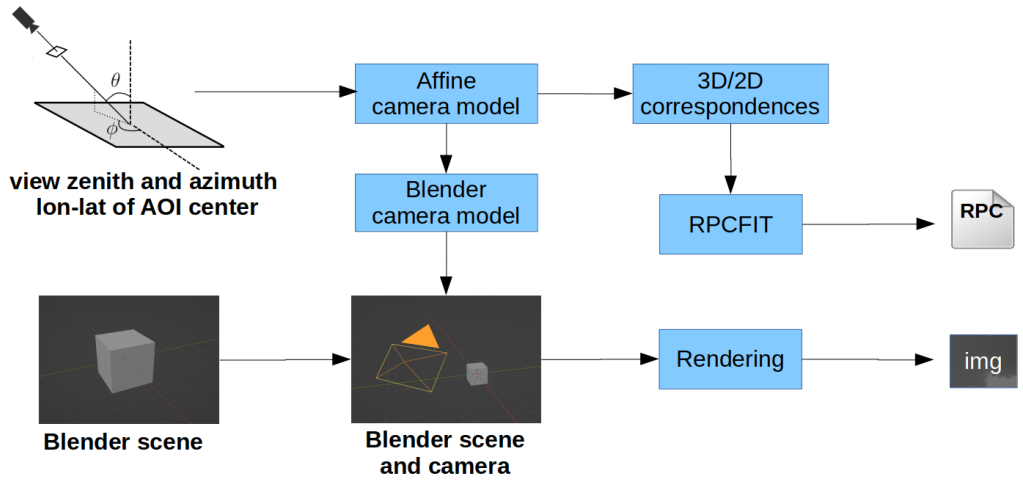

Figure 5.1: Block diagram of the simulator. Please refer to the text for the description of the blocks and the flow of data.

Figure 5.1 presents a block diagram of the simulation tool. Given a scene and a view direction, an affine camera model is determined. The affine camera model is a sensible approximation of a real satellite projection for a small area of interest (AOI) [20]. This model gives corresponding 3D/2D coordinates between

the volume of interest (AOI plus height range) and the image. The correspondences are then used to adjust an RPC camera model using the RPCFIT tool [2], which fits an RPC model to the 3D/2D correspondences through a regularized least squares minimization.

In order to render the image of the scene, a camera model, compatible with the affine camera model, is created in Blender. Figure 5.2 shows examples of the simulation tool with two different scenes: a simple one with a cylinder on a flat surface, and an artificial urban scene.[1] The simulation tool is available at `https://github.com/zemogoravla/simsatool`.

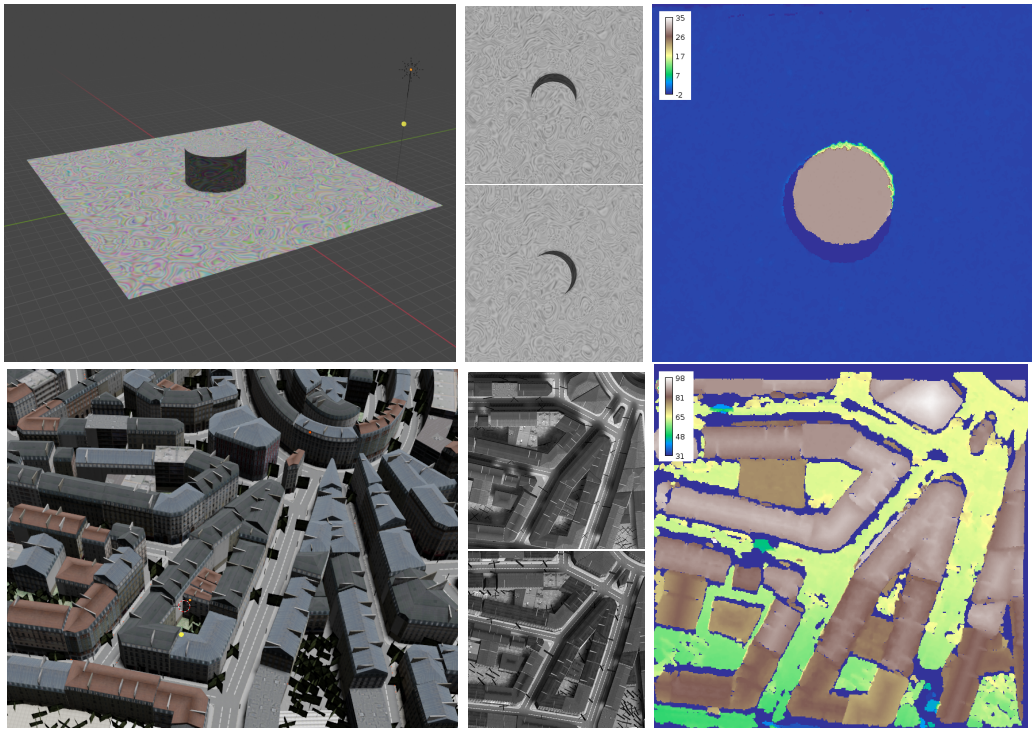

Figure 5.2: Results of the simulation tool with two different 3D scenes. From left to right: a view of the 3D scene, a stereo pair generated with the simulation tool, and the DSM reconstructed with the S2P pipeline. Above: Cylinder scene. The scene is composed of a cylinder with a radius of 25m and a height of 30m. Surfaces have a random texture. Below: Artificial city scene[1].

5.2.1. Sun orientation

The simulator accepts a sun orientation. The sun position can also be specified by the location and the date/time of acquisition. Figure 5.3 shows examples of images generated with different sun orientations.

[1] Urban scene downloaded from https://open3dmodel.com/. Accessed on October 2022.

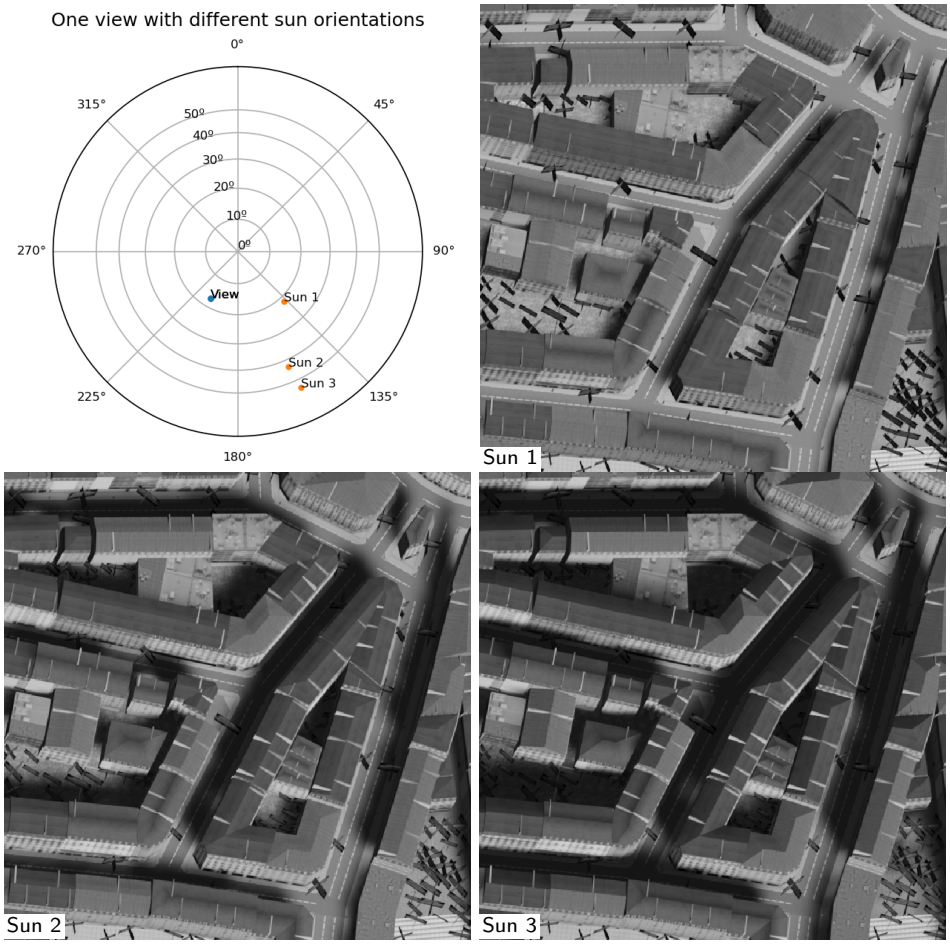

Figure 5.3: Simulated images of the city scene (see 3D scene in Figure 5.2). The views have the same orientation and different sun positions.

5.2.2. Image and noise levels of the images

The image and noise levels for a simulated image can be adjusted and approximately matched to a reference real image. This enables to generate images that better resemble concrete real images.

Figure 5.4 shows an example of a simulated image whose values are matched to the ones of a reference image taken from the MVS3D dataset. The histogram matching [36] is a simple technique to get a monotonic mapping between a pair of histograms. The monotonic mapping of the pixel values preserves the contrast from the original image.

The noise in the simulated images can also be adjusted to approximately match the noise levels in the reference image. This can be done by the method devised by Ponomarenko et al. in [67] and implemented and extended in [14]. This method estimates a noise curve from a single image. The image is analyzed by blocks that are binned according to their mean and standard deviation values. The use of multiple bins allows to analyze signal-dependent noise (e.g. Poisson model) and

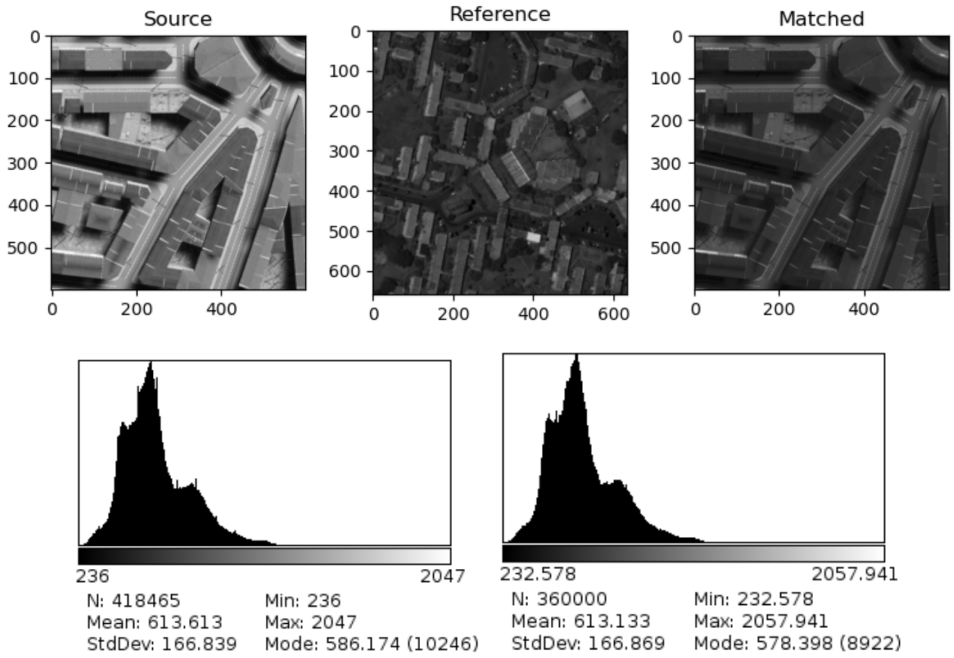

Figure 5.4: Histogram matching. Top left: output image from the simulator, Top center: Reference image from MVS3D dataset, Top right: Image from the simulation with histogram matched to the reference image. Bottom: reference and matched histograms.

get a curve of noise level (as the standard deviation) versus image level. With the estimation of a noise curve for the reference image, the noise in the simulated (matched to reference) image can be added in such a way to get an approximately similar noise curve. In general, a perfect match cannot be achieved but the range of noise can be fairly adapted to similar ranges.

Figure 5.5 shows an example of approximate noise matching where the noise in the reference image is estimated by the Ponomarenko method and the noise in the simulated image is adjusted to have similar levels as the reference.

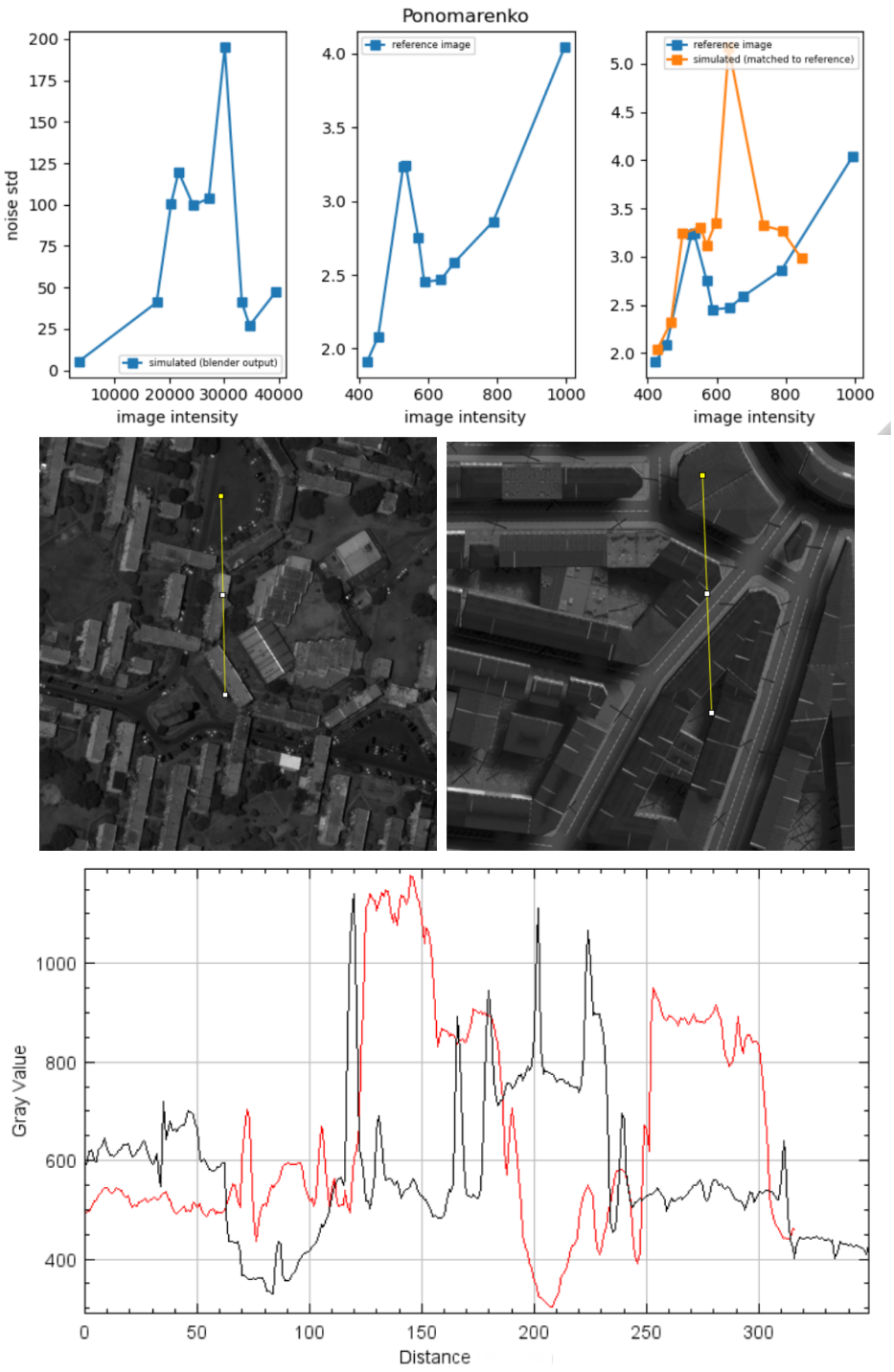

Figure 5.5: Noise matching using the Ponomarenko estimation. Top left: noise curve of the output image from the simulator. Top center: noise curve of the reference image. Top right: Comparison of noise curves after adjusting the noise level in the simulated image. Middle: Profile segments on reference image and matched source image. Bottom: Plot of image values on the profile segments of the reference image (in red) and matched source image (in black). The reference and matched images present similar ranges for signal and noise.

5.3. Examples of use

5.3.1. One view with different sun orientations

Images in Figure 5.3 correspond to same view with different sun orientations. They are generated with the code in Listing 5.1.

The orientation of a view and the location of the sun are expressed both in terms of azimuth and zenith angles in degrees. Azimuth angle is the number of degrees rotating east from north. The azimuth can be expressed as a positive number of degrees ranging from 0 to 360 where 0^{o} corresponds to north, 90^{o} to east and so forth. Zenith angle is the number of degrees down from the vertical direction. The zenith is expressed as a positive number of degrees ranging from 0 to 90 where 0^{o} corresponds to overhead and 90^{o} to the horizon.

```
from simulator import Simulator
from satellite import Satellite
from location import Location
from blender import Blender

#%% DIRECTORIES AND FILENAMES ------------------------------
# The simulation results will be created under this directory
simulation_base_dir = 'data/SIMULATION_EXAMPLE_SUNPOS'

# Select the 3D scene
blender_scene_filename ='data/model/city_scene.blend'

#%% VIEWS AND SUN ORIENTATION ------------------------------
# All angles are in degrees
ref_zenith_list =  [17, 17, 17]
ref_azimuth_list = [210, 210, 210]
ref_sun_zenith_list =[21.74, 42.98, 54.23]
ref_sun_azimuth_list = [137.21, 156.32, 155.07]

#%% SETUP THE SIMULATOR  -----------------------------------
satellite = Satellite()
blender = Blender(blender_scene_filename,
                  image_xy_size=(600,600))
location = Location()
sim = Simulator(simulation_base_dir,
                satellite, blender, location)

#%% GENERATE THE IMAGES ------------------------------------
for i in range(len(ref_zenith_list)):
  ref_image_filename, ref_rpc_filename = \
    sim.simulate_image_and_rpcfit(ref_zenith_list[i],
                                  ref_azimuth_list[i],
                                  None,
                                  ref_sun_zenith_list[i],
                                  ref_sun_azimuth_list[i])
```

Listing 5.1: Generation of images of Figure 5.3

5.3.2. A stereo pair to be reconstructed with S2P

Listing 5.2 presents an example code to generate a stereo pair of images and an S2P configuration for the stereo reconstruction of the pair. The values and noise in the images of the pair are approximately matched to the values and noise of a reference image as done in Section 5.2.2.

Figure 5.6 shows the generated pair of images. Figure 5.7 shows the ground truth DSM and the reconstructed DSM.

```
import os
from simulator import Simulator
from satellite import Satellite
from location import Location
from blender import Blender
from s2p_configurator import S2PConfigurator

#%% DIRECTORIES AND FILENAMES -------------------------------
# The simulation results will be created under this directory
simulation_base_dir = 'data/SIMULATION_EXAMPLE_STEREO_PAIR'

# Select the 3D scene
blender_scene_filename ='data/model/city_scene.blend'

# Each generated image can be approximately matched in values
# and noise to a different reference image.
# Here we use a single reference image for both images of the
# pair.
target_image_filename = 'data/images/IARPA_15DEC18140510.tif'

# The configuration to run the stereo reconstruction
# will be created under:
# <simulation_base_dir>/S2P_CONFIGS
s2p_configs_dir = os.path.join(simulation_base_dir,
                               'S2P_CONFIGS')

#%% VIEWS AND SUN ORIENTATION -------------------------------
# All angles are in degrees
# Generate a pair of images [(ref)erence - (sec)ondary]
ref_zenith = 5
ref_azimuth = 0
ref_sun_zenith = 35
ref_sun_azimuth = 156
# ---------------
sec_zenith = 10
sec_azimuth = 210
sec_sun_zenith = 21
sec_sun_azimuth = 137

#%% SETUP THE SIMULATOR  ------------------------------------
satellite = Satellite()
blender = Blender(blender_scene_filename,
```

```
                    image_xy_size=(600,600))
location = Location()
sim = Simulator(simulation_base_dir,
                satellite, blender, location)

#%% SETUP THE S2P CONFIGURATOR  ------------------------------
s2p_configurator = S2PConfigurator(s2p_configs_dir)

#%% GENERATE THE IMAGE PAIR AND S2P CONFIGURATION -----------

ref_image_filename, ref_rpc_filename = \
        sim.simulate_image_and_rpcfit(ref_zenith,
                                      ref_azimuth, None,
                                      ref_sun_zenith,
                                      ref_sun_azimuth,
                                      target_image_filename)

sec_image_filename, sec_rpc_filename = \
        sim.simulate_image_and_rpcfit(sec_zenith,
                                      sec_azimuth, None,
                                      sec_sun_zenith,
                                      sec_sun_azimuth,
                                      target_image_filename)

s2p_config_filename = \
        s2p_configurator.create_config(ref_image_filename,
                                       ref_rpc_filename,
                                       sec_image_filename,
                                       sec_rpc_filename)
```

Listing 5.2: Generation of a stereo pair and an S2P configuration.

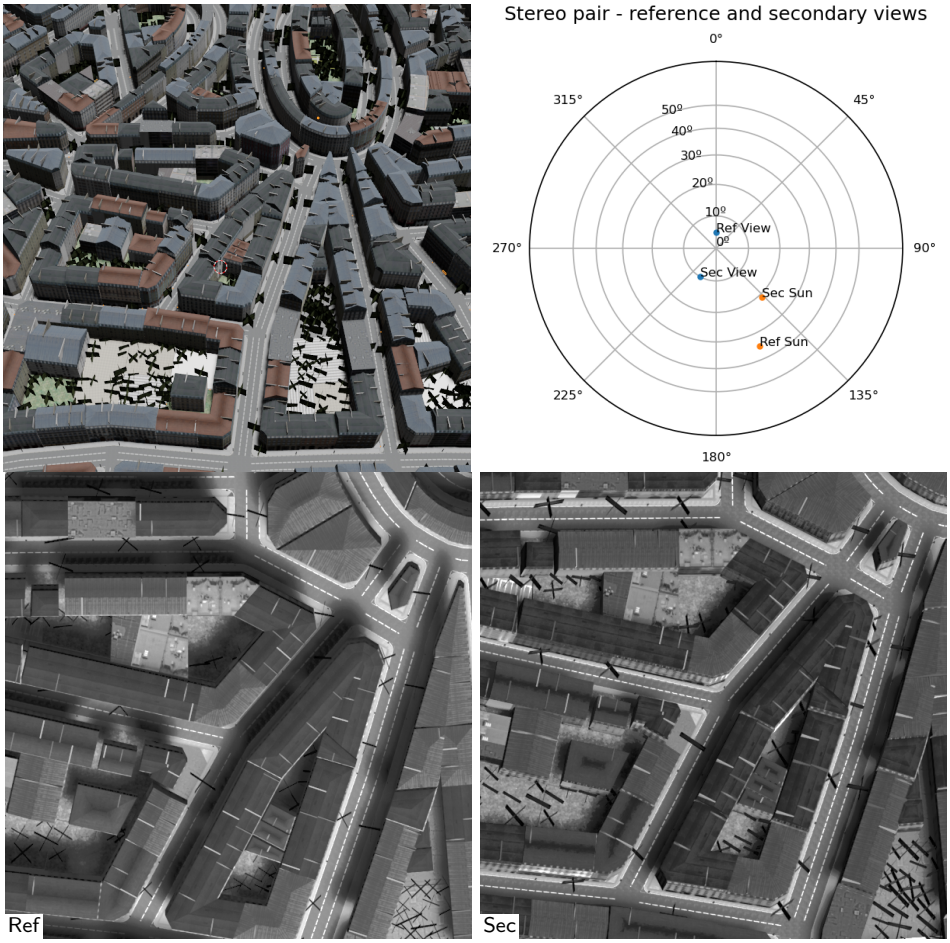

Figure 5.6: Generated image pair. Top: the 3D scene and the orientations of the views and sun positions in the stereo pair. Bottom: Reference and secondary images of the simulated pair.

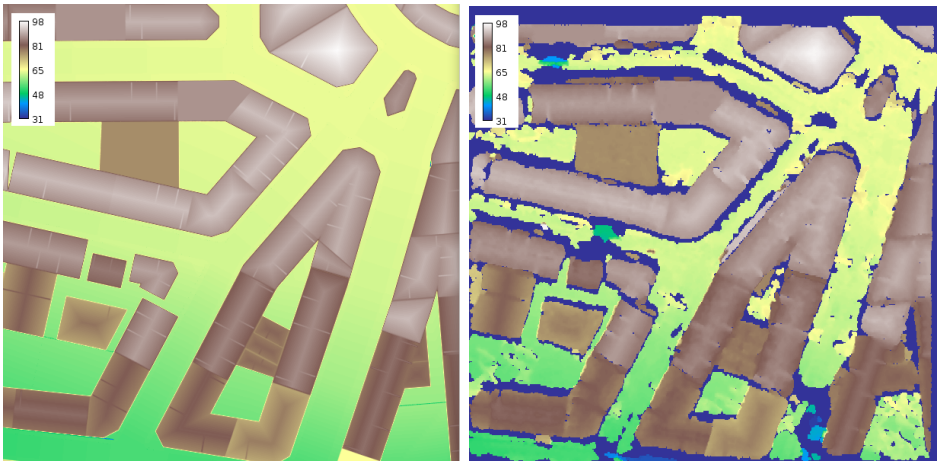

Figure 5.7: Left: Ground truth DSM of the city scene. Right: DSM reconstructed with S2P from the simulated pair of images.

This page intentionally left blank.

Chapter 6

Pair selection and model fusion

In pair-wise MVS, given a set of images of a scene, the reconstruction of the scene can be obtained by fusing the information of the DSMs computed from multiple pairs drawn from the set of images. The quality of the final reconstruction depends on the quality of the pair-wise reconstructions and on the method adopted for the fusion.

The quality of DSMs reconstructed from stereo pairs depend on several factors such as the orientations of the views in a pair and changes in the conditions between the acquisition of the images. The selection of good pairs from the set is a difficult task that has been traditionally tackled by designing heuristics on the metadata of the images. An alternative selection method based on the simulation of images was devised and evaluated on this thesis. The method maps the relation between the orientation of the views and the reconstruction quality through simulation.

A simulation tool was implemented that allows to produce views of a 3D scene from multiple orientations generating images along with RPC models suitable for a pair-wise stereo pipeline. The stereo reconstruction from a pair of simulated images can be assessed by comparing to the known altitude of the scene. Synthetic stereo pairs are simulated under all possible geometric configurations on the hemisphere surrounding an artificial scene and the stereo reconstruction quality can be assessed for each pair. This pre-computed quality is then used as a proxy for the quality of real pairs of images.

On the other end of the pipeline, a method to fuse the DSMs based on an iterated bilateral filter is proposed. The method robustly integrates the DSMs imposing local coherence on altitude and color. The iterative scheme, with progressively more restrictive ranges, allows to refine the solution and also be tolerant to the inclusion of poor quality DSMs in the fusion. Experiments show that the method yields a systematic improvement on the performance of the pipeline.

6.1. Introduction

In pair-wise MVS, given a set of N images taken from a scene, $N(N-1)$ ordered pairs can be considered for stereo reconstruction. For each pair, a Digital Surface Model (DSM) of the scene is determined. The final MVS reconstruction of the scene can then be obtained by the integration of all the computed DSMs. The quality of the final reconstruction is determined by the quality of the pair-wise DSMs, which depend on several factors such as the orientations of the views of a pair and changes in the acquisition conditions between the images, among others. Besides the stereo matching step, two other steps are crucial in a pair-wise MVS pipeline to achieve a good reconstruction: (a) the selection of the best pairs to run the pair-wise pipeline and (b) the final integration of the resulting DSMs.

Regarding the pair selection step, multiple factors may influence the quality of a pair and it is hard to identify all of them and tell their relative importance. This difficult task has been traditionally tackled by designing heuristics that take into account the metadata of the images [17, 29]. In [71] a supervised machine learning approach was proposed to derive a quality indicator from the metadata of a pair.

On the other end of the pair-wise MVS pipeline, different methods can be applied for the integration (also called fusion or aggregation) of the information of the computed DSMs. Averaging is the most basic approach; but integration by the median is usually preferred as it takes into account the presence of outliers in the DSMs, as can be seen in a recent review on the matter [63].

This chapter focuses on the pair selection and the DSM integration steps and presents two contributions to enhance the performance of a satellite MVS pipeline. Firstly, for the pair selection, an approach based on the simulation of image and camera model pairs is presented. Synthetic stereo pairs are simulated under all possible geometric configurations on the hemisphere surrounding an artificial scene and the stereo reconstruction quality can be assessed for each pair. This precomputed quality is then used as a proxy for the quality of real pairs of images. Secondly, for the integration step, an approach based on the bilateral filtering [80] is presented. Contrary to the most commonly used per-pixel median, the approach allows to better integrate DSMs considering the spatial coherence of different properties of the data such as height and gray level. The method produces a spatial regularization effect, without affecting the borders of the structures as can be seen in Figure 6.1.

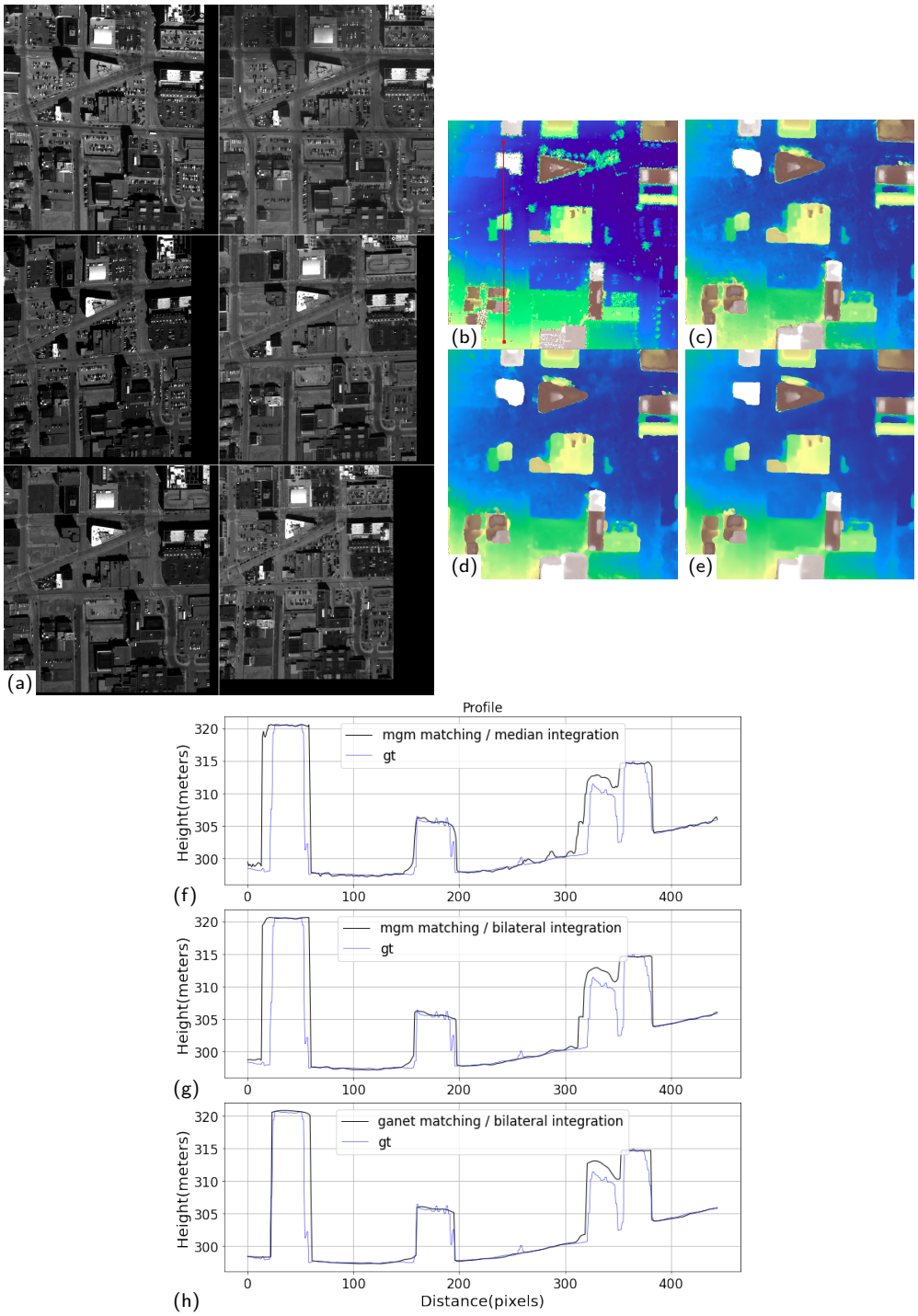

Figure 6.1: (a) Six images of a region of the Omaha dataset. (b) Ground truth (GT) height map of the region. MVS reconstructions using all 30 stereo pairs: (c) DSMs computed using the MGM [28] correlator and integrated by the median of the DSMs, (d) DSMs computed using the GANet [39, 88] correlator and integrated by the median of the DSMs, (e) DSMs computed using the GANet correlator and integrated by bilateral filtering. Profiles corresponding to the red line: In (f), profiles from images b and c, in (g) from images b and d, and in (h) from images b and e.

6.2. Pair selection for multi-view stereo

In pair-wise MVS, it is well known that the DSM aggregation improves in general the completeness [29, 64]. A new stereo pair may give information of an occluded part of the scene. However, if a DSM computed from a bad pair is included, the result may degrade (see Figure 6.2). This issue, along with the fact that the number of possible pairs grows as $O(N^2)$, with N the number of images, makes necessary to pre-select the best pairs to be used for the reconstruction.

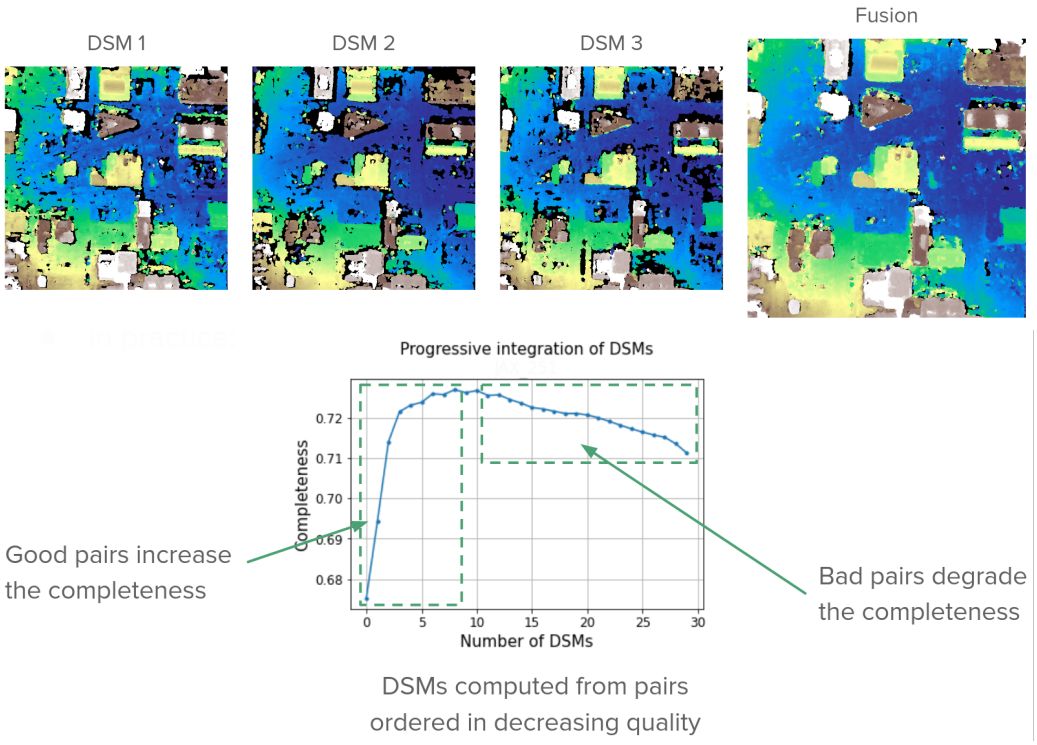

Figure 6.2: Importance of pair selection. The fusion of DSMs increases in general the completeness of a reconstruction but, in practice, the inclusion of DSMs from bad pairs may degrade it. Top: Three DSMs complement each other and increase the completeness of the reconstruction. Bottom: Typical completeness curve for the progressive integration of DSMs from pairs with decreasing quality.

In [29] a simple heuristic based on the images metadata was proposed: images in the pair must have an incidence angle smaller than 40^{o}, the angle between views should be in the range $[5^{\text{o}}, 45^{\text{o}}]$, preferably around 20^{o} and pairs with near acquisition dates are preferred. In [17] pairs with angle between views in the range $[15^{\text{o}}, 25^{\text{o}}]$ are preferred for urban and industrial areas. The preference for 20^{o} for the angle between view is called in this chapter "20^{o} heuristic".

Here we present a method to empirically map the relation between the orientation of the views and the reconstruction quality through simulation. The simulation tool produce views of a 3D scene from multiple orientations generating images

along with RPC models suitable for a pair-wise stereo pipeline. The stereo reconstructions can be assessed by comparing to the known altitude of the scene. This enables to pre-compute a map that encodes the reconstruction quality in relation to the incidence angles of the views with the vertical (or zenith angles) and the intersection angle (angle between views) of any pair of views sampled from the hemisphere surrounding the scene. This map acts as a proxy for the quality of real pairs and can be used to sort the pairs in a more funded way than the previous heuristics.

6.2.1. Stereo reconstruction from simulated image-RPC pairs

The simulator tool presented in Chapter 5 allows to draw any pair of views in the hemisphere surrounding a 3D scene. With the generated image pair and their corresponding RPC it is possible to compute a stereo reconstruction with a satellite pipeline and evaluate the reconstruction against the ground truth (GT) altitude of the scene. This enables to empirically study and map the relation between the orientation of the views and the quality of the 3D reconstruction of a pair.

We sampled the hemisphere over a 3D scene and generated pairs of image-RPC from those positions. Figure 6.3 shows the distribution of the considered reference and secondary views in the hemisphere over the scene. In the plots, the sampled reference views are depicted with a square dot and secondary views with circle dots. Relative orientations are considered with the reference view in zero azimuth. A reference-secondary pair keeps the same relative orientation when a vertical rotation is applied, so should give similar results. Texture or noise can favor some orientations over others. To smooth out these effects, six orientations are considered for each reference-secondary pair. The results are computed as the median over the six cases.

The DSMs computed from the sampled pairs were compared against the GT DSM to assess the reconstruction performance. The completeness (COMP) metric, defined as the proportion of the evaluated pixels where the altitude of the computed map differs from the GT less or equal than $z\_tol = 1m$, was considered for the tests.

The completeness is a comprehensive metric traditionally used for the evaluation of satellite stereo reconstructions [9, 10]. Among the evaluated pixels (i.e. with GT information) there are two types of errors in a reconstructed DSM: (a) *Invalid* pixels where the altitude could not be computed, (b) *Bad* pixels where the computed altitude differs from the GT more than a given threshold. *Invalid* pixels are places with incoherent disparities between left and right disparity maps on the stereo matching step. These are mostly caused by occlusions. *Bad* pixels may arise due to matching or triangulation errors. In the first case, repetitive textures may cause coherent matching at a wrong position. Regarding triangulation, the angle between views is the main factor that determines the uncertainty of this step. Small angles between views result in a worse conditioning of the triangulation, which amplifies the small matching errors. On the other hand, an off-nadir view implies a foreshortening in one direction causing an anisotropic loss of resolution

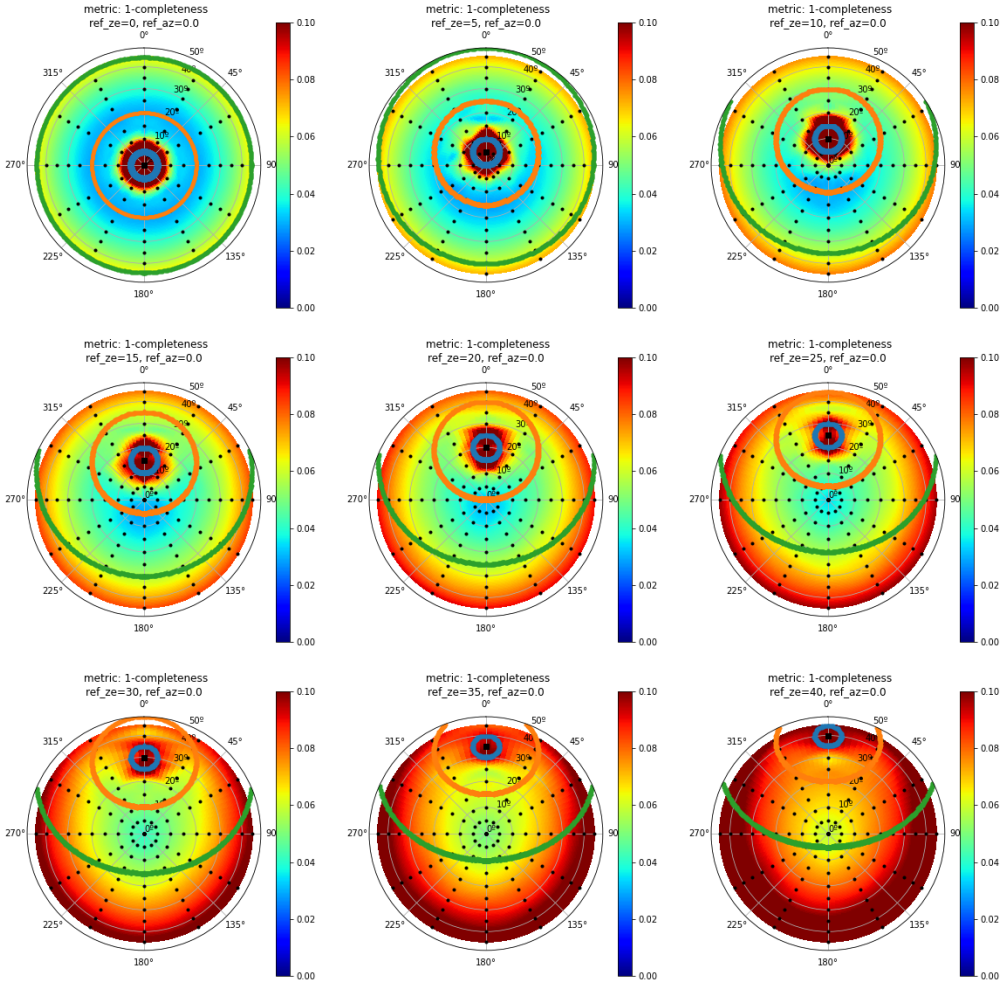

Figure 6.3: Reconstruction errors on the simulated stereo pairs for different reference-secondary orientations. The square point represents the reference view and the circular points represent the tested secondary views. Metric 1-COMP is shown for increasing zenith angle of the reference view. Curves in blue, orange and green indicate the positions in the hemisphere where the views are 5º, 20º and 45º apart from the reference, respectively.

in the image. Indeed, as shown in [61], the relative position of the views in the hemisphere surrounding the scene entails an affine deformation between the images that may bedevil the matching. *Totalbad* is defined as the sum of *Bad* and *Invalid* pixels and is the complement of the completeness *Totalbad = 1 - COMP*.

Stereo reconstruction is affected in a complex manner by all these factors. The analytical study of all these contributions and interactions is hard, thus simulation becomes a good alternative to tackle this problem.

Figure 6.3 illustrates the reconstruction error as a function of the reference-secondary relative image orientations. These results are computed using simulations from the cylinder scene, but similar results are obtained with more complex scenes. In each case the reference image—the black square—is positioned in a cer-

tain zenith angle and the secondary image—the black circular dots—are positioned in a sampling of the hemisphere. Intermediate values are obtained by interpolation of the calculated values at the sampled positions. Blue corresponds to small errors while red indicates large errors.

6.2.2. MVS pair selection based on simulation results

Given a set of N real satellite images taken from the same region, there are $N \times (N-1)$ possible ordered pairs. For each candidate pair, we can estimate the reconstruction error (as $1 - \text{COMP}$) by querying the orientations of the real images in the pre-computed error maps of Figure 6.3. This provides an ordering for the integration of the DSMs reconstructed from the pairs. This ordering based on the completeness obtained from the simulation acts as a proxy for the true completeness, which cannot be computed in a real scene where the GT is not available.

6.3. DSM integration

Multiple strategies to integrate DSMs have been proposed. A recent review [63] presents a list of methods.

The most common way of integrating a set of DSMs is to apply a per-pixel median of the heights in the set of DSMs. This usually yields a robust estimation and removes most outliers in the DSMs. However, this pixel-wise approach does not introduce spatial coherence.

In this chapter an approach based on the bilateral filter [80] is presented. The method is related to the one presented in [70] in the sense that both try to include a spatial regularization inspired on the bilateral filter. In [70] for each pixel an irregular region around it is determined considering spatial and color proximity and then a median of the values of the DSMs is applied on that region. Instead, we directly apply a bilateral filter to the samples in the DSMs. The bilateral filter framework allows to robustly integrate the spatial information along with other available sources of information that can regularize the final integrated DSM. Typically, the framework can integrate not only the height of the DSMs and the gray level or color of a reference image, but also other features as a semantic segmentation or confidence maps if available.

The bilateral filter framework is applied in an iterative scheme. This allows to gradually refine the solution. Using progressively more restrictive ranges for the height allows to focus on the height samples that are close to the previous estimation and are then probably more accurate. The bilateral filter integration of a set L of K DSMs for pixel i at iteration n is computed as

$$B[i] = \frac{1}{\nu(i)} \sum_{k} \sum_{j \in [w_{s_n}, w_{s_n}]} W[k][i,j] L[k][i-j], \tag{6.1}$$

Algorithm 1: Iterative bilateral DSM integration

input : List of DSMs: L = [DSM[k], k:0...K-1]
Ortho-rectified Reference image: I
Number of iterations: N
List of range sigmas: R=[r_n, $n : 0...N-1$]
List of spatial sigmas: S=[s_n, $n : 0...N-1$]
List of color sigmas: C=[c_n, $n : 0...N-1$]

output: Integrated DSM: D

$D \leftarrow pixel\_wise\_median(L)$
for n *in* $0...N-1$ **do**
  $L \leftarrow [register\_in\_height(L[k], D), for\, k\, in\, 0...K-1]$
  for *each pixel* i **do**
    $B[i] \leftarrow$ as in equation (1)
  $D \leftarrow B$

where

$$W[k][i,j] = e^{-\frac{|j|^2}{2.s_n^2}} e^{-\frac{|L[k][i-j]-D[i]|^2}{2.r_n^2}} e^{-\frac{|I[i-j]-I[i]|^2}{2.c_n^2}}. \tag{6.2}$$

Here k is an index on the DSMs list, j is an index on the spatial neighbors of pixel i, r_n is the standard deviation of the Gaussian that determines the height range neighborhood, s_n is the standard deviation of the Gaussian that determines the spatial neighborhood and c_n is the standard deviation of the Gaussian for the gray/color value neighborhood on iteration n in all cases, and the normalization factor is

$$\nu(i) = \sum_k \sum_{j \in [w_{sn}, w_{sn}]} W[k][i,j]. \tag{6.3}$$

Algorithm 1 shows the main steps of the method. The inputs are a list of registered DSMs L, a reference gray/color level image I and lists of sigmas to be applied in each iteration to weight the contribution of neighbors to the integrated altitude of each pixel (S: proximity, R: altitude similarity, and C: gray/color value similarity). The image I can be one of the images from the stereo pairs used to compute the DSMs (ortho-rectified to match the DSMs). The integration D is initialized by the per-pixel median of the set of DSMs. For each iteration, (a) the DSMs in L are registered in height to D (shift of each DSM to match the median height of D), (b) the integrated altitude of each pixel is computed.

Table 6.1: Analysis of the pair rankings given by the heuristic and the simulation compared to the rankings given by the true reconstructions. For a given metric each cell shows the Kendall-tau correlation and its p-value. For example, in the cell corresponding to Bad/Cylider/OMA_203, the ranking by metric Bad of the pairs from that region (metric computed comparing the DMSs against the GT) and the ranking by metric Bad for the cylinder based simulation, have a Kendall-tau correlation of 0.68 with p-value $< 0{,}01$. The water and vegetation pixels were masked out for this analysis. Cells highlighted in bold correspond to correlations with p-value $< 0{,}05$.

	Bad			**Invalid**			**Totalbad = 1 - COMP**		
Region	**20º heuristic**	**Cylinder**	**City**	**20º heuristic**	**Cylinder**	**City**	**20º heuristic**	**Cylinder**	**City**
OMA_203	-0.19 (0.91)	**0.68 (<0.01)**	-0.08 (0.71)	**0.25 (0.03)**	**0.69 (<0.01)**	**0.63 (<0.01)**	0.15 (0.13)	**0.30 (0.01)**	0.09 (0.26)
OMA_247	-0.36 (1.00)	**0.78 (<0.01)**	-0.13 (0.82)	-0.05 (0.64)	**0.30 (0.01)**	**0.26 (0.03)**	-0.20 (0.93)	-0.01 (0.53)	-0.28 (0.98)
OMA_251	-0.41 (1.00)	**0.72 (<0.01)**	-0.13 (0.81)	0.12 (0.21)	**0.50 (<0.01)**	**0.43 (<0.01)**	-0.05 (0.63)	0.19 (0.10)	-0.10 (0.75)
OMA_287	-0.40 (1.00)	**0.68 (<0.01)**	-0.08 (0.68)	0.00 (0.50)	**0.47 (<0.01)**	**0.46 (<0.01)**	-0.21 (0.91)	0.00 (0.50)	-0.03 (0.56)
OMA_353	-0.44 (1.00)	**0.74 (<0.01)**	-0.14 (0.86)	**0.29 (0.01)**	**0.66 (<0.01)**	**0.65 (<0.01)**	-0.09 (0.74)	0.12 (0.19)	-0.10 (0.78)
JAX_156	-0.07 (0.71)	**0.68 (<0.01)**	0.17 (0.10)	0.08 (0.28)	**0.24 (0.03)**	**0.38 (<0.01)**	0.07 (0.31)	**0.38 (<0.01)**	-0.01 (0.54)
JAX_165	-0.08 (0.73)	**0.72 (<0.01)**	0.20 (0.08)	0.08 (0.29)	**0.31 (0.01)**	**0.37 (<0.01)**	**0.28 (0.02)**	**0.35 (<0.01)**	**0.39 (<0.01)**
JAX_214	-0.10 (0.76)	**0.58 (<0.01)**	0.14 (0.17)	0.14 (0.17)	**0.26 (0.03)**	**0.41 (<0.01)**	0.19 (0.09)	**0.28 (0.02)**	**0.30 (0.01)**
JAX_251	-0.06 (0.68)	**0.70 (<0.01)**	0.16 (0.11)	0.05 (0.35)	**0.32 (0.01)**	**0.44 (<0.01)**	**0.28 (0.02)**	**0.43 (<0.01)**	**0.31 (0.01)**
JAX_264	-0.10 (0.77)	**0.58 (<0.01)**	0.07 (0.29)	0.07 (0.31)	**0.28 (0.02)**	**0.38 (<0.01)**	0.17 (0.10)	**0.35 (<0.01)**	0.02 (0.43)
MVS_001	**0.28 (0.02)**	**0.80 (<0.01)**	**0.60 (<0.01)**	-0.22 (0.95)	**0.44 (<0.01)**	**0.65 (<0.01)**	**0.45 (<0.01)**	**0.27 (0.02)**	**0.37 (<0.01)**
MVS_002	**0.36 (<0.01)**	**0.86 (<0.01)**	**0.63 (<0.01)**	-0.11 (0.80)	**0.43 (<0.01)**	**0.64 (<0.01)**	**0.65 (<0.01)**	**0.79 (<0.01)**	-0.11 (0.80)
MVS_003	**0.30 (0.01)**	**0.88 (<0.01)**	**0.58 (<0.01)**	-0.25 (0.97)	**0.41 (<0.01)**	**0.64 (<0.01)**	**0.43 (<0.01)**	0.15 (0.13)	**0.53 (<0.01)**
MVS_004	**0.28 (0.02)**	**0.86 (<0.01)**	**0.57 (<0.01)**	-0.21 (0.94)	**0.40 (<0.01)**	**0.62 (<0.01)**	**0.54 (<0.01)**	**0.30 (0.01)**	**0.48 (<0.01)**
MVS_005	0.17 (0.11)	**0.75 (<0.01)**	**0.48 (<0.01)**	-0.26 (0.97)	**0.37 (<0.01)**	**0.59 (<0.01)**	**0.45 (<0.01)**	**0.38 (<0.01)**	0.00 (0.49)

6.4. Experiments

Experiments were conducted to test the behavior of the S2P pipeline when changing the pair selection step as proposed in Section 6.2 and the integration step as proposed in Section 6.3. We used three datasets, consisting on satellite images from the Multiple View Stereo Benchmark for Satellite Imagery (MVS3D) [10] and the US3D dataset [9] already presented on chapter 2.

For our evaluation, 5 subregions from each of the datasets are considered. In each subregion, a set of 6 images is considered in order to allow a tractable pairwise analysis. This gives a set of 30 ordered pairs for each subregion. Images in each set span a small time interval (same day or some days apart) to avoid seasonal changes that could hinder the study.

We tested with two different matching algorithms in the S2P pipeline (MGM [28] and GANet [88]) to show that the presented improvements are rather independent of the used method. In order to evaluate the performance of the different approaches two metrics were considered [9, 10]: (a) Completeness (COMP): Proportion of evaluated pixels where the altitude of the computed map differs from the GT less or equal than $z\_tol = 1m$. (b) Accuracy as the Median Absolute Error (MAE) between computed and GT maps considering only the pixels that have valid information in both maps.

6.4.1. Analysis of the pair selection strategy

To analyze the usefulness of the simulation for pair selection, we study if the simulation proxy ranks the pairs in a better way than the commonly used heuristics [29]. This is done by comparing the reconstructed DSMs against the GT.

Table 6.1 presents the correlation results for the pair rankings given by the heuristic (described in Section 6.2) and the presented simulation proxy, compared to the rankings obtained by evaluating the true reconstructions. The analysis is repeated for each of the error metrics (Bad, Invalid and Totalbad). For a given metric each cell in the table shows the Kendall-tau (KT) rank correlation coefficient and its corresponding p-value [1, 45]. Simulations are made on both scenes shown in Figure 5.2 (i.e. Cylinder and City).

As mentioned in Section 6.2, the errors of a stereo reconstruction are comprised of *Bad* and *Invalid* pixels. While the *Bad* pixels are more related to the orientation of the views, *Invalid* pixels have a strong relation to the geometry of the scene (e.g. occlusions are related to the contents and the spatial relation of the objects in the scene). Results on Table 6.1 show that for the simulation with the Cylinder scene, there is a significant correlation between the rankings for both the Bad and Invalid metrics on simulations using the Cylinder and City scenes. This simple scene correctly captures the main error components given the view orientations and can rank the pairs in a similar way as with the real images. In particular, the simulation on the Cylinder scene presents a strong correlation for the Bad metric, which is mostly related to the views and not to the scene. Bad pixels are mostly related to the view orientations and not to errors in the matching step and the simple Cylinder scene allows to observe these errors independently of problems

that a more complex scene could introduce. In the case of the Invalid metric, the Cylinder has still a positive and significant correlation but the City scene, with a more complex structure, represents better the inter-occlusions of a urban scene.

From Table 6.1 we see that the proposed simulation based pair selection rightly predicts the best ordering in relation with the *Bad* and the *Invalid* number of pixels, but is less conclusive for the number of *Totalbad* pixels. We shall note that Bad and Invalid are antagonistic metrics. A large angle between views reduces the uncertainty for the triangulation while causes large occluded regions. This antagonistic relation and the dependence on the scene layout explains that the correlation for *Totalbad* with the ideal ordering is not as strong as the correlation of its components. We posit that simulating using an adequate layout adapted to each particular scene (a priori unknown), would improve this correlation. Despite this limitation, the results show that the pair selection method using a very simple simulation model (Cylinder) gives good results in mean, as illustrated in Figure 6.5, and works better than the currently used heuristic method.

The experiments show that it is possible to develop a better founded pair selection strategy than the currently used heuristic [29]. The simulation tool allows to consider all the possible configurations of incidence and angles between views, related to the two error types mentioned in Section 6.2. Overall, the correlation of the heuristic strategy with the different metrics is rather disappointing, except for the case of the MVS sets, on which it was fine-tuned [29]. This seems to indicate that the existing heuristic miss some relevant cases. For instance, note that in Figure 6.3 the heuristic of 20^{o} preference for the angle between views [29] is confirmed by the first plots. But the simulation reveals that as the incidence angle of one of the views grows, it is preferable to have the other view near the nadir even if the angle between views moves away from 20^{o}. Less error is found when the secondary view has a similar azimuth to the reference (that is, moving from the reference to the nadir). A secondary view near the nadir in the same azimuth as the reference does not increase the occlusions and has maximum resolution as it minimizes the foreshortening. That view will be better than one 20^{o} apart from the reference with different azimuth (e.g a view to a side of the reference with same zenith angle).

The simulation can be used as a pre-computed mapping to estimate the expected completeness and select the pairs in such a way as to minimize the 3D reconstruction error. The analysis of the simulation results shows that a simple scene as the Cylinder can be used to order the pairs.

6.4.2. Analysis of the end-to-end performance.

In order to assess the end-to-end effects of the presented contributions (pair selection and DSM integration) the following configurations of the pipeline were tested: (C1) Selection by the 20^{o} heuristic, integration by median, as in the current S2P pipeline, (C2) Selection by simulation as proposed in section Section 6.2, integration by median, (C3) selection by simulation, integration by iterative bilateral filtering as presented in Section 6.3. Regarding the iterative bilateral filtering, the

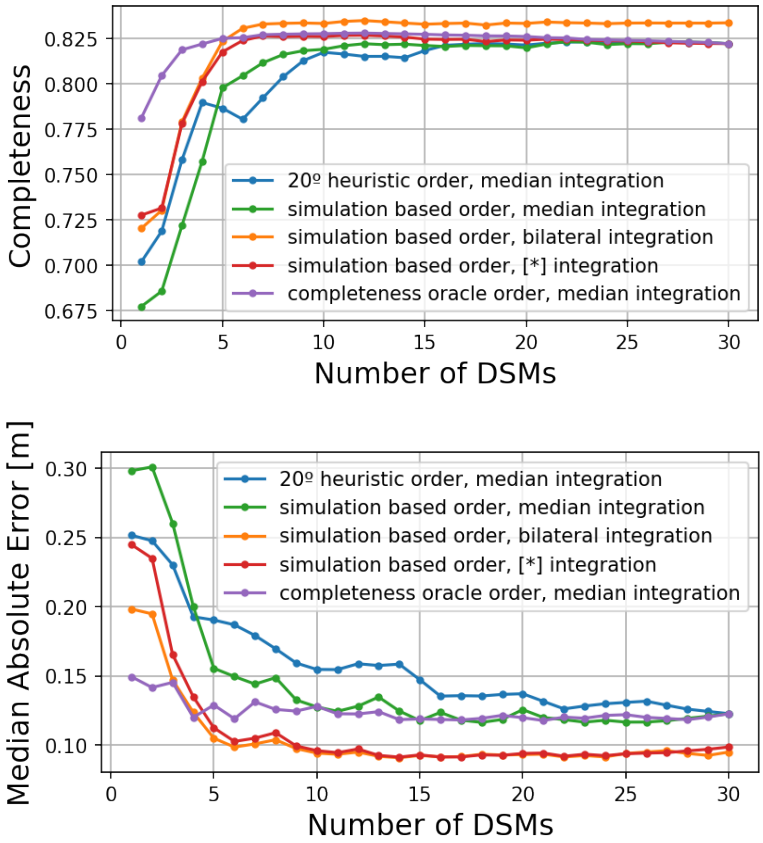

Figure 6.4: Progressive integration of DSMs for region JAX_156. Completeness (top) and MAE (bottom) evolution when integrating a growing number of DSMs. The purple curve corresponds to an integration by the median and an oracle ordering. Curves in blue, green and orange correspond to the C1, C2 and C3 configurations of the pipeline respectively. Red curve (marked with [*]) corresponds to an implementation of the method in [70]. Please refer to the text for a complete description.

shown results use a decreasing sigma for the height range of $[2{,}5, 2{,}0, 1{,}5, 1, 0{,}5]$, spatial sigma of 6 and color sigma of 20 % of the gray level range.

Figure 6.4 illustrates the performance change on one region of the dataset when the contributions of this work are introduced in the pipeline (results for other regions can be seen in Figure 6.7 and Figure 6.7). The graphs show the behavior of the two metrics—completeness and median of the absolute error—for the region as the number of integrated DSMs is increased according to different ordering criteria. The purple curve depicts an integration with the median and an "oracle" ordering based on the actual completeness computed with an available altitude GT. In the oracle ordering, the next DSM is selected to maximize the completeness of the integration up to the moment. This almost optimal ordering illustrates the common situation where the completeness peak is achieved with the DSMs from a few good pairs and the inclusion of more DSMs degrades the aggregated result. These "toxic" DSMs are the result of bad image pairs. The oracle puts these

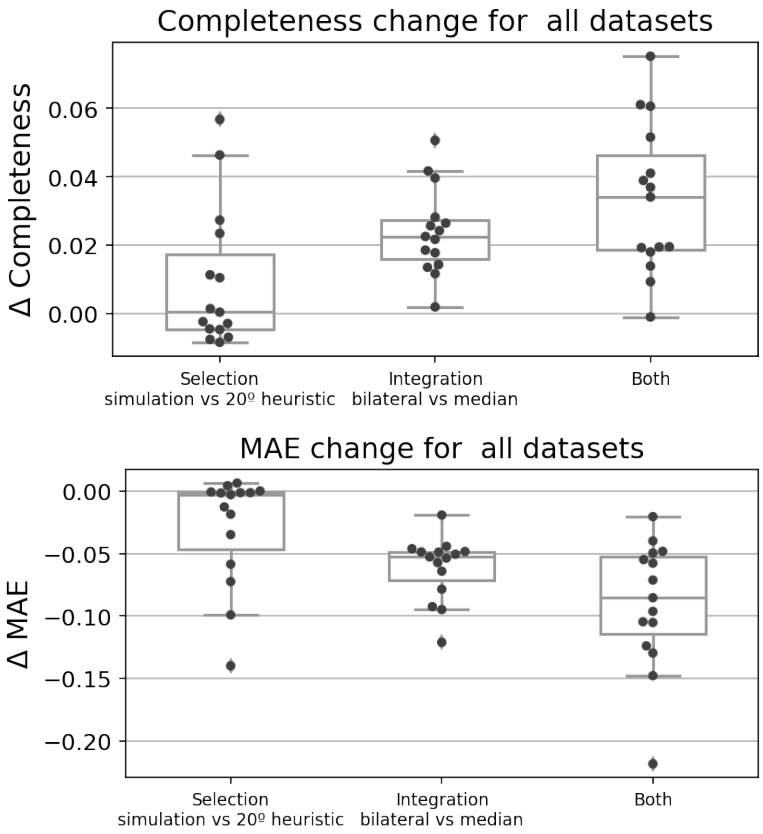

Figure 6.5: Incremental results between configurations C1, C2 and C3 (see text). For each region in the datasets, the top 5 DSMs are integrated and the resulting DSM compared to the GT. Left: C2 - C1 (pair selection by simulation vs. heuristics), Center: C3 - C2 (integration by bilateral filter vs. integration by the median). Right: C3 - C1 (both improvements vs. the original baseline pipeline). Note how each contribution increases the global performance.

bad DSMs at the end of the ordering. Blue, green and orange curves correspond to the C1, C2 and C3 configurations respectively. In the example of Figure 6.4 the introduced methods enhance the completeness and the accuracy allowing to achieve better results with fewer DSMs. Considering, for example, the first five selected DSMs, the selection by the simulation is closer to the ideal selection by the oracle; integration by iterated bilateral filtering adds another performance boost that surpasses the peak performance of the integration by the median.

This trend is general for the ensemble of the datasets as depicted in Figure 6.5, which shows, for all the tested regions (dots in the figure), the variation in the reconstruction metrics when the presented improvements are introduced into the pipeline. The three configurations defined before (C1, C2 and C3) are considered: C1 is the baseline, C2 changes the selection of pairs with respect to C1, and C3 changes the integration step with respect to C2. Figure 6.5 shows the error metrics change considering (C2-C1), (C3-C2) and (C3-C1) to evaluate the contribution of each proposed improvements in the global performance. The boxplots show that

Table 6.2: Results for the whole satellite pipeline on the tested data sets for configurations C1, C2 and C3. The results are the average of the metrics on the datasets. In all cases the comparison is against the GT and using the best five pairs chosen either by the heuristics or by the simulation tool. Note how the integration by bilateral filter improves the completeness (COMP) over the median approach both when using the MGM or the GANet matching methods. The same behavior is observed for the Median of the absolute differences (MAE).

			COMP				MAE			
Selection 5/30	**Matching method**	**DSM Integration**	**JAX**	**OMA**	**MVS3D**	**All**	**JAX**	**OMA**	**MVS3D**	**All**
20^{o} Heuristic	MGM	Median	0.700	0.811	0.720	0.744	0.306	0.231	0.285	0.274
By simulation	MGM	Median	0.733	0.811	0.715	0.753	0.225	0.227	0.284	0.245
By simulation	MGM	Bilateral	**0.747**	**0.829**	**0.732**	**0.770**	**0.199**	**0.180**	**0.255**	**0.212**
20^{o} Heuristic	GANet	Median	0.695	0.832	0.723	0.750	0.373	0.232	0.306	0.304
By simulation	GANet	Median	0.725	0.830	0.718	0.758	0.296	0.246	0.307	0.283
By simulation	GANet	Bilateral	**0.736**	**0.838**	**0.731**	**0.769**	**0.275**	**0.228**	**0.287**	**0.263**

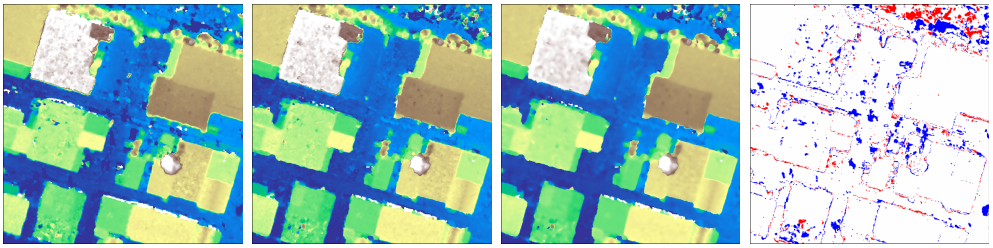

Figure 6.6: An example of results for a region from the Jacskonville dataset. From right to left the first 3 columns show the reconstruction using the best 5 DSMs and the configurations: (C1) Selection by the 20^{o} heuristic, integration by median, (C2) Selection by simulation, integration by median, (C3) selection by simulation, integration by iterative bilateral filtering. All results use the MGM stereo matcher. Last column graphically compares the completeness difference between (C3) and (C1): blue color are correctly reconstructed pixels (height error $< 1m$) by (C3) and badly reconstructed by (C1); red color are badly reconstructed pixels by (C3) and correctly reconstructed by (C1). Note that the improvements –in blue– prevail and are concentrated on the edges of the structures, with the exception of an upper right region with vegetation.

each contribution produce a consistent improvement in the reconstructions both in completeness and accuracy.

As mentioned in Section 6.3, the presented bilateral filtering integration method is related to [70]. The method, hereafter called bilateral median (BM), was implemented in order to compare it with our proposal. Figure 6.4 compares, for a given image, the integration evolution with the BM (red) and with the bilateral filtering (green), using the same parameters. BM exhibits very good results with the first few DSMs but falls behind bilateral filtering integration as the number of DSMs increases. This evolution is similar for all the tested regions. While color range and spatial regularization are common to both methods, the ability to take into account the height range with decreasing sigmas is key to integrate the best of all available DSMs. Note that the implementation and parameter values for the BM method might differ from the actual method in [70]. Results on other sub-regions are shown in Figure 6.7 and Figure 6.8.

Table 6.2 summarizes the results obtained for configurations C1, C2, and C3,

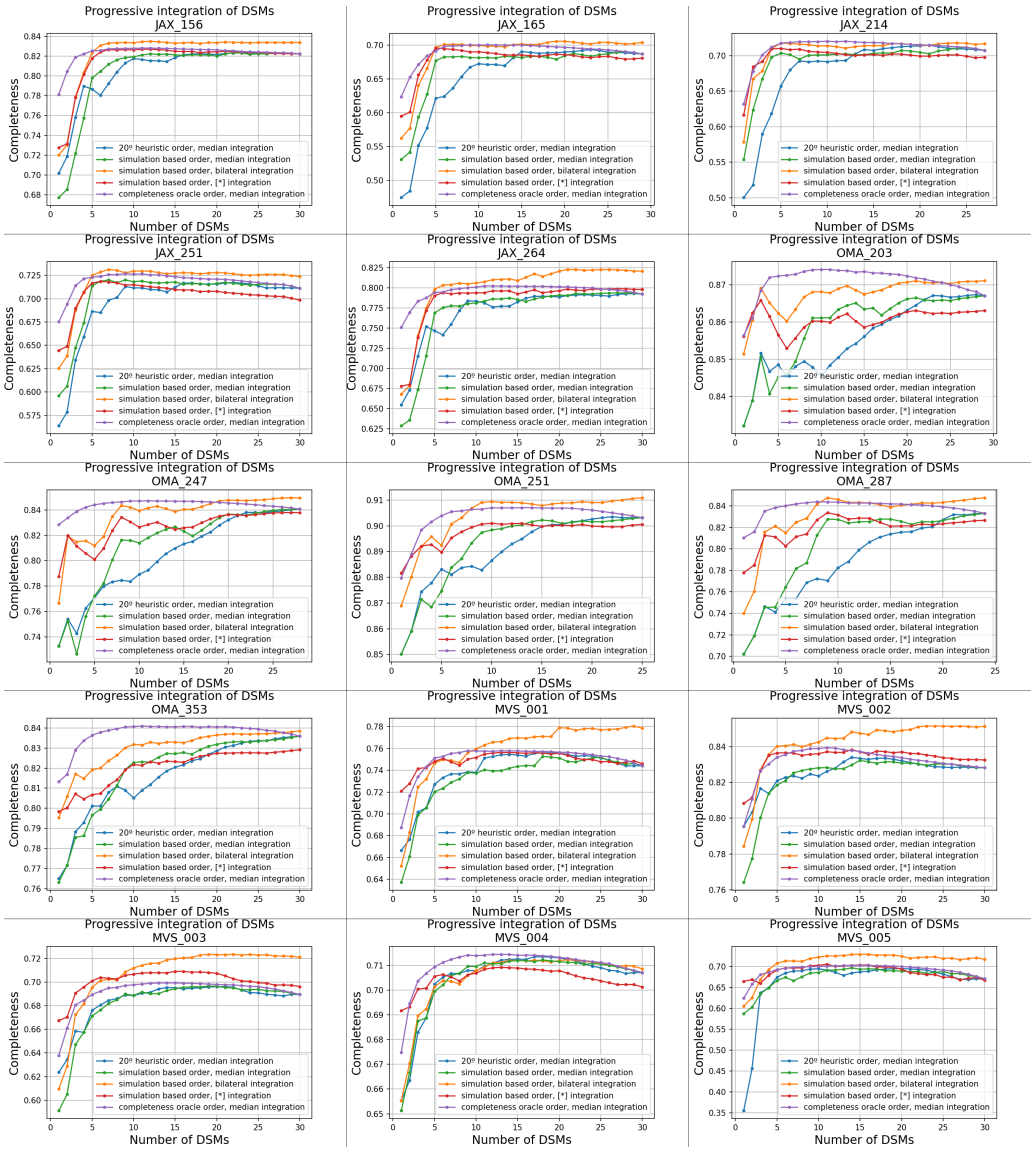

Figure 6.7: Completeness metric for the progressive integration of DSMs for several regions. [*] integration is based on an implementation of [70].

averaged by site. Performance gain is mainly due to the integration by the bilateral filter. Figure 6.5 shows that the contribution of the selection is positive in mean, in spite of the fact in Table 6.2 that for some sites the simulated based selection is not optimal. Both contributions combined improve the overall performance of the pipeline in terms of completeness and accuracy and this observation persists regardless of the matching method used (MGM or GANet). Figure 6.6 shows that the completeness improvements of the integration method are concentrated on the borders of the structures like buildings. This contributes to a better definition and fidelity to the GT of the reconstructed 3D structures as seen also in Figure 6.1.

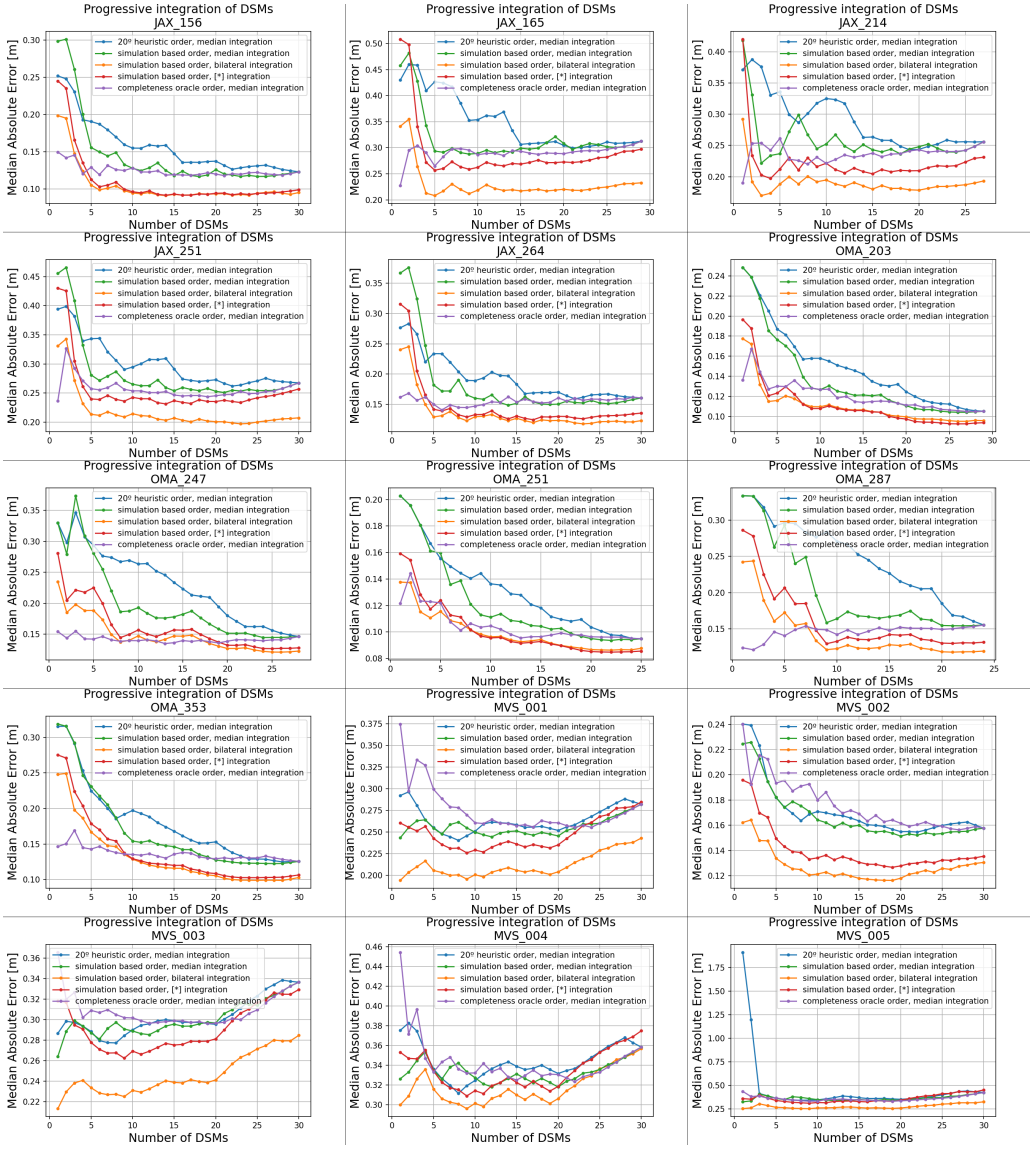

Figure 6.8: Median absolute error metric for the progressive integration of DSMs for several regions. [*] integration is based on an implementation of [70].

6.5. An alternative metric for pair selection

The completeness (COMP) of a reconstruction is defined as the proportion of the evaluated pixels where the computed map has an altitude that differs less or equal than a certain threshold from the ground-truth. The completeness accounts for the pixels that are correct given the threshold, but it does not give information related to the distance of those pixels to the ground truth altitude. The metrics such as the RMSE and the MAE are used to complement the information of the accuracy of the reconstruction.

Usually, a single threshold is used to evaluate the completeness that achieves

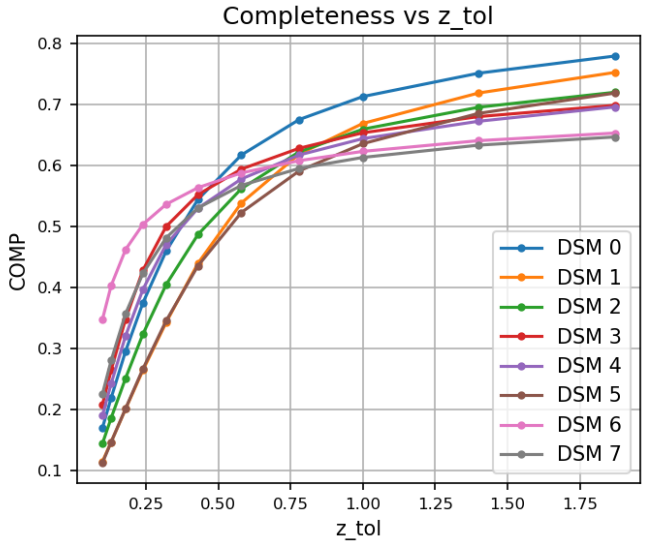

Figure 6.9: Completeness metric as a function of the altitude tolerance z_tol. Pixels are considered correct if their altitude difference to the ground-truth is less than z_tol. Although several of the DSMs have a similar completeness at 1m, the completeness curves show that certain reconstructions are more accurate than others.

a method or a pipeline [10, 19].

Instead of evaluating the COMP with a single threshold or a set of few thresholds, it is possible to tests it in a more dense set of thresholds. Figure 6.9 shows the COMP achieved for the stereo reconstruction of different pairs of images of the JAX dataset. In that figure, the COMP is calculated for a set of thresholds. The curves of COMP as a function of the threshold have different evolutions for the set of reconstructions. Note that the reconstruction for DSMs 5 and 6 achieve the same COMP at 1 meter but their COMP curves are very different. The curves for that pairs show that in the DSM number 6 the pixels are closer to the ground truth than in the other case. Hence, the curve of the COMP as a function of the threshold holds valuable information of the accuracy of the reconstruction that is lost when one considers only discrete thresholds.

While working with the whole curve of COMP vs threshold can be an interesting metric for the reconstruction it may not be practical and simple to manage and use for routine comparison tasks. Instead, we can consider the Area Under the Completeness Curve (AUCC) as a single number that summarizes the information of completeness and accuracy of the reconstruction.

6.5.1. The AUCC in pair selection

As discussed in Section 6.4.1, simulation based pair selection rightly predicts the best ordering in relation with the Bad and the Invalid number of pixels, but is less conclusive for the number of Totalbad (Totalbad = Bad + Invalid = 1-COMP) pixels.

The AUCC as a joint indicator of completeness and accuracy can help improve the pair selection by simulation. Instead of ordering real pairs by the COMP of their respective simulated pairs, the ordering can be done by the AUCC of their respective pairs.

Tests were conducted on regions of the JAX and OMA datasets. For a region, all the available images are considered. For each real image, a simulated image is generated with the same orientation of the real view. All the ordered pairs of real and of simulated images are reconstructed with S2P. The reconstructions from real pairs are compared to the real ground-truth altitude map and the simulated pairs are compared to the ground-truth altitude of the artificial scene. The obtained metrics COMP_real and AUCC_real give an ordering for the real pairs and the simulated pairs.

The metrics computed on simulated pairs COMP_sim and AUCC_sim are a proxy for the metrics computed on real images COMP_real and AUCC_real. COMP_real and AUCC_real are "oracles" that we can compute in this test but could not be done in real life when you don't have the real ground-truth altitude maps.

Figure 6.10 presents the results for a region of the JAX dataset where the DSMs reconstructed from real pairs are progressively fused by the per-pixel median. The integration of the DSMs ordered by the AUCC from simulation is better than the integration of DSMs ordered by the COMP from simulation. The first 50 pairs from 436 pairs picked by the different orderings are shown.

Figure 6.11 shows a similar result for a region of the OMA dataset. In this case the first 50 pairs from 927 pairs are picked by the different orderings.

Another interesting finding is that, in the simulation, it is important not only to test for pairs with the same orientation as the real images, but also with the corresponding sun orientations. This can be seen in Figure 6.12 that shows that the ordering is degraded when the sun is not considered in the simulation of the images.

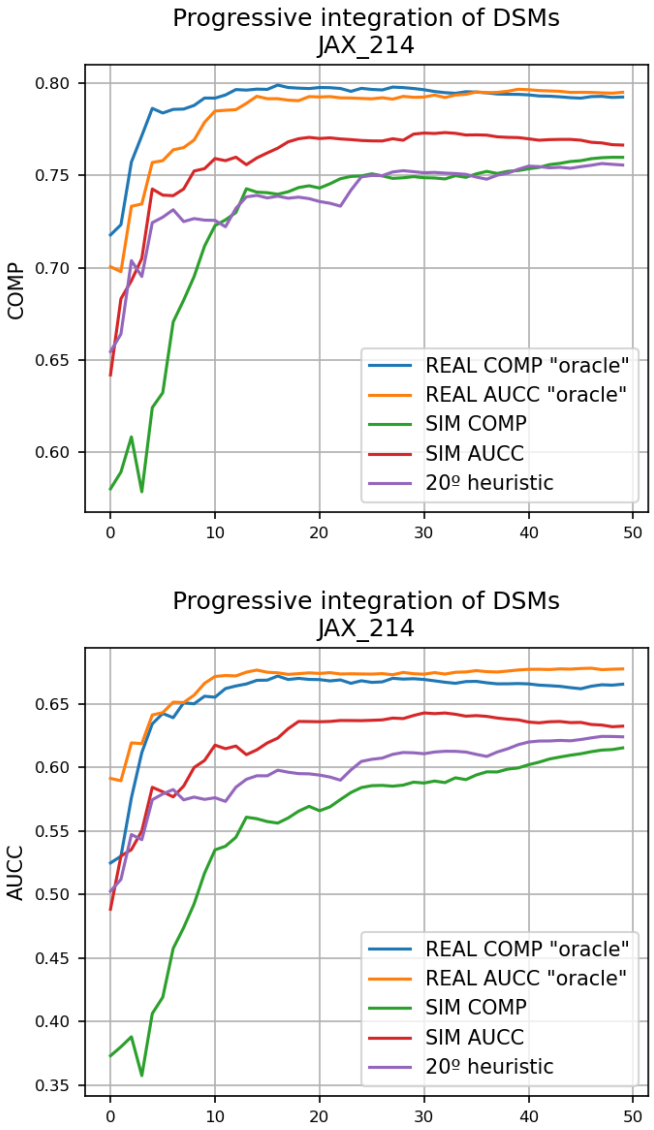

Figure 6.10: Metrics COMP and AUCC achieved by the progressive integration of the first 50 DSMs of 436 selected by the different criteria (oracles, results on simulation and the heuristic of 20º preference for the angle between views).

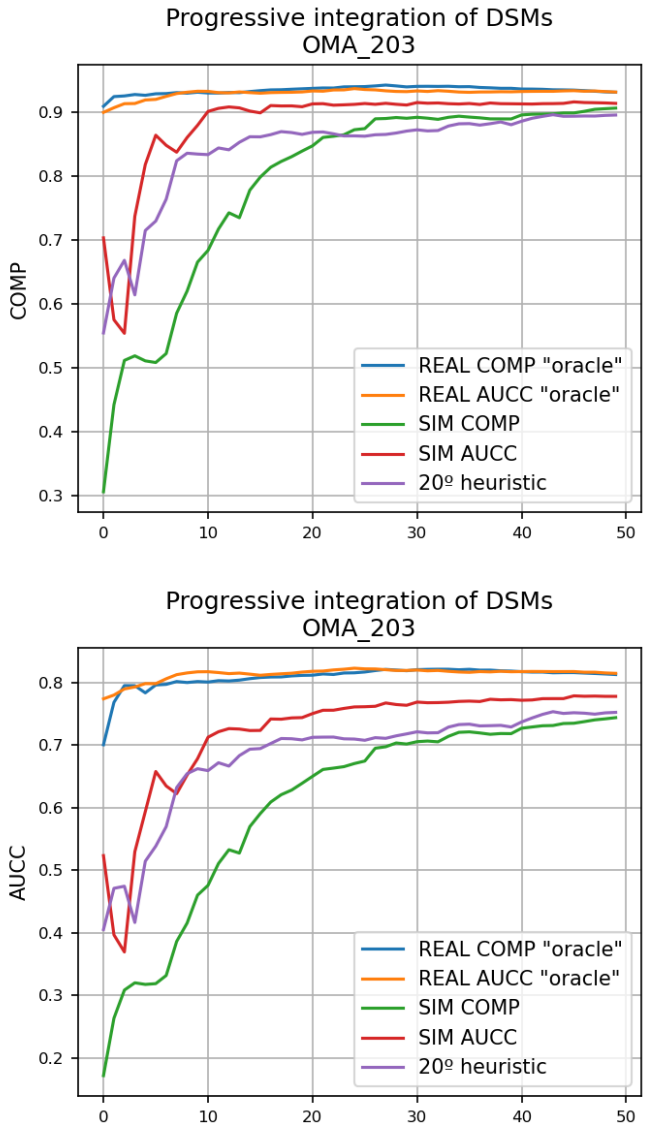

Figure 6.11: Metrics COMP and AUCC achieved by the progressive integration of the first 50 DSMs of 927 selected by the different criteria (oracles, results on simulation and the heuristic of 20º preference for the angle between views).

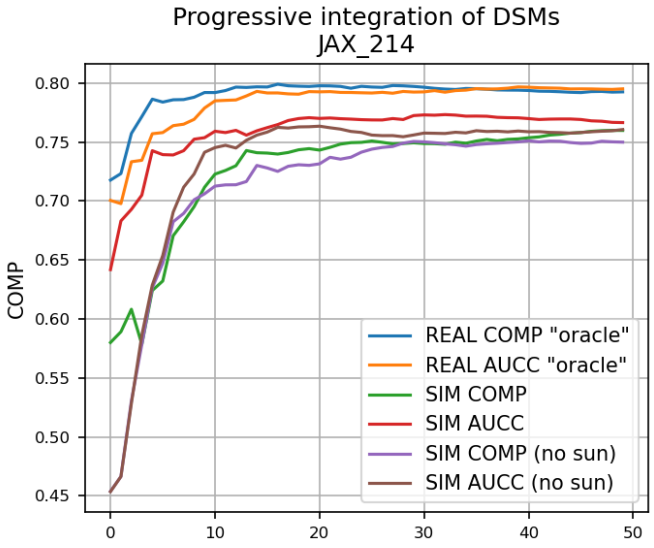

Figure 6.12: When selecting pairs by simulation, the sun position is important. The ordering is degraded when the sun is not considered in the simulation of the images.

This page intentionally left blank.

Chapter 7

Point cloud analysis

Not directly connected to stereo pipelines, other works on point clouds were done during the thesis. Based on the *a contrario* methodology, Lezama et al. proposed in [51] a point alignment detector. This method, originally in 2D were extended to 3D.

7.1. 3D point alignment detector

The method described in this chapter is based on the *a contrario* methodology proposed by Desolneux, Moisan and Morel [22, 23]. It is a mathematical formalization of the *non-accidentalness principle* proposed for perception [3, 85, 86] (also refered as *Helmholtz principle*). In this approach, an observed structure is considered relevant if it rarely occurs by chance. This is implemented assuming a null-hypothesis H_0 for the data where no detections should occur (the a contrario model). The rarity or non-accidentalness of a structure is quantified as the probability of observing that structure under the H_0 hypothesis.

The 3D algorithm described in this chapter is an extension of the 2D algorithm developed by Lezama et al. in [51] and also available in IPOL [52].

In order to detect significant alignments, the algorithm estimates a local density around a candidate alignment, evaluates the regularity of the spacing of the points in the alignment, and defines a criterion and a procedure to select the best interpretation among redundant detections.

Given a data set of 3D points, a candidate alignment consists of a thin cylinder in space defined by two points of the data set. The idea is to evaluate if the point density inside the cylinder is significantly high with respect to the local background. To provide a local estimation of the background density, a coaxial larger cylinder is used to count the points surrounding the candidate alignment (hereinafter we will refer to this density as the "local background density", or abbreviated as "local density").

A meaningful alignment should also have regularly spaced points inside the candidate cylinder. In order to evaluate this regularity the candidate cylinder is divided into equally shaped intervals or pill-boxes and the number of occupied boxes is counted. A candidate alignment will be validated if the number of occupied boxes is significantly larger in a statistical sense relative to the local background density.

The last step of the algorithm consists of a redundancy reduction. Given a meaningful alignment, many smaller or larger cylinders overlapping the main alignment may also be meaningful. This masking phenomenon can involve points that belong to the real alignment as well as background points near but not necessarily part of the alignment. To this aim the non-maximal detections are removed. Candidate alignments are first ordered by decreasing significance. A test is conducted to know if an event masks the neighbouring ones. The candidate alignments are kept if they are not masked by a more significant one.

7.2. Method Description

Consider a set of N points defined in a 3D domain D with volume V_D. We are interested in detecting groups of points that are well aligned. A reasonable a contrario hypothesis H_0 for this problem is to suppose that the N points are the result of a random process where points are independent and uniformly distributed in the domain. This does not mean that the method will only work when the

background points follow exactly this hypothesis. What is important is that this is a good model for isotropic elements where any alignment is accidental.

The validation is done using the a contrario framework. This methodology consists in assuming a null hypothesis H_0 for the data, where no detections should occur, and defining a detection as a violation of this hypothesis (i.e. events that could hardly happen under the hypothesized model). The fundamental quantity of the a contrario approach is the Number of False Alarms (NFA) associated with an event e (up to a certain precision) and a set of points X. Given an a contrario hypothesis H_0 for the set of points in the domain, the NFA of an event is a bound on the expected number of occurrences of the event under H_0. Given an event e, its NFA is

$$NFA(e) = N_{tests} \cdot P_{H_0}(e), \tag{7.1}$$

where N_{tests} is the number of tested events and $P_{H_0}(e)$ is the probability of the event e under H_0.

In order to implement the a contrario approach, the proposed algorithm is organized in two main parts: The first part comprises the search and validation of candidate alignments. This part implies an exhaustive search across the candidates determined by each pair of points in the domain, the estimation of the local density of points around each candidate and the estimation of the regularity of the point locations along the alignment. A candidate alignment will be validated if the regularity of point locations is significantly large in a statistical sense given the local point density. In the second part, the previously validated candidates undergo a non-maximum suppression strategy to reduce redundancy. In this part a candidate alignment is kept if it is not masked by a more significant one.

7.2.1. Candidate Alignment Cylinder

Candidate alignments will be formed by taking every pair of points and constructing a cylinder r, whose axis ends at those points, and has variable width. With N points in the domain this gives a total of $\frac{N(N-1)}{2}$ candidates.

Figure 7.1 shows a schema of the cylindrical regions considered for the tests. Given a pair of points in the domain we can define:

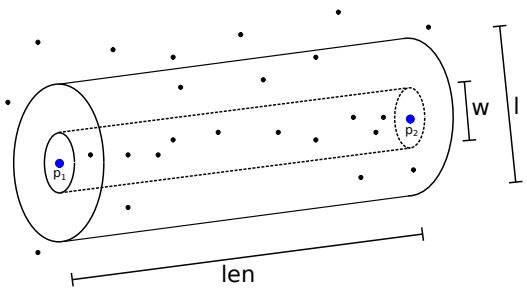

Figure 7.1: A schematic representation of the evaluated alignment determined by a pair of points in the domain. For each candidate, a set of alignment cylinders (with diameter w) and local cylinders (with diameter l) are considered. See the text for further details.

- **$\mathbf{p_1}$**: first point in the pair that determines the candidate alignment,
- **$\mathbf{p_2}$**: second point in the pair that determines the candidate alignment,
- **len**: distance between p_1 and p_2 and also the height of the cylinders considered for the test,
- **w**: diameter of the candidate alignment cylinder. Points within this cylinder *belong* to the alignment,
- **l**: diameter of the local cylinder. The local cylinder is used to estimate the local background density around the candidate alignment.

Since an alignment should be an elongated structure, the ratio $\frac{len}{w}$ must be kept over a reasonably minimum. In our implementation, a minimum $\frac{len}{w}$ ratio of 10 is used.

For each candidate alignment a set of diameters w and l are considered in order to take into account a range of possible density relations. In the implementation a set of $W = 8$ different candidate diameters and $L = 8$ local diameters are considered. The values w and l follow a geometric series with a factor of $2^{-\frac{1}{4}}$ or equivalently $\frac{1}{\sqrt[4]{2}}$:

$$w = \frac{len}{10},\ \frac{len}{10}\frac{1}{\sqrt[4]{2}},\ \frac{len}{10}\frac{1}{\sqrt[4]{2}}\frac{1}{\sqrt[4]{2}},\ \dots\ \dots,\ \frac{len}{10}\{\frac{1}{\sqrt[4]{2}}\}^{W-1},$$

$$l = \frac{len}{\sqrt{10}},\ \frac{len}{\sqrt{10}}\frac{1}{\sqrt[4]{2}},\ \frac{len}{\sqrt{10}}\frac{1}{\sqrt[4]{2}}\frac{1}{\sqrt[4]{2}},\ \dots\ \dots,\ \frac{len}{\sqrt{10}}\{\frac{1}{\sqrt[4]{2}}\}^{W-1}.$$

7.2.2. Density Estimation

In order to evaluate if a configuration of points forms a statistically significant alignment, the method requires an estimation of the density of points around that candidate. The density should be estimated locally around the candidate since the significance is related to the saliency of the alignment with respect to its near background.

The local density can be estimated by counting the points in the local cylinder of diameter l which has a volume $V_l = len \cdot \frac{\pi l^2}{4}$. Although this is a reasonable approach when the points are more or less uniformly distributed in space it can lead to unwanted detections in the case of flat borders between regions of different density as shown in Figures 7.2 and 7.3. When the points are gathered in these flat structures the estimated local density in the border can be low with respect to the density in the alignment cylinder and points in the border of the structure can be detected as an alignment.

To overcome the problem in this kind of structures, the local density can be estimated conservatively by dividing the local cylinder in sectors and considering the maximum count among the sectors. Figure 7.4 depicts the idea in the case of four sectors. The volume of the local cylinder, excluding the alignment cylinder, is divided in S sectors (4 in this example) named S_1 to S_S. Let M_1 to M_S be the respective point counts in those sectors. Let M be the point count in the alignment

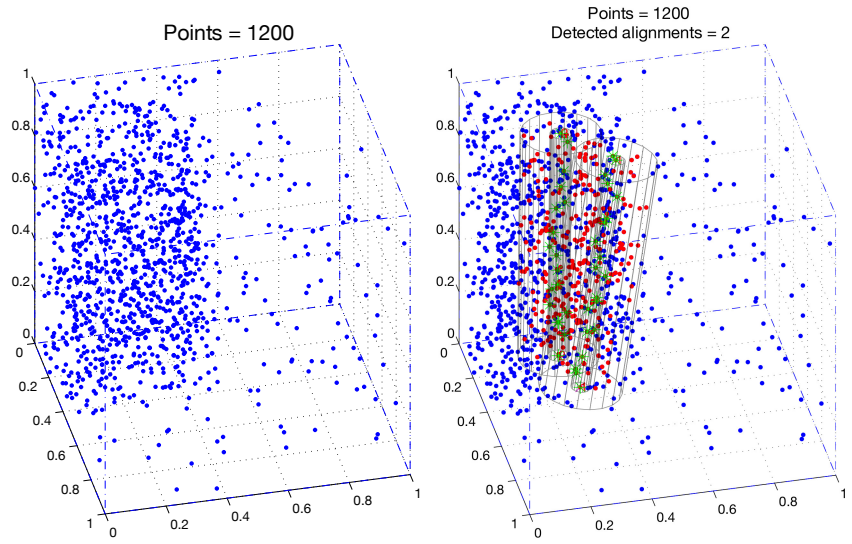

Figure 7.2: Detection of alignments in the border of two regions with different density. If the local density is estimated by counting in the whole local cylinder, the computed density in the border is low with respect to the density in the alignment cylinder leading to the deceptive detection of unexpected alignments. If seen in color, green points are part of the alignment and red points are part of the local cylinder but not of the alignment cylinder.

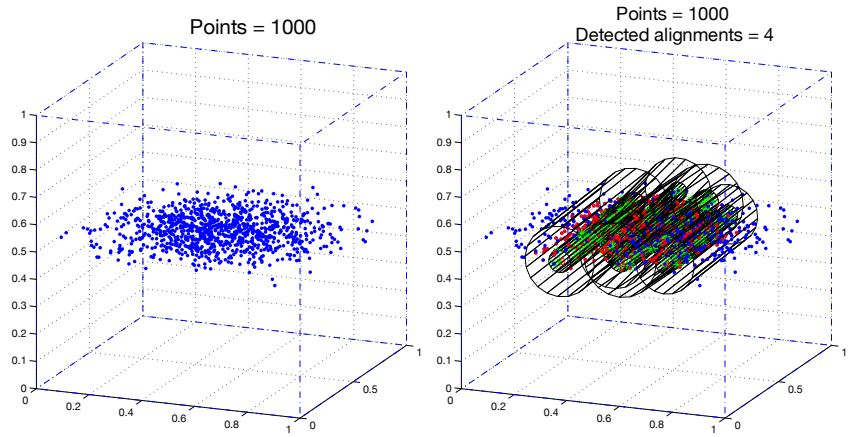

Figure 7.3: Detection of alignments in a flat region. If the local density is estimated by counting in the whole local cylinder, the computed local density is low with respect to the density in the alignment cylinder leading to the deceptive detection of unexpected alignments. If seen in color, green points are part of the alignment and red points are part of the local cylinder but not of the alignment cylinder.

cylinder. A conservative estimate of the local number of points can be computed as

$$n^*(R, \mathbf{x}) = S \cdot \text{máx}(M_1, \ldots, M_S) + M. \tag{7.2}$$

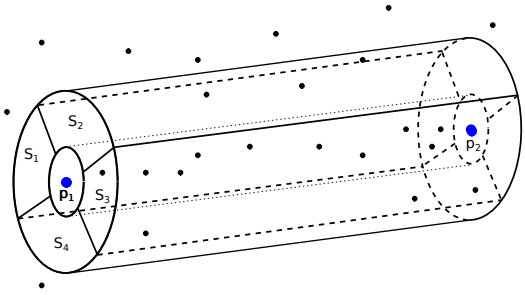

Figure 7.4: In order to conservatively estimate the local point density, the local cylinder is divided into sectors and the maximum count for the sectors is considered for the estimation. This approach is appropriate to avoid the unwanted detection of simple borders as alignments in structures such as corners or flat regions.

7.2.3. Point Regularity

As suggested in other studies [68,82,83], apart from the density with respect to the background, a significant alignment should also have regularly spaced points inside the candidate alignment cylinder.

This reasonable condition is called the law of *constant spacing* by Gestaltists and is an important factor that makes an alignment perceptually meaningful. The condition is also mandatory to avoid the detection of cluster-like structures as alignments. Figure 7.5 shows an example of this cluster-issue.

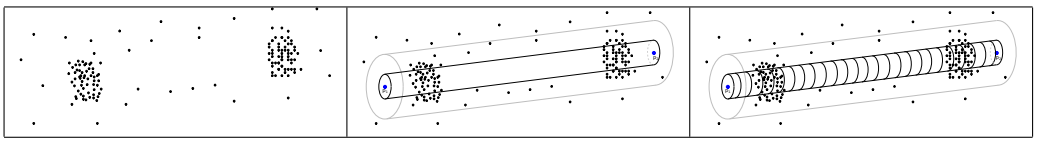

Figure 7.5: Left: Dot pattern with two point clusters but no alignment. Center: Although there is no "true.ªlignment, a thin cylinder with a high point density with respect to the local background may be found across the two clusters giving a false detection. Right: If the cylinder is divided into pill-boxes and only the occupation of the boxes is considered in the estimation, this misleading cluster effect can be avoided.

In order to evaluate this regularity the candidate cylinder is divided into equally shaped intervals or pill-boxes as shown in Figure 7.6. Instead of counting the total number of points in the alignment cylinder, the algorithm counts the number of boxes that are occupied by at least one point. We call them *occupied* boxes. In this way, the minimal NFA is attained when the points are perfectly distributed along the alignment. In addition, a concentrated cluster in the alignment has no more influence on the alignment detection than a single point in the same position.

A candidate alignment will be validated if the number of occupied boxes is statistically significant given the local point density. A set C of different values are tried for the number of boxes c into which the cylinder is divided. C can be roughly estimated as $C = \sqrt[3]{N}$ noticing that in a cube with N evenly distributed points, the longest alignments will have $\sqrt[3]{N}$ points.

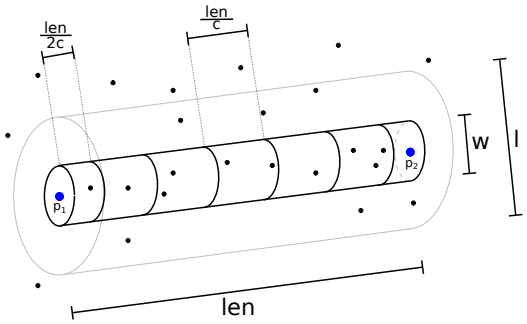

Figure 7.6: A schematic representation of the division of a candidate alignment cylinder into c pill-boxes. Two half boxes are placed at the extremes of the cylinder, and only the full boxes in the middle are count. This is because the two points forming the alignment must not be counted in the statistical test since they are the ones that define the test.

7.2.4. Candidate Validation

The validation is done using the a contrario framework. The a contrario methodology consists in assuming a null hypothesis H_0 for the data, where no detections should occur, and finding violations of this hypothesis (i.e. events that could hardly happen under the hypothesized model). In the case of this algorithm, we consider for the null hypothesis H_0 that the points are independent and uniformly distributed in the domain.

Number of Tests

For each pair of points in the domain a set of $W \times L \times C$ candidate alignments are considered where:

- **W** is the number of different diameters considered for the alignment cylinder,
- **L** is the number of different diameters considered for the local cylinder,
- **C** is the number of different values considered for the number of boxes into which the alignment cylinder is divided.

A candidate alignment is determined by a pair of points. Then, with N points in the domain, we will need to evaluate N_{tests} candidate alignments defined as

$$N_{tests} = W \cdot L \cdot C \cdot \frac{N(N-1)}{2}. \tag{7.3}$$

Non-Accidentalness / Number of False Alarms

A candidate alignment will be accepted based on two principles:

a) a good candidate should be non-accidental,

b) any equivalent or better candidate should be kept as well.

We will denote by $b(r, c, \mathbf{x})$ the observed number of occupied boxes in the candidate alignment cylinder r when divided into c boxes ($\mathbf{x}$ is the actual set of N points in the domain). Let $b(r, c, \mathbf{X})$ be the expected number of occupied boxes under the null hypothesis ($\mathbf{X}$ denotes a random set of N points following H_0).

The degree of non-accidentalness of a candidate alignment cylinder r can therefore be measured by how small the probability $\mathbb{P}\Big[b(r, c, \mathbf{X}) \geq b(r, c, \mathbf{x})\Big]$ is. In the same vein, a candidate r' will be considered at least as good as r given the observation $\mathbf{x}$, if $\mathbb{P}\Big[b(r', c, \mathbf{X}) \geq b(r', c, \mathbf{x})\Big] \leq \mathbb{P}\Big[b(r, c, \mathbf{X}) \geq b(r, c, \mathbf{x})\Big]$.

Given that N_{tests} candidates will be tested, the expected number of cylinders which are as good as r under H_0 is less than

$$\mathrm{NFA}(r, c, \mathbf{x}) = N_{tests} \cdot \mathbb{P}\Big[b(r, c, \mathbf{X}) \geq b(r, c, \mathbf{x})\Big]. \tag{7.4}$$

This fundamental quantity of the a contrario methodology is denoted the Number of False Alarms (NFA). It will be interpreted as a bound of the expected number of candidate cylinders containing enough *points* to be as rare as r under H_0. When the NFA associated with a candidate cylinder is large, this means that such an event is to be expected under the a contrario model and therefore is not relevant. On the other hand, when the NFA is small, the event is rare and probably meaningful. A rarity threshold ε must nevertheless be fixed for each application. Candidate cylinders with $\mathrm{NFA}(r, c, \mathbf{x}) \leq \varepsilon$ will be called *ε-meaningful cylinders* [23], constituting the detection result of the algorithm.

Calculating the NFA of a Candidate

We want to estimate the expected number of occupied boxes in the background model H_0. The probability of one point falling in one of the boxes is $p_0 = \frac{V_B}{V_L}$, where V_B and V_L are the volumes of the box and the local cylinder respectively. Then, the probability of having one box occupied by at least one of the $n^*(R, \mathbf{x})$ points (where $n^*(R, \mathbf{x})$ is the local number of points estimated in Equation (7.2)) can be computed as the complement to "none of the points fall in the box"

$$p_1(R, c) = 1 - (1 - p_0)^{n^*(R, \mathbf{x})}. \tag{7.5}$$

As mentioned before, we will denote by $b(r, c, \mathbf{x})$ the observed number of occupied boxes in the candidate cylinder r when divided into c boxes. Finally, the probability of having at least $b(r, c, \mathbf{x})$ of the c boxes occupied is

$$\mathcal{B}\big(c, b(r, c, \mathbf{x}), p_1(R, c)\big), \tag{7.6}$$

where $\mathcal{B}(n, k, p)$ is the tail of the binomial distribution

$$\mathcal{B}(n, k, p) = \sum_{j=k}^{n} \binom{n}{j} p^j (1 - p)^{n-j}. \tag{7.7}$$

The NFA of the alignment event is then

$$\begin{aligned}\mathrm{NFA}(r, R, c, \mathbf{x}) = \\ N_{tests} \cdot \mathbb{P}\Big[b(r, c, \mathbf{X}) \geq b(r, c, \mathbf{x}) \,\Big|\, n(R, \mathbf{X}) = n^*(R, \mathbf{x})\Big] \\ = \frac{N(N-1)}{2} WLC \cdot \mathcal{B}\big(c, b(r, c, \mathbf{x}), p_1(R, c)\big). \quad (7.8)\end{aligned}$$

The algorithm computes the NFA for each candidate alignment and validates only those with $NFA \leq \varepsilon$. Usually the value of ε is set to 1, which means that, in average, only one false detection will occur in a random data set.

Influence of the Number of Sectors

As mentioned in Section 7.2.2, the idea of using sectors is basically to avoid deceptive detections in the border between regions of different density. Ideally, n^* (Equation (7.2)) should give an estimate as close as possible to the number of points of the most dense region. Then, the ideal number of sectors depends on the type of structure where one wants to avoid unwanted detections. If only corner like structures matter, four to eight sectors can be a good choice. But wedge-like and flat structures will require more sectors. As shown in Figures 7.14 and 7.15, four sectors are good to avoid detections on the corner-like structure but are not enough to avoid detections on a quasi planar region.

The conservative estimate of the local number of points n^* as computed in Equation (7.2) always increases with the number of sectors. The estimation of n^* influences the probability p_1 (Equation (7.5)) and hence the probability of having at least $b(r, c, \mathbf{x})$ of the c boxes occupied given by the binomial tail (Equations (7.6) and (7.7)). As n^* increases with the number of sectors, p_1 increases and the tail of the binomial also increases. Then, the computed NFA (Equation (7.8)) will increase with the number of sectors. Eventually, as sectors increase, the NFA of a significant alignment will surpass the ε limit loosing the alignment detection. The number of sectors must be chosen taking into account the trade-off between removing unwanted detections and the chance of loosing some true alignments.

Figure 7.7 presents an example with two alignments in uniform noise and the evolution of the NFA with respect to the number of sectors used in the algorithm.

7.2.5. Redundancy Reduction

Given a very meaningful alignment, many smaller or larger cylinders overlapping the main alignment may also be meaningful as exemplified in Figure 7.8. This result is not desirable and redundancy should be eliminated in order to keep the most significant detections.

In this implementation, redundancy reduction is done in the same way as in [52] following the masking principle: *A meaningful structure B will be said "masked by a structure A" if B is no longer meaningful when evaluated without counting its building elements belonging to A*. For the implementation, the alignments detected

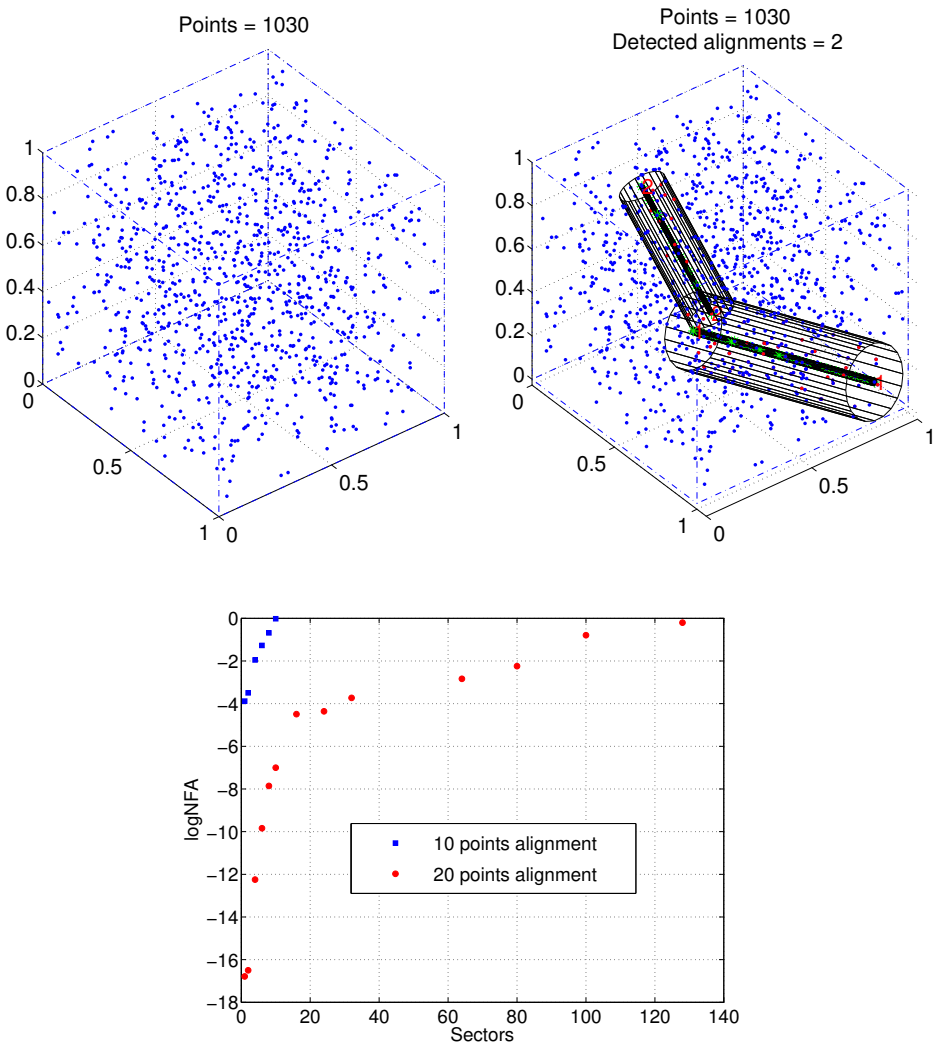

Figure 7.7: Influence of the number of sectors in the NFA of two alignments. Top-left: points, Top-right: the two detected alignments when one sector is used in the algorithm, Below: evolution of the $logNFA$ with the number of sectors. In this example, the detection of the less significant alignment is lost with more than ten sectors.

in the first stage of the algorithm (those with $NFA < \varepsilon$) are ordered by decreasing meaningfulness (increasing NFA). The first alignment in the ordered list is kept. The subsequent alignments are analyzed one by one and kept if it is checked that they are not masked by any one of the previously validated alignments. To check if an alignment B is masked by another alignment A, the algorithm recomputes the NFA of B using its same structure but removing the points belonging to A. If the new NFA is no longer significant ($NFA \nleq \varepsilon$), then B is said to be masked by A.

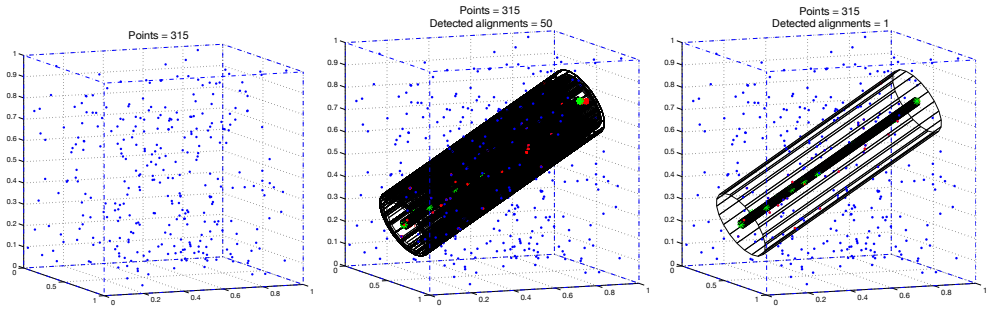

Figure 7.8: Left: An alignment in noise. Center: 50 most significant alignments. Right: The detected alignment after eliminating the redundant detections. If seen in color, green points are part of the alignment and red points are part of the local cylinder but not of the alignment cylinder.

7.3. Algorithm

7.3.1. Main Body

Algorithm 1: Main body of the algorithm

input : A set $\mathbf{x}$ of N points [$W = 8$, $L = 8$, C (set of values for the number of boxes), $\varepsilon = 1$, S(number of sectors)]

output: A list $\mathbf{m}$ of non-redundant point alignments

1 Create a symmetric extension of $\mathbf{x}$;
2 $\mathbf{l} = detect\_alignments(\mathbf{x})$;
3 $\mathbf{m} = redundancy\_reduction(\mathbf{l})$;

Algorithm 1 presents the main body of the algorithm. In the first step, the points are symmetrically extended outside the domain. This extension avoids having border problems when estimating point densities near the frontier of the domain. Figure 7.9 shows an example of a symmetrization.

In the second step, the N_{tests} tests are performed and the significant candidate cylinders (those with $NFA < \varepsilon$) are kept. The output from the second step may have redundant detections. The third step of the algorithm reduces the redundancy by applying the masking principle (Section 7.2.5).

7.3.2. Point Alignment Detection

Algorithm 2 presents the pseudocode for the detection of the candidate alignments. Figure 7.10 aids with the notation used in the algorithm description. Following are some additional comments:

- Line 4: The searched alignments are expected to be elongated structures. In this implementation, a minimum $\frac{len}{w}$ ratio of 10 is used.

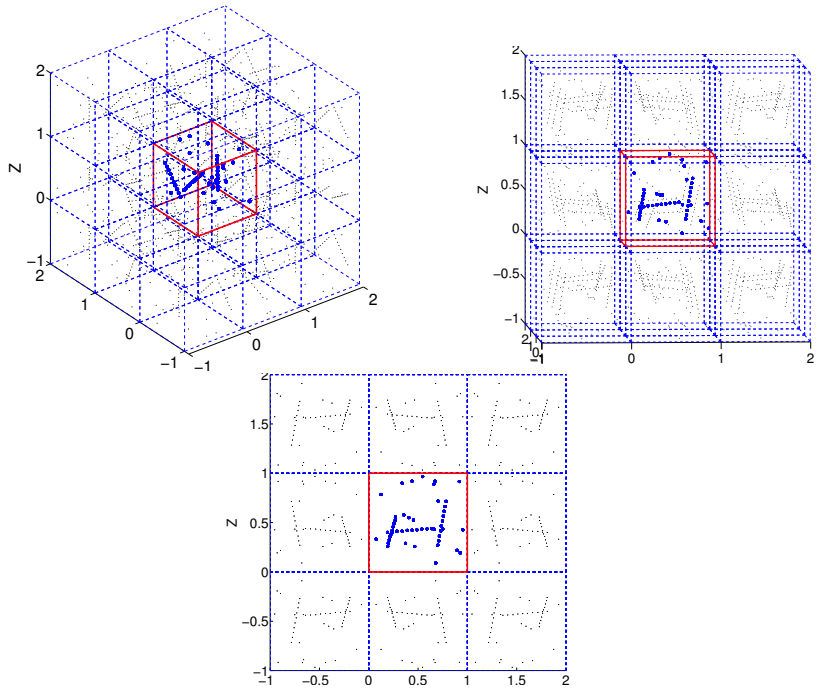

Figure 7.9: Three views of a symmetrization example. The N points in the domain are extended by symmetrization. This gives a set of $27 \times N$ points that allows to correctly estimate the densities near the frontier of the domain. If seen in color, blue points are the points in the domain and black points are the points extended by symmetrization outside the domain. The boundaries of the domain are outlined in red.

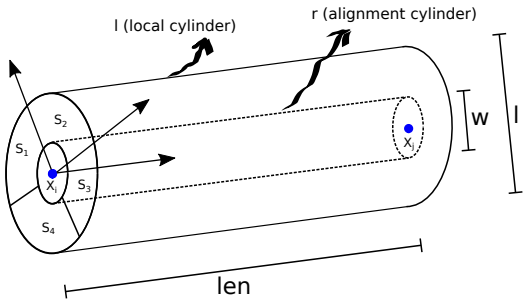

Figure 7.10: Schema of the variables used in Algorithm 2.

- Line 6: r is the alignment cylinder whose main axis is the segment $\overline{X_i X_j}$ and whose diameter is w.
- Line 7: The local cylinder R surrounds the alignment cylinder and is used to estimate the local background point density related to the tested alignment.
- Line 12: In practice, for each alignment cylinder, the set of values for the number of boxes that are considered is $C = \left[\frac{M}{2}, 2M\right]$ where M is the number of points in the alignment cylinder.

Algorithm 2: Point alignment detection

input : A set $\mathbf{X}$ of N points $\{X_0, \ldots, X_{N-1}\}$
$[W = 8,\ L = 8,\ C$ (set of values for the number of boxes), $\varepsilon = 1,\ S$(number of sectors)]
output: A list **align** of point alignments

```
1  for i = 0 to N − 1 do
2      for j = i + 1 to N − 1 do
3          len ← distance(X_i, X_j) // length of the alignment cylinder
4          w ← len/10 // initialize the diameter of the alignment
           cylinder
5          for iw = 1 to W do
6              r ← cyl(X_i, X_j, w) // r is the alignment cylinder with
               diameter w
7              l ← len/√10 //initialize the diameter of the local cylinder
               R
8              for il = 1 to L do
9                  Count M, the number of points in the alignment
                   cylinder r
10                 Count M_1 ··· M_S, the number of points in the sectors
                   of the local cylinder
11                 Compute n*(R, X) // conservative number of local
                   points, see Equation (7.2)
12                 for c ∈ C do
13                     Divide r into c pill-boxes (box = len/c)
14                     p_0 ← box·w^2/(len·l^2) //prob. of one point of R falling in a
                       pill-box
15                     Compute p_1(r, R, c) //see Equation (7.5)
16                     Count b(r, R, c, X), the number of occupied boxes
17                     Compute NFA(r, R, c, X) //see Equation (7.8)
18                     if NFA(r, R, c, X) ≤ ε then
19                         align.append(r)
20                     end
21                 end
22                 l ← l/∜2
23             end
24             w ← w/∜2
25         end
```

- Line 13: The cylinder is divided into c pill-boxes leaving two half-size boxes at each extreme of the alignment as show in Figure 7.10. There are two reasons for this. One is that the points defining the statistical test must

not be counted in the test. The second reason is that if there are c points perfectly spaced inside the alignment, then these would fall exactly in the center of each of the c boxes.

- Line 14: The probability of one point of the local cylinder falling in one box is given by the ratio of volumes $p_0 = \frac{V_{box}}{V_r} = \frac{box \cdot w^2}{len \cdot l^2}$.

7.3.3. Redundancy Reduction

Algorithm 3 shows the pseudocode for the redundancy reduction procedure. The procedure deals with the list of alignment candidates and applies the masking principle. The pseudocode is the same as in [52] for the 2D case.

Algorithm 3: Redundancy reduction

input : A list **l** of all significant alignments
output: A list **m** of maximally significant alignments

```
1  m ← [ ]; // initialize the output list
2  if isempty(l) then
3      return;
4  l ← sort(l); //sort l by ascending NFA (decreasing significance)
5  m.append(l[0]); // keep the most significant alignment from the
     sorted list
6  //check each alignment in l against the already validated alignments
     in m
7  for i = 1 to length(l) − 1 do
8      B = l[i]; //the alignment to check
9      masked ← False
10     for j = 0 to length(m) − 1 do
11         A = m[j]; //an already validated alignment
12         X' = {x|x ∉ r_A}; //consider only the points that are not in
             the alignment cylinder of A
13         //check the significance of B excluding the points from the
             alignment A
14         if NFA(r_B, R_B, c_B, X') > ε then
15             masked ← True;
16             break;
17     // if B is not masked by the already validated alignments, then
         add it to the output list
18     if masked == False then
19         m.append(B)
```

7.3.4. Computational Complexity

In the first step (Algorithm 2), an exhaustive search is performed to detect the candidate alignment cylinders. A total of $N_{tests} = W \cdot L \cdot C \cdot \frac{N(N-1)}{2}$ is performed. The number of tests is then $O(\sqrt[3]{N}N^2)$ since the number of tested boxes can be estimated as $C = \sqrt[3]{N}$.

Each test on an alignment implies counting the points in the alignment cylinder and in the local cylinder (steps 9 and 10 in Algorithm 2).

In the simplest approach, where no special data structure nor nearest neighbor search is used, for an alignment defined by two points, all the other $N-2$ points in the domain must be tested to see if they are part of the alignment or the immediate local vicinity. Moreover, since we also have symmetrization of the points outside the domain, for the alignments that lie near the boundary of the domain an extra scale factor must be considered (up to $27 \times N$). Finally, each test contributes an N factor and the total complexity of the first step is $O(\sqrt[3]{N}N^3)$

A reduction of the N factor can be achieved if, for each alignment, only the neighboring points (in a certain radius depending on the maximum tested background cylinder) are considered. Although the speed up is not easy to determine analytically, this approach significantly reduces the factor in the complexity and enables, in practice, to run the algorithm on larger sets of points.

The source code provided with this article implements both approaches. For the nearest neighbors approach, the Fast Library for Approximate Nearest Neighbors (FLANN) [62] is used.

In the redundancy reduction step (Algorithm 3), a list of final detections is constructed and the alignments of the first step are tested to see if they are masked or not by the already selected final alignments. The complexity of each test is once again $O(N)$. The number of tests is much smaller than in the first step since each validated alignment from the first step is only checked for masking against the already selected as final detections.

7.4. Experiments

7.4.1. Experiments on Synthetic Data

Figure 7.11 shows the results of the algorithm when the points are uniformly distributed random points in the domain. This is the case where the points are in the H_0 hypothesis. Since the NFA of the alignments is limited to $\varepsilon = 1$, at most one false detection is expected to occur in this random datasets. In the examples of Figure 7.11 no detections were found.

Figure 7.12 shows the results of the algorithm when the points have a Gaussian distribution and the points are spread in the whole domain ($\mu = [0{,}5, 0{,}5, 0{,}5]^t$, $\sigma = [0{,}2, 0{,}2, 0{,}2]^t$). Although this distribution does not follow the a contrario model no alignment is detected.

Figures 7.13 and 7.14 present the case where the points are distributed on regions of different density. In this case alignments can be detected in the interface

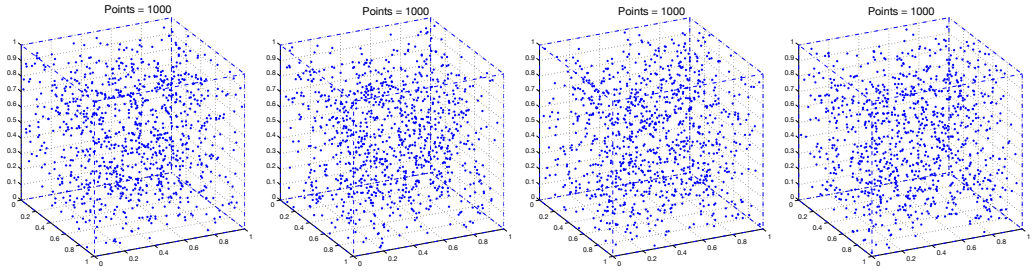

Figure 7.11: Uniformly distributed points.

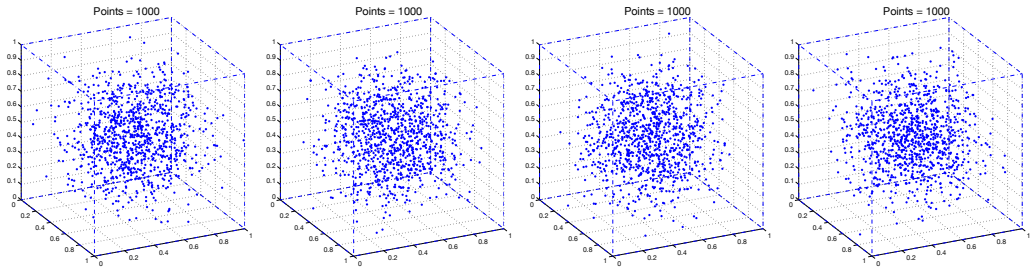

Figure 7.12: Gaussian distributed points.

between regions if only one sector is considered in the local cylinder. As explained in Section 7.2.2, with only one sector, the background estimation can be underestimated since a big portion of the local cylinder intersects a low density region. The use of sectors allows the conservative estimation of the local density and helps overcome this problem.

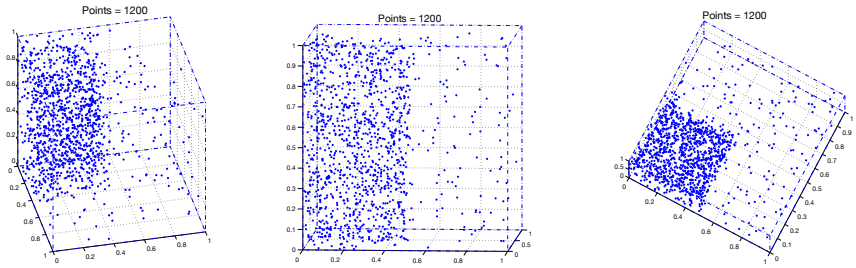

Figure 7.13: Points distributed in regions of different density. Three different views of the same set of points.

An extreme case for the local density estimation arises when the points group in nearly planar regions. In this case the distribution of the points is clearly far from the underlying hypothesis of the method. However, spurious detections in this kind of regions can be avoided using a larger number of sectors to estimate the local density. Figure 7.15 shows the results on several experiments with nearly planar regions of points. Figures 7.16, 7.17, 7.18 present the detection of true alignments for several point distributions.

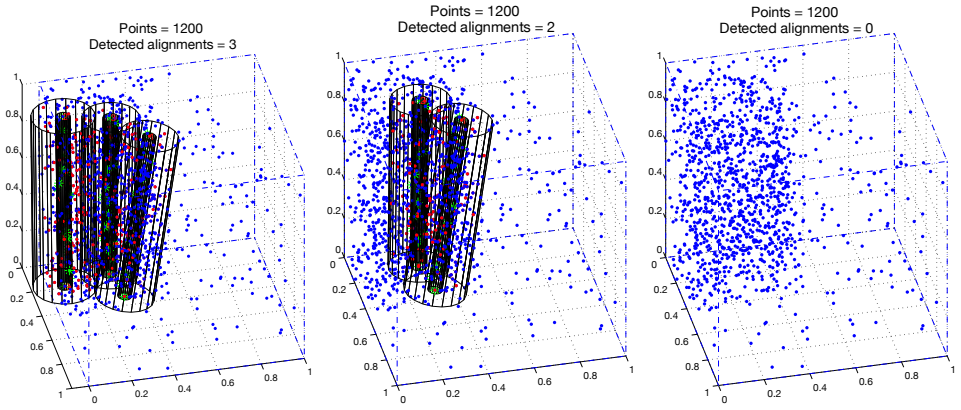

Figure 7.14: Alignments detected on the points of Figure 7.13 using different number of sectors. From left to right: detections computed using 1, 2 and 4 sectors. If seen in color, green points are part of the alignment and red points are part of the local cylinder but not of the alignment cylinder.

7.4.2. Sensitivity of the Alignment Significance

For a particular alignment defined by its extreme points, the significance (in terms of the NFA of the alignment) was computed for different amounts of jitter on the interior points. Figure 7.19 and 7.20 show examples of different amount of off-axis and on-axis jitter on the interior points of an alignment.

For the off-axis jitter, starting from an ideal alignment, the interior points are allowed to move away from the axis a distance drawn from a normal distribution $N(0, (len \times jitter)^2)$.

For the on-axis jitter, each interior point is allowed to move along the axis following a uniform distribution $\pm jitter \times ideal\_distance\_between\_points$ centered on the ideal position of the point (the position if the alignment were perfect and the space between points constant). Note that an on-axis jitter of 1 allows each point to move to any position between the ideal positions of its neighbors.

Figures 7.21 and 7.22 present the results (values and standard deviations) for the sensitivity with respect to the off-axis and on-axis jitter. An alignment of 10 points was considered and 20 experiments were performed for each value of jitter.

Off-axis jitter does not change the position of the points with respect to the boxes in the alignment so the number of boxes remains stable. In the on-axis jitter, when jitter is above 0,5, the points may move close to other points so the number of boxes in the alignment may drop.

The NFA value increases (significance decreases) with jitter in both cases but it is, as expected, influenced more by off-axis jitter where the points may move away from the ideal line.

7.4.3. Experiments on Acquired Images

Figure 7.23 presents a point cloud acquired with a Kinect sensor. In the cloud there is a person holding linear objects. The original cloud with 59290 points is

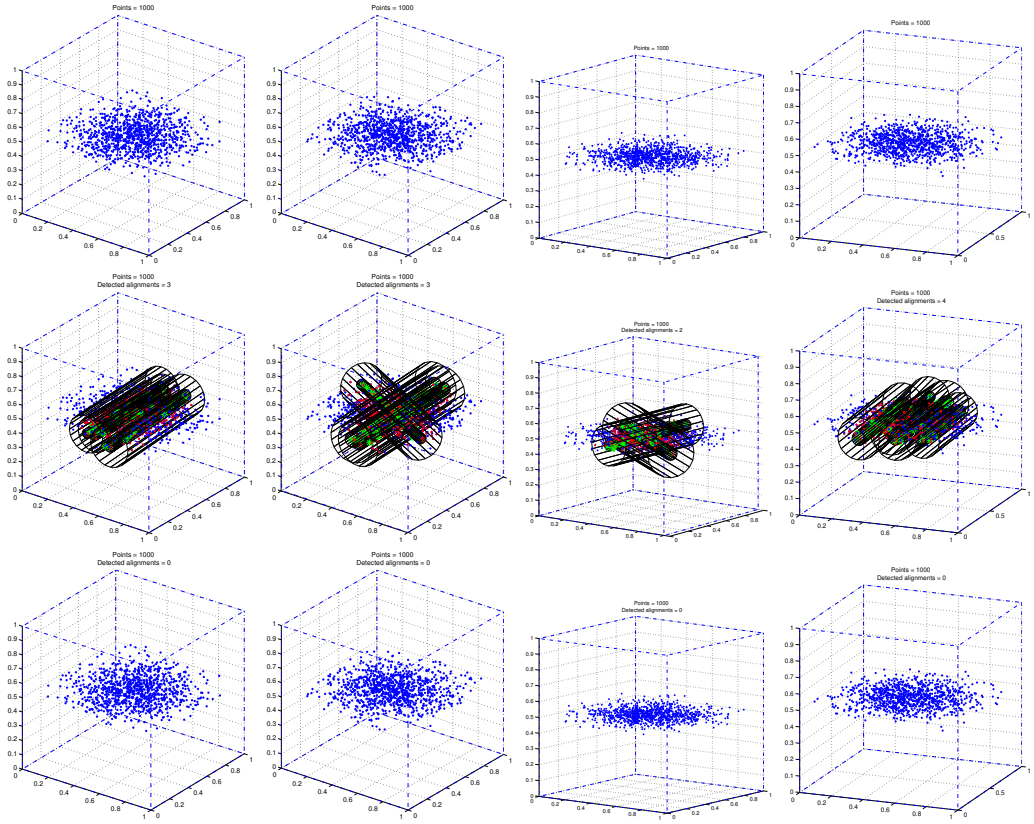

Figure 7.15: Points grouped in a nearly planar region. The first row shows four experiments where the points follow a Gaussian distribution with $\mu = [0{,}5, 0{,}5, 0{,}5]^t$ and $\sigma = [0{,}2, 0{,}2, 0{,}02]^t$. Second and third rows show the detections with 4 and 180 sectors respectively. If seen in color, green points are part of the alignment and red points are part of the local cylinder but not of the alignment cylinder.

downsampled to 501 points in order to apply the alignment detector. Downsampling is performed in a voxel grid where all the points in a voxel are approximated with their centroid. The detected alignments are shown in Figure 7.24.

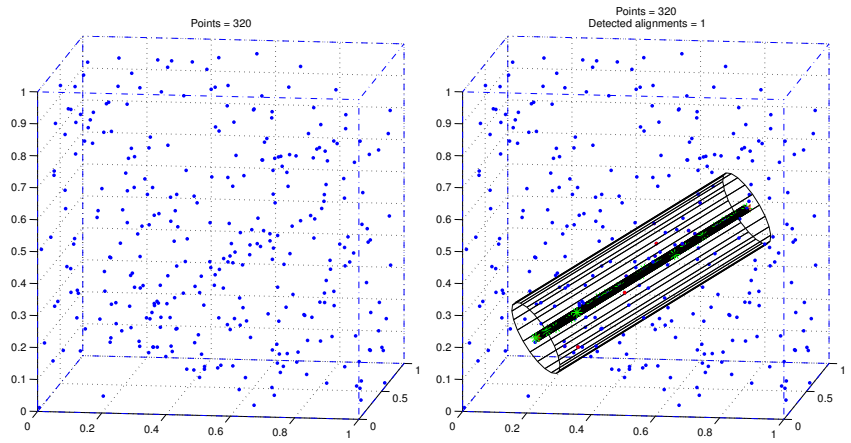

Figure 7.16: Detection of an alignment in noise. Left: an alignment in uniform noise, Right: detection.

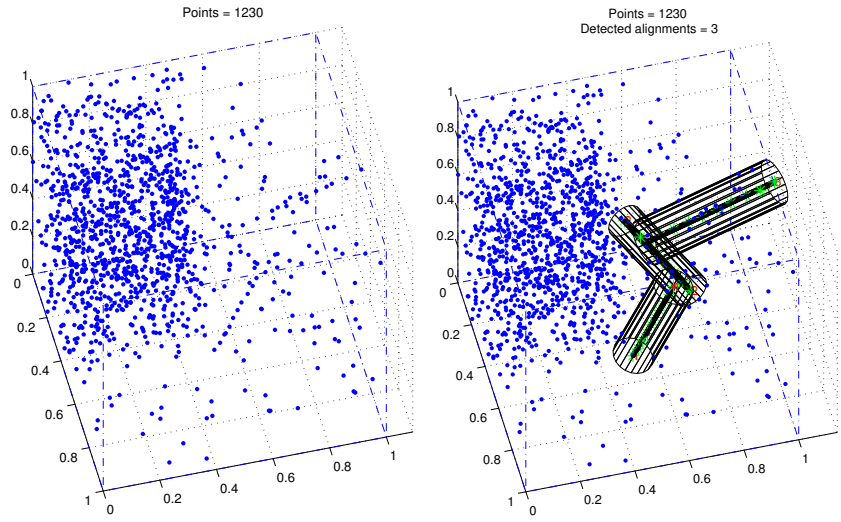

Figure 7.17: Detection of true alignments. Left: points uniformly distributed in two regions of different density and 3 true alignments, Right: detections with 8 sectors. If seen in color, green points are part of the alignment and red points are part of the local cylinder but not of the alignment cylinder.

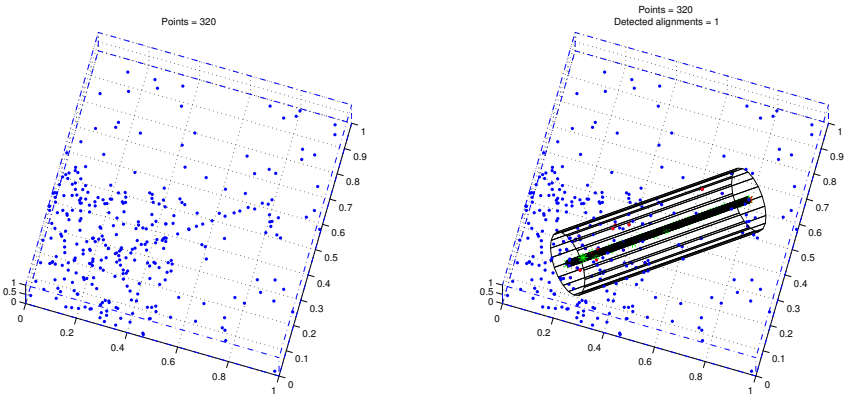

Figure 7.18: Detection of an alignment across two regions of different density. Left: uniform points in two regions of different density and one true alignment, Right: detection. If seen in color, green points are part of the alignment and red points are part of the local cylinder but not of the alignment cylinder.

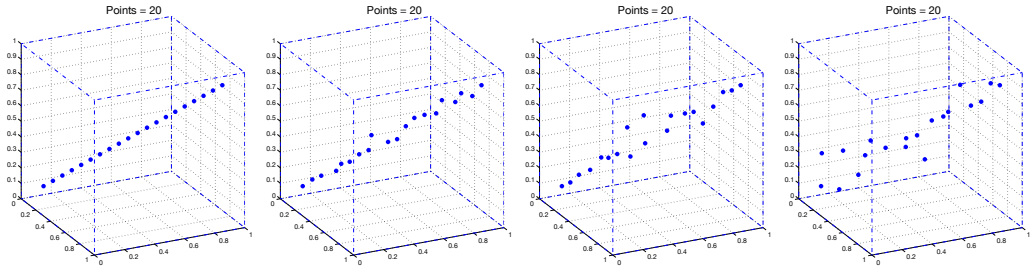

Figure 7.19: Off-axis jitter. From left to right: jitter=0 (perfect alignment), jitter=$0{,}02$, jitter=$0{,}05$, jitter=$0{,}08$.

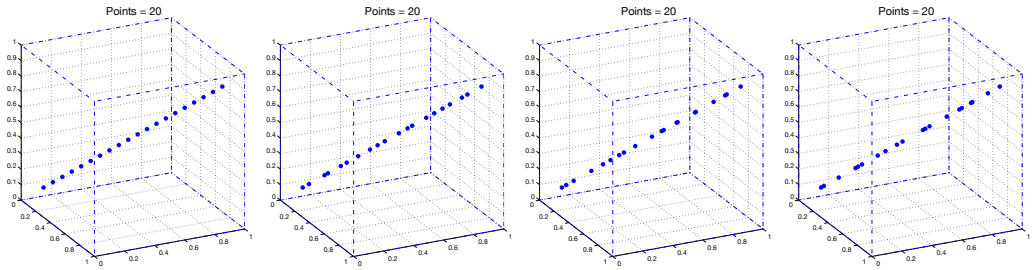

Figure 7.20: On-axis jitter. From left to right: jitter=0 (perfect alignment), jitter=$0{,}5$, jitter=$0{,}9$, jitter=$1{,}2$.

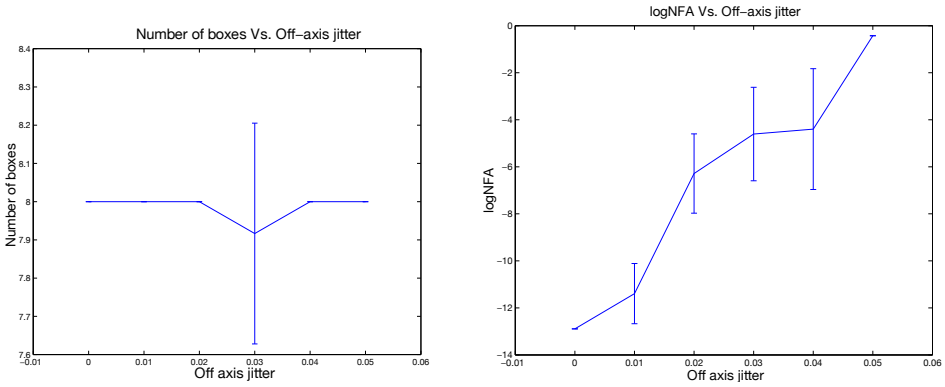

Figure 7.21: Off-axis jitter results.

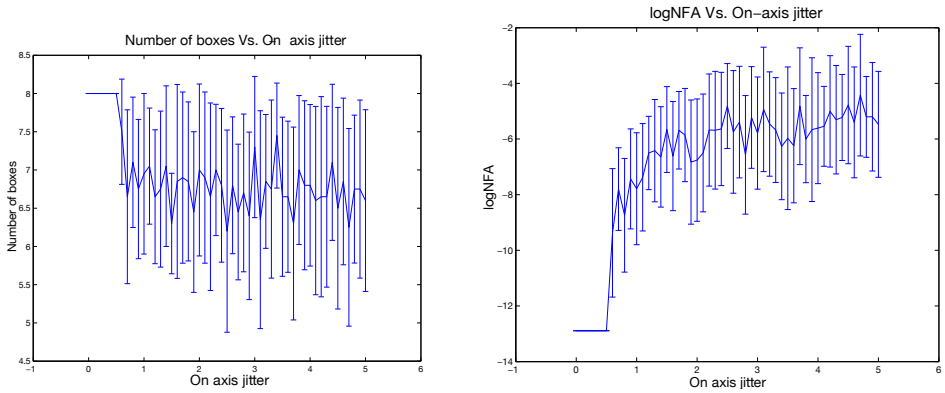

Figure 7.22: On-axis jitter results.

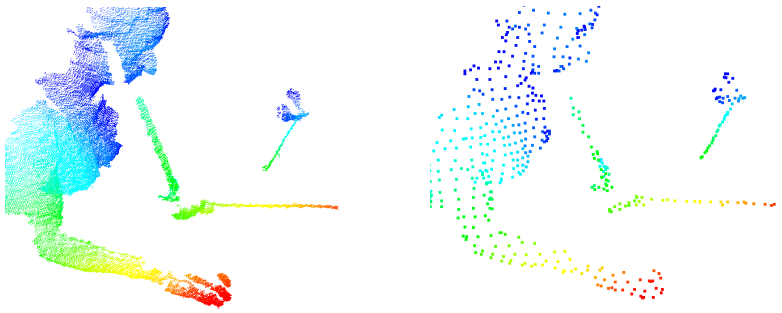

Figure 7.23: Point cloud acquired with a Kinect. Left: full point cloud of 59290 points. Right: Downsampled point cloud with 501 points.

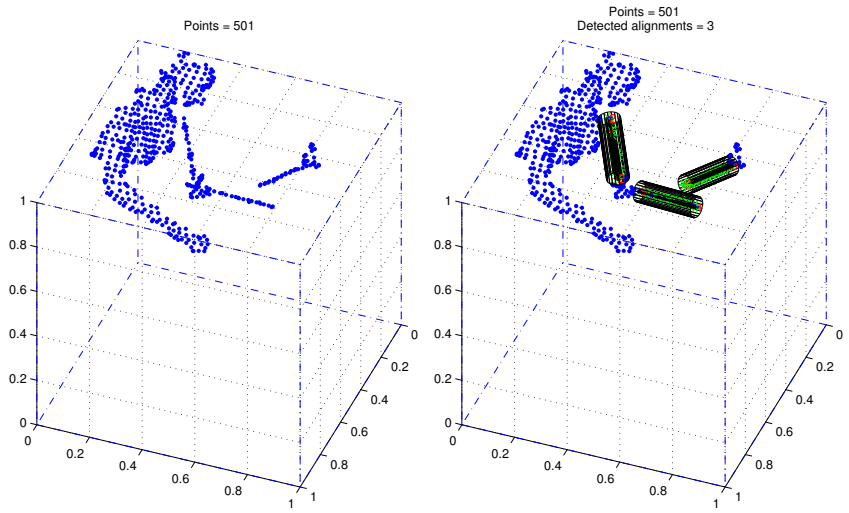

Figure 7.24: Alignments detected on data acquired with a Kinect sensor. The point cloud is the same as in Figure 7.23 right. If seen in color, green points are part of the alignment and red points are part of the local cylinder but not of the alignment cylinder.

Chapter 8

Conclusion

In this thesis we analyzed a satellite stereo pipeline and studied its constituent stages. From the selection of pairs at the entrance of the pipeline to the fusion of information at the output, several of the key stages of the pipeline were analyzed and some alternative methods proposed in order to enhance the end-to-end reconstruction performance. With the studied methods, reconstruction improvements were achieved to a greater or lesser extent. Notwithstanding, with each small improvement, many questions and challenges aroused that could be interesting to address in future work.

The comparison of MVS methods in Chapter 3, showed that a true MVS method such as CasMVSNet achieved results close to the pair-wise MVS baseline without a specific fine-tuning on satellite images. Fine-tuning seems a natural follow-up but the scarcity of public appropriate and ready to use satellite datasets poses restrictions to the task.

Apart from the necessary fine-tuning on satellite imagery, DL methods such as CasMVNet could probably perform better on satellite images if they had a view selection strategy as the classic MVS methods do [77]. It would be worth exploring the ways of adding a learnable view selection for CasMVNet.

The implemented simulation tool presented in Chapter 5 was used in this thesis to develop a pair selection strategy. In that case, the tool was used to generate stereo pairs from multiple orientations of a single scene. The tool could also be used to complement the few available real datasets and/or to generate a synthetic dataset for training the DL algorithms. Since the tool is 3D scene agnostic, the only new requirement would be to have enough realistic 3D scenes of different types to be used for the simulations.

The simulator tool uses, in its current implementation, an affine camera model that is valid, as a satellite projection, only to generate images covering small regions. The modifications of the tool to generate large images with the corresponding real-like RPCs should be tackled in future work. The small currently

generated images are appropriate to work with pipelines or methods that use tiling strategies. The generation of big images and non-affine RPCs could render the tool more general and useful for the remote sensing community.

In pair-wise MVS the selection of the most appropriate pairs has been traditionally tackled by heuristics based on the metadata of the pairs. An alternative selection method based on the simulation of images is proposed that could be useful to select the pairs in a more consistent way than the heuristics. While the new method is a step forward in the selection criteria, it is still an ordering of pairs by their individual alleged quality and does not take into account the important complementary characteristics of sets of pairs. More appropriate pairs for pair-wise MVS or a better set of images for true MVS will surely be selected when considering this aspect. The exhaustive analysis of the best sets of images is not feasible as it leads to a combinatorial explosion, but a study restricted to small sets could be investigated. Once again, the simulation tool could be useful for this task.

Experiments of pair selection in chapter 6 showed that it is important to consider the sun orientation and also to use the AUCC metric when evaluating the reconstructions of the simulated pairs. The challenge is to study how to reformulate the quality maps to take into account all these valuable findings, while maintaining a manageable computational volume and data size.

Regarding DSM fusion, the iterative scheme based on the bilateral filter showed to be a robust method for the fusion of pair-wise DSMs. It is well known that the naif implementation of the bilateral filter has an important computational burden but on the other hand fast implementations are available. For routine use of the DSM fusion method, a fast implementation should be coded.

Appendix A

An overview of GANet

Guided Aggregation Net for End-to-end Stereo Matching (GANet) is a stereo matching method that uses Deep Neural Networks (DNN) to compute a disparity map from a pair of images of a scene. As other classic and DNN stereo methods, it follows the traditional stereo steps: dense features are extracted from both images, the cost of matching the features at different disparities is organized in a Cost Volume (CV) which is regularized by aggregation and local filtering and finally a map with minimal cost is derived from the CV. In GANet, the aggregation of the CV is done by a Semi-Global Guided Aggregation layer (SGA) which implements a differentiable approximation of the well known Semi-Global Matching (SGM) algorithm. SGA is followed by a Local Guided Aggregation layer (LGA) that performs a local filtering. SGA and LGA weights are generated by an auxiliary guidance subnet fed with the original reference image and its extracted features. This chapter presents an overview of GANet. An online demo, running on CPU, is made available.

A.1. Introduction

Most stereo algorithms perform these four steps: (1) matching cost computation, (2) cost aggregation, (3) disparity computation, (4) disparity refinement.

The first step implies finding sparse or dense correspondences between the images. In the sparse case, characteristic points along with their local features are extracted and compared. In the dense approach, image patches in both images are compared computing the cost of matching the patches for different possible disparities. The search of corresponding patches is simplified by the geometric constraints of the stereo pair (epipolar constraints). Instead of a 2D search for correspondences, the epipolar constraints restrict the search for corresponding image points from the entire image plane to a single line. Moreover, the images can be resampled (stereo-rectification) in such a way that corresponding points are located on the same row.

The matching information is organized usually in a cost volume that stores the costs $C_p(d)$ of matching the position p of the reference image with $p + d$ in the

second image for all the considered possible disparity values d.

Matching at the correct disparity is challenging in real life due to the photometric and geometric distortions introduced by the change of viewpoint and by ambiguities due to occlusions, low texture or repetitive patterns in the scene. The step of cost aggregation tries to overcome this difficulty by imposing spatial coherence to the matching. This can be done by a simple local filtering of the cost volume or, in a more comprehensive approach, by formulating a global energy minimization problem with a regularization term that enforces the regularity of the disparity map.

Once the cost volume has been regularized, the disparity values can be estimated by processing the volume using argmin (usually mentioned as winner-takes-all), soft-argmin or a maximum a posteriori approximation.

The resulting disparity can be post-processed to filter erroneous values and fill-in missing values.

A.1.1. Global Energy Minimization Methods

This section presents an overview of global energy minimization methods based on [27], where the reader is referred to for more details.

Global methods formulate stereo matching as a global energy minimization problem that includes a regularity term. The energy E is defined on the graph $G = (\mathcal{V}, \mathcal{E})$

$$E(\mathrm{D}) = \sum_{\mathbf{p} \in \mathcal{V}} C_{\mathbf{p}}(\mathrm{D}_{\mathbf{p}}) + \sum_{(\mathbf{p},\mathbf{q}) \in \mathcal{E}} V(\mathrm{D}_{\mathbf{p}}, \mathrm{D}_{\mathbf{q}}), \tag{A.1}$$

where $C_{\mathbf{p}}(d)$ is a unary data term that represents the pixel-wise cost of matching $\mathbf{p}$ with disparity $d \in \mathcal{D}$ (the cost volume), where $\mathcal{D} = \{d_{\min}, \cdots, d_{\max}\}$ defined on a discrete search space (often denoted label set). 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 $\mathcal{E}$, which is usually the 4-connected image graph. Popular choices of regularity are

$$V(d, d) = |d - d'|, \tag{A.2}$$

or

$$V(d, d') = \begin{cases} 0 & \text{if } d = d' \\ P1 & \text{if } |d - d'| = 1 \\ P2 & \text{otherwise} \end{cases} . \tag{A.3}$$

The latter imposes a small penalty $P1$ for small jumps in disparity (up to one pixel), which are common on slanted surfaces, and a constant penalty $P2$ (with $P2 > P1$) accounts for larger disparity jumps.

The exact minimization of energy (A.1) on a 2D graph is NP-hard, except for some particular cases [43, 72].

On the other hand, when defined on acyclic graphs, the energy (A.1) can be minimized exactly in polynomial time using dynamic programming.

Tree-based dynamic programming approaches allow to incorporate more regularity (illustrated in Figure A.1), leading to better approximations of the problem (A.1). Some methods build a single tree that spans the entire image [84].

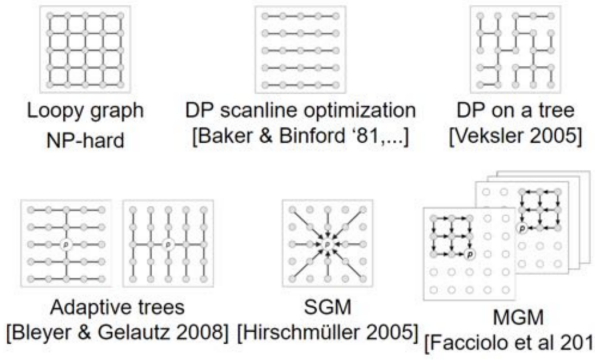

Figure A.1: Approximations of the 2D MRF energy using trees [5, 7, 28, 41, 84]. Reproduced from [27].

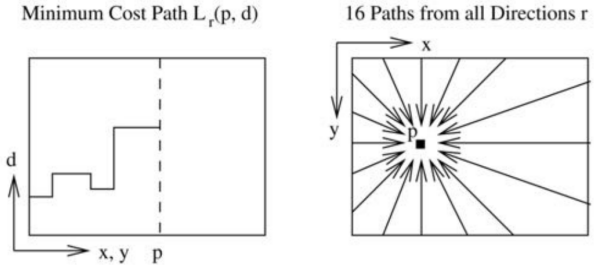

Figure A.2: Semi-Global matching aggregates the results of scanline optimization performed along 8 or 16 different orientations. This is equivalent to solving the problem restricted to a star-shaped graph associated to each pixel. Figures reproduced from [41]. Caption text reproduced from [27].

Others construct trees that vary their grid structure with the position of the pixel [7, 28, 41]. The Semi-Global Matching (SGM) algorithm [41] is equivalent to optimizing an energy restricted to a star-shaped graph centered at the current pixel. Even though these algorithms do not yield the most accurate reconstructions, they produce very fast and high-quality results.

A.1.2. Semi-Global Matching Algorithm

Semi-Global matching [41] proposes to approximately minimize energy (A.1) with the smoothness term of (A.3). Semi-Global matching approximation consists in dividing the grid-shaped problem into multiple (N_{dir}) one-dimensional problems defined on scanlines, which are straight lines that run through the image in 4, 8 or 16 cardinal directions (illustrated in Figure A.2). For simplicity, here we will consider only $N_{dir} = 4$ directions.

For each cardinal direction $\mathbf{r} \in \{(1,0), (-1,0), (0,1), (0,-1)\}$ SGM computes a matrix of costs $C^A_{\mathbf{r}}$. The costs $C^A_{\mathbf{r}}(\mathbf{p}, d)$ are computed recursively starting from the image borders along a path in the direction $\mathbf{r}$

$$C^A_{\mathbf{r}}(\mathbf{p}, d) = C_{\mathbf{p}}(d) + \min_{d' \in \mathcal{D}}(C^A_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d') + V(d, d')). \tag{A.4}$$

This recursion is in fact a dynamic programming algorithm that solves the problem restricted to the directed graph induced by the scanline $\mathbf{p} - \mathbb{N}\mathbf{r} = \{\mathbf{p} - k\mathbf{r} | k \in \mathbb{N}\}$.

In the case of SGM, with the regularity term as in (A.3), the aggregated cost volume along each of the directions can be computed as

$$C^A_{\mathbf{r}}(\mathbf{p}, d) = C(\mathbf{p}, d) + \min \begin{cases} C^A_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d), \\ C^A_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d-1) + P_1, \\ C^A_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d+1) + P_1, \\ \min_i C^A_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, i) + P_2. \end{cases} \tag{A.5}$$

These costs computed in each direction $\mathbf{r}$ are then added to obtain an aggregated cost volume

$$S(\mathbf{p}, d) = \sum_{\mathbf{r}} C^A_{\mathbf{r}}(\mathbf{p}, d) - (N_{dir} - 1) C_{\mathbf{p}}(d). \tag{A.6}$$

The subtraction of $(N_{dir} - 1)C_{\mathbf{p}}(d)$ is an over-counting correction analogous to the correction proposed by Drory et al. in [25] and that is not present in the original SGM description [41].

The final disparity for each pixel is then selected by winner-takes-all with respect to d on the aggregated cost $S(\mathbf{p}, d)$. This amounts to minimizing a different problem at each pixel defined as a restriction of energy (A.1) to the star-shaped graph illustrated in Figure A.2.

A.2. GANet Method

The method addressed in this chapter, Guided Aggregation Net for End-to-end Stereo Matching (GANet) [88] is a stereo matching method that uses Deep Neural Networks (DNN) to compute a disparity map. Figure A.3 depicts the architecture overview.

As other DNN methods [49] it follows the traditional stereo steps: dense features are extracted from both images, the cost of matching the features at different disparities is organized in a Cost Volume (CV), which is regularized by aggregation and local filtering and finally a map with minimal cost is derived from the CV.

In most DNN based stereo methods, cost aggregation is done by 3D convolutions, usually in an hourglass configuration [49]. 3D convolutions imply large memory requirements; the computational burden restricts the size of the images that can be processed.

GANet, despite using also some 3D convolutions, takes a different approach for the aggregation by introducing a Semi-Global Guided Aggregation layer (SGA) which implements a differentiable approximation of Semi-Global Matching (SGM) [42]. SGA is followed by a Local Guided Aggregation layer (LGA) that performs a local

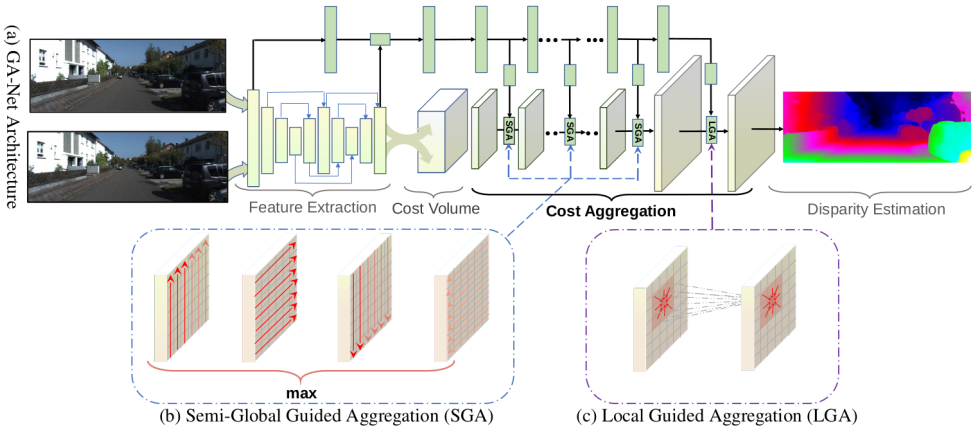

Figure A.3: GANet architecture overview.
Reproduced from [88].

filtering. SGA and LGA weights are generated by an auxiliary "guidance subnet" fed with the input reference image and its extracted features.

A.2.1. Semi-Global Guided Aggregation (SGA)

Inspired by SGM, GANet introduces the SGA step which supports backpropagation. The SGA step that aggregates along a direction is

$$C_{\mathbf{r}}^{A}(\mathbf{p},d) = C(\mathbf{p},d) + \text{sum}\begin{cases} \mathbf{w}_1(\mathbf{p},\mathbf{r}) \cdot C_{\mathbf{r}}^{A}(\mathbf{p}-\mathbf{r},d), \\ \mathbf{w}_2(\mathbf{p},\mathbf{r}) \cdot C_{\mathbf{r}}^{A}(\mathbf{p}-\mathbf{r},d-1), \\ \mathbf{w}_3(\mathbf{p},\mathbf{r}) \cdot C_{\mathbf{r}}^{A}(\mathbf{p}-\mathbf{r},d+1), \\ \mathbf{w}_4(\mathbf{p},\mathbf{r}) \cdot \max\limits_{i} C_{\mathbf{r}}^{A}(\mathbf{p}-\mathbf{r},i). \end{cases} \tag{A.7}$$

and presents several differences with respect to (A.5).

The main difference with the SGM approach is that the weights are learnt and hence adaptive and more flexible compared to the user-defined parameters from (A.3). Other changes can be noted between (A.5) and (A.7): (a) the outer min is changed to a weighted sum making the step all convolutional, (b) noting that the learning target of GANet is to maximize the probabilities at the ground truth depths and not to directly minimize the matching costs, the authors also change the inner min to a max.

Considering that the sum on a path can lead to large values, the weights are normalized. In practice, (A.7) is finally implemented as

$$C_{\mathbf{r}}^{A}(\mathbf{p},d) = \text{sum}\begin{cases} \mathbf{w}_0(\mathbf{p},\mathbf{r}) \cdot C(\mathbf{p},d), \\ \mathbf{w}_1(\mathbf{p},\mathbf{r}) \cdot C_{\mathbf{r}}^{A}(\mathbf{p}-\mathbf{r},d), \\ \mathbf{w}_2(\mathbf{p},\mathbf{r}) \cdot C_{\mathbf{r}}^{A}(\mathbf{p}-\mathbf{r},d-1), \\ \mathbf{w}_3(\mathbf{p},\mathbf{r}) \cdot C_{\mathbf{r}}^{A}(\mathbf{p}-\mathbf{r},d+1), \\ \mathbf{w}_4(\mathbf{p},\mathbf{r}) \cdot \max\limits_{i} C_{\mathbf{r}}^{A}(\mathbf{p}-\mathbf{r},i). \end{cases} \quad \text{s.t.} \quad \sum_{i=0,1,2,3,4} \mathbf{w}_i(\mathbf{p},\mathbf{r}) = 1. \tag{A.8}$$

A.2.2. Network Architecture

Figure A.4 shows the main blocks of the GANet architecture and Table A.1 lists their layers and parameters for the "GANet-deep" model.

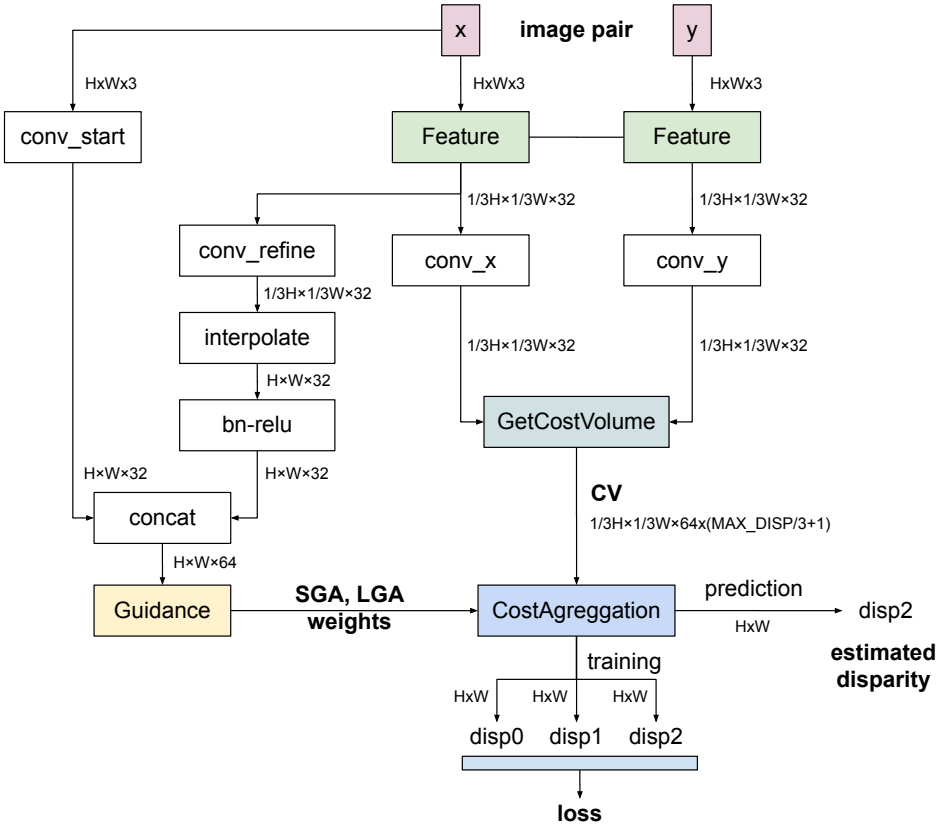

Figure A.4: GANet architecture overview. The main blocks of the net are depicted in color.

Table A.1: Network layers of the main blocks of the "GANet Deep" model.

Layer id	Inputs	Layer description	Output tensor	Output
Feature extraction				
input		image	H×W×3	
1	image	conv	H×W×32	
2	1	conv	1/3H×1/3W×32	
3	2	conv	1/3H×1/3W×32	
4	3	conv	1/6H×1/6W×48	
5	4	conv	1/12H×1/12W×64	
6	5	conv	1/24H×1/24W×96	
7	6	conv	1/48H×1/48W×128	
8	7,6	deconv / concat / conv	1/24H×1/24W×96	
9	8,5	deconv / concat / conv	1/12H×1/12W×64	
10	9,4	deconv / concat / conv	1/6H×1/6W×48	
11	10,3	deconv / concat / conv	1/3H×1/3W×32	
12	11,10	deconv / concat / conv	1/6H×1/6W×48	
13	12,9	deconv / concat / conv	1/12H×1/12W×64	
14	13,8	deconv / concat / conv	1/24H×1/24W×96	
15	14,7	deconv / concat / conv	1/48H×1/48W×128	
16	15,14	deconv / concat / conv	1/24H×1/24W×96	
17	16,13	deconv / concat / conv	1/12H×1/12W×64	
18	17,12	deconv / concat / conv	1/6H×1/6W×48	
19	18,11	deconv / concat / conv	1/3H×1/3W×32	feature
Guidance branch				
input		concat 1 and up-sampled feature as input	H×W×64	
(1)		3×3 conv	H×W×16	
(2)		5×5 conv, stride 3	1/3H×1/3W×32	
(3)		3×3 conv	1/3H×1/3W×32	
(4)		3×3 conv (no bn & relu)	1/3H×1/3W×640	
(5)		split, reshape, normalize	4×1/3H×1/3W×5×32	sg1
(6)		from (3), 3×3 conv	1/3H×1/3W×32	
(7)		3×3 conv (no bn & relu)	1/3H×1/3W×640	
(8)		split, reshape, normalize	4×1/3H×1/3W×5×32	sg2
(9)-(11)	(6)	from (6), repeat (6)-(8)	4×1/3H×1/3W×5×32	sg3
(12)	(9)	from (9), 3x3 conv, stride 2	1/6H×1/6W×48	
(13)		3x3 conv	1/6H×1/6W×48	
(14)		3×3 conv (no bn & relu)	1/6H×1/6W×960	
(15)		split, reshape, normalize	4×1/3H×1/3W×5×48	sg11
(16)	(13)	from (13), 3x3 conv	1/6H×1/6W×48	
(17)		3×3 conv (no bn & relu)	1/6H×1/6W×960	
(18)		split, reshape, normalize	4×1/6H×1/6W×5×48	sg12
(19)-(21)	(16)	from (16), repeat (16)-(18)	4×1/6H×1/6W×5×48	sg13
(22)-(24)	(19)	from (19), repeat (19)-(21)	4×1/6H×1/6W×5×48	sg14
(25)	(1)	from (1), 3×3 conv	H×W×16	
(26)		3×3 conv (no bn & relu)	H×W×75	lg1
(27)-(28)		repeat (25)-(26)	H×W×75	lg2
Cost aggregation				
input		4D cost volume	1/3H×1/3W×64x(MAX_DISP/3+1)	
[1]	CV	3×3×3, 3D conv	1/3H×1/3W×32x(MAX_DISP/3+1)	
[2]	[1]	SGA layer: weight matrices from (5)	1/3H×1/3W×32x(MAX_DISP/3+1)	
		3×3×3, 3D to 2D conv, upsampling	H×W×(MAX_DISP+1)	
output		softmax, regression	H×W×1	disp0 (for training loss)
[3]	[2]	3×3×3, 3D conv, stride 2	1/6H×1/6W×48x(MAX_DISP/6+1)	
[4]	[3]	SGA layer: weight matrices from (15)	1/6H×1/6W×48x(MAX_DISP/6+1)	
[5]	[4]	3×3×3, 3D conv, stride 2	1/12H×1/12W×64x(MAX_DISP/12+1)	
[6]	[5],[4]	3×3×3, 3D deconv, stride 2	1/6H×1/6W×48x(MAX_DISP/6+1)	
[7]	[6]	SGA layer: weight matrices from (18)	1/6H×1/6W×48x(MAX_DISP/6+1)	
[8]	[7],[2]	3×3×3, 3D deconv, stride 2	1/3H×1/3W×32x(MAX_DISP/3+1)	
[9]	[2]	SGA layer: weight matrices from (8)	1/3H×1/3W×32x(MAX_DISP/3+1)	
		3×3×3, 3D to 2D conv, upsampling	H×W×(MAX_DISP+1)	
output		softmax, regression	H×W×1	disp1 (for training loss)
[10]	[9]	3×3×3, 3D conv, stride 2	1/6H×1/6W×48x(MAX_DISP/6+1)	
[11]	[10]	SGA layer: weight matrices from (21)	1/6H×1/6W×48x(MAX_DISP/6+1)	
[12]	[11]	3×3×3, 3D conv, stride 2	1/12H×1/12W×64x(MAX_DISP/12+1)	
[13]	[12],[11]	3×3×3, 3D deconv, stride 2	1/6H×1/6W×48x(MAX_DISP/6+1)	
[14]	[13]	SGA layer: weight matrices from (24)	1/6H×1/6W×48x(MAX_DISP/6+1)	
[15]	[13],[9]	3×3×3, 3D deconv, stride 2	1/3H×1/3W×32x(MAX_DISP/3+1)	
[16]	[15]	SGA layer: weight matrices from (11)	1/3H×1/3W×32x(MAX_DISP/3+1)	
[17]	[16]	3×3×3, 3D to 2D conv, upsampling	H×W×(MAX_DISP+1)	
[18]	[17]	LGA layer: weight matrices from (26)	H×W×(MAX_DISP+1)	
[19]	[18]	softmax	H×W×(MAX_DISP+1)	
[20]	[19]	LGA layer: weight matrices from (28)	H×W×(MAX_DISP+1)	
output	[20]	normalization, regression	H×W×1	disp2 (estimated disparity)

A.2.3. Data

GANet developers present in [88] the evaluation on three datasets: SceneFlow [58], KITTI2012 and KITTI2015 [34, 59]. The SceneFlow dataset contains stereo frames rendered from various synthetic sequences. The KITTI datasets comprise images from urban and road scenes taken from the viewpoint of a car. In all the cases they are close range images where the camera, real or virtual, is close to the scene. The main characteristics of the images of these datasets are shown in Tables A.2 and A.3.

Table A.2: SceneFlow data characteristics.

	Input stereo pair	Target disparity
Product name	"RGB images (finalpass)"	"Disparity"
File format	PNG	PFM
Channels	3 (RGB)	1
Pixel depth (type)	8 bits (unsigned byte)	32 bits (floating point)
Image size	960x540	960x540

Table A.3: KITTI2012 and KITTI2015 data characteristics.

	Input stereo pair	Target disparity
Channels	3 (RGB)	1
Pixel depth (type)	8 bits (unsigned byte)	32 bits (floating point)
Image size	1240x376	1240x376

A.2.4. Training

Table A.4 presents the training parameters on the SceneFlow, KITTI2012 and KITTI2015 datasets. The authors of the method have disclosed "GANet-deep" models trained on these datasets on their Github page[1].

Table A.4: Training parameters on SceneFlow, KITTI2012 and KITTI2015.

Dataset	**SceneFlow**	**KITTI2012 / KITTI2015**
Training set size (stereo pairs)	35454	194 / 199
Hardware	8 GPUs (*)	8 GPUs (*)
Batch size	16 (**)	16 (**)
Image size (W x H)	576x240 random crops	576x240 random crops
Image preprocessing	Per channel image normalization (***)	Per channel image normalization (***)
Initial weights	Random	From training on SceneFlow
Optimizer	Adam ($\beta_1 = 0{,}9$, $\beta_2 = 0{,}999$)	Adam ($\beta_1 = 0{,}9$, $\beta_2 = 0{,}999$)
Learning rate	0.001	0.001 (first 300 ep.), 0.0001 (remaining ep.)
Epochs	10	640
	(*) P40 - 22GB (**) 8 for the disclosed pretrained models (***) subtract mean divide by std	(*) P40 - 22GB (**) 8 for the disclosed pretrained models (***) substract mean divide by std

A.3. Results

The GANet method has achieved very good results on the KITTI2012 and KITTI2015 [34, 59]. The original model and other more recent variants based on GANet are placed high on the rankings of these benchmarks[2].

In the KITTI benchmarks, specific training on the concrete datasets was performed. But GANet also exhibits great generalization abilities and can perform well on other datasets without a specific training or fine tuning. Some result examples are presented by the authors of GANet on the Cityscapes [16] and the Middlebury [74] datasets on their Github page[3]. Figure A.9 shows the result on one of the images of the Middlebury dataset computed with the demo associated to this chapter (see Section A.4) that uses a model trained on SceneFlow.

The generalization ability of the method was also pointed out in [39] where a model trained on SceneFlow (comprised of close range images) was used on satellite images with encouraging results. Despite the current popularity of deep learning stereo matching methods, they are still not the preferred matching option in satellite stereo pipelines [19, 60, 69, 73]. Satellite images have specific characteristics that hinder the adaptation of well established methods used on close range images: a) the extremely small ratio between the depth range and the distance from the camera to the scene implies working with a camera model that deviates from the standard pinhole and deals with structures that occupy few pixels in the images; (b) the images for a certain location can only be acquired through several

[1] `https://github.com/feihuzhang/GANet` [Accessed on December 2022].

[2] `http://www.cvlibs.net/datasets/kitti/eval_stereo.php` [Accessed on December 2022].

[3] `https://github.com/feihuzhang/GANet` [Accessed on December 2022].

sweeps which may be days, weeks or even months apart, introducing variability in illumination, seasonal changes and man-made changes, among others. The variability poses important challenges for the matching of correspondent regions across the images. Despite the differences between the train and test sets, [39] shows that reconstruction results with GANet, used as the matching step in the S2P [19] satellite pipeline, were comparable to the results with the classic matching counterpart [28] currently in use in the pipeline. It is interesting to note that part of the internal structure of GANet mimics SGM [42] which has been extensively used as the main aggregation strategy in classic matching methods of satellite stereo pipelines.

A.4. Demo

The IPOL demo related to this chapter can be accessed at: https://www.ipol.im/pub/pre/441/.

This is a demo of GANet adapted but not optimized to run on CPU. Execution is restricted to small images. The demo uses the "GANet-deep" model trained on SceneFlow mentioned in Section A.2.4. Figure A.5 shows the demo interface in the IPOL system.

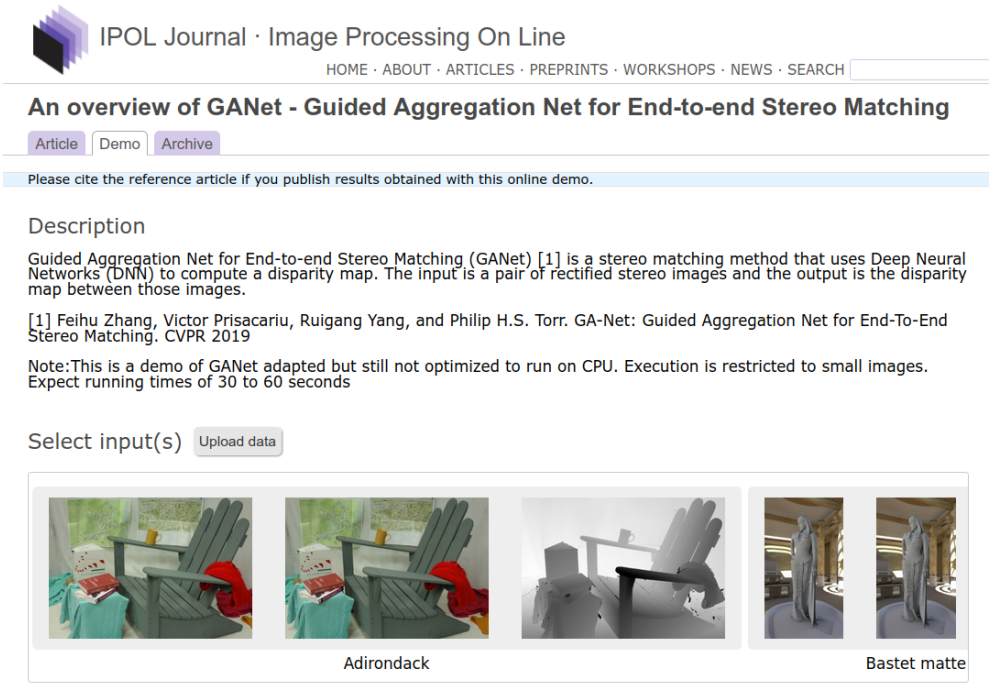

Figure A.5: GANet demo in IPOL.

To run the demo the users must first select a pair of images from the gallery, or upload their own images. The gallery (see Figure A.6) has also the ground truth

for the disparity, which can be compared with the result of an execution. In the case of uploaded images the ground truth is optional.

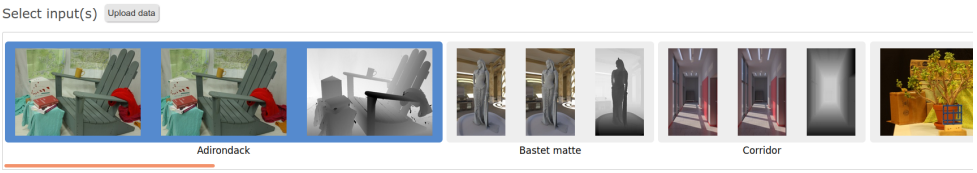

Figure A.6: Gallery of available image pairs. The demo also allows to upload images.

Once the input images are selected, they can be inspected in the Inputs section as shown in Figure A.7.

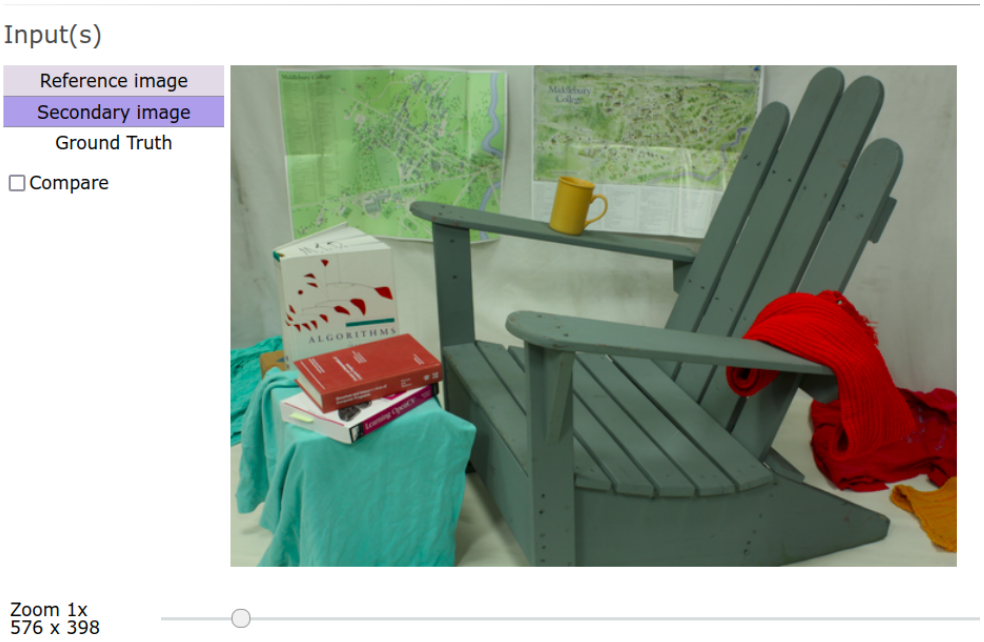

Figure A.7: Input images can be inspected by hovering over the buttons. This allows to easily view the disparity between the images of the pair and the ground truth if available.

Next, the parameters must be selected and the Run button must be pressed. Figure A.8 presents the parameters pane.

When the execution is finished, the computed disparity map can be inspected in the Results section by alternating the images (hovering over the buttons) or by a side-to-side comparison as shown in Figure A.9.

Figure A.8: Parameter selection. The demo is an adaptation of GANet to run on a CPU. Small images are necessary for moderate running times. The size of the crops are controlled with the first two sliders. The max_disp parameter controls the number of disparity steps considered in the reconstruction. Smaller values of this parameter result in shorter running time but coarser results.

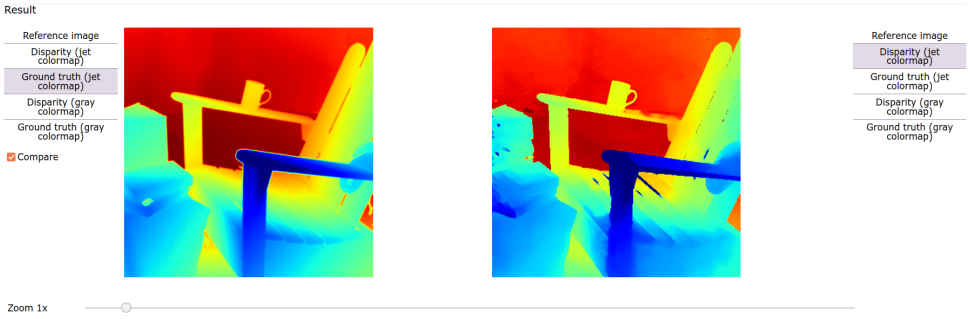

Figure A.9: Results section and a side-to-side comparison of the computed disparity and the ground truth.

References

[1] Hervé Abdi. The kendall rank correlation coefficient. *Encyclopedia of measurement and statistics*, 2:508–510, 2007.

[2] Roland Akiki, Roger Marí, Carlo De Franchis, Jean-Michel Morel, and Gabriele Facciolo. Robust rational polynomial camera modelling for sar and pushbroom imaging. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pages 7908–7911, 2021.

[3] Marc K Albert and Donald D Hoffman. Genericity in spatial vision. In D. Luce, K. Romney, D. Hoffman, and D'Zmura M., editors, *Geometric Representations of Perceptual Phenomena: Arts. in Honor of Tarow Indow's 70th Birthday*, pages 95–112. Erlbaum, 1995.

[4] Amir Atapour-Abarghouei and Toby P. Breckon. A comparative review of plausible hole filling strategies in the context of scene depth image completion. *Computers & Graphics*, 72:39–58, May 2018.

[5] H Harlyn Baker and Thomas O Binford. Depth from edge and intensity based stereo. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, page 631–636, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[6] Serge Beucher. Use of watersheds in contour detection. In *Proc. Int. Workshop on Image Processing, Sept. 1979*, pages 17–21, 1979.

[7] Michael Bleyer and Margrit Gelautz. Simple but effective tree structures for dynamic programming-based stereo matching. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 2, pages 415–422, 2008.

[8] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo-stereo matching with slanted support windows. In *British Machine Vision Conference*, volume 11, pages 1–11, 2011.

[9] Marc Bosch, Kevin Foster, Gordon Christie, Sean Wang, Gregory D Hager, and Myron Brown. Semantic stereo for incidental satellite images. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1524–1532. IEEE, 2019.

[10] Marc Bosch, Zachary Kurtz, Shea Hagstrom, and Myron Brown. A multiple view stereo benchmark for satellite imagery. In *2016 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1–9. IEEE, 2016.

[11] Sebastian Bullinger, Christoph Bodensteiner, and Michael Arens. 3d surface reconstruction from multi-date satellite images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B2-2021:313–320, 2021.

[12] Steven D. Cochran and Gérard Medioni. 3-d surface description from binocular stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):981–994, oct 1992.

[13] Robert T Collins. A space-sweep approach to true multi-image matching. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 358–363, June 1996. ISSN: 1063-6919.

[14] Miguel Colom and Antoni Buades. Analysis and extension of the Ponomarenko et al. method, estimating a noise curve from a single image. *Image Processing On Line*, 3:173–197, 2013.

[15] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[16] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[17] Pablo d'Angelo, Cristian Rossi, Christian Minet, Michael Eineder, Michael Flory, and Irmgard Niemeyer. High resolution 3d earth observation data analysis for safeguards activities. In *Symposium on International Safeguards*, pages 1–8, 2014.

[18] Carlo de Franchis, Enric Meinhardt-Llopis, Daniel Greslou, and Gabriele Facciolo. Attitude Refinement for Orbiting Pushbroom Cameras: a Simple Polynomial Fitting Method. *Image Processing On Line*, 5:328–361, 12 2015.

[19] Carlo de Franchis, Enric Meinhardt-Llopis, Julien Michel, Jean-Michel Morel, and Gabriele Facciolo. An automatic and modular stereo pipeline for pushbroom images. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3:49–56, 2014.

[20] Carlo de Franchis, Enric Meinhardt-Llopis, Julien Michel, Jean-Michel Morel, and Gabriele Facciolo. On stereo-rectification of pushbroom images. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5447–5451, Paris, France, October 2014. IEEE.

[21] Dawa Derksen and Dario Izzo. Shadow neural radiance fields for multi-view satellite photogrammetry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1152–1161, 2021.

[22] Agnès Desolneux, Lionel Moisan, and Jean-Michel Morel. Meaningful alignments. *International Journal of Computer Vision*, 40(1):7–23, 2000. `https://doi.org/10.1023/A:1026593302236`.

[23] Agnès Desolneux, Lionel Moisan, and Jean-Michel Morel. *From Gestalt Theory to Image Analysis: A Probabilistic Approach*, volume 34. Springer-Verlag New York, 2008. `https://doi.org/10.1007/978-0-387-74378-3`.

[24] Gene Dial and Jacek Grodecki. Rpc replacement camera models. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34, 2005.

[25] Amnon Drory, Carsten Haubold, Shai Avidan, and Fred A. Hamprecht. Semi-global matching: A principled derivation in terms of message passing. In Xiaoyi Jiang, Joachim Hornegger, and Reinhard Koch, editors, *Pattern Recognition*, pages 43–53. Springer International Publishing, Cham, 2014.

[26] Sébastien Drouyer, Serge Beucher, Michel Bilodeau, Maxime Moreaud, and Loïc Sorbier. Sparse Stereo Disparity Map Densification Using Hierarchical Image Segmentation. In Jesús Angulo, Santiago Velasco-Forero, and Fernand Meyer, editors, *Mathematical Morphology and Its Applications to Signal and Image Processing*, Lecture Notes in Computer Science, pages 172–184. Springer International Publishing, 2017.

[27] Gabriele Facciolo. Stereovision for satellite images. Lecture notes of the course: Remote sensing data: from sensor to large-scale geospatial data exploitation, February 2018.

[28] Gabriele Facciolo, Carlo de Franchis, and Enric Meinhardt. MGM: A significantly more global matching for stereovision. In *Procedings of the British Machine Vision Conference 2015*, pages 90.1–90.12. British Machine Vision Association, 2015.

[29] Gabriele Facciolo, Carlo De Franchis, and Enric Meinhardt-Llopis. Automatic 3D Reconstruction from Multi-date Satellite Images. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1542–1551, Honolulu, HI, July 2017. IEEE.

[30] Pascal Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine vision and applications*, 6(1):35–49, 1993.

[31] Yasutaka Furukawa and Carlos Hernández. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015.

[32] David Gallup, Marc Pollefeys, and Jan-Michael Frahm. 3d reconstruction using an n-layer heightmap. In *Joint Pattern Recognition Symposium*, pages 1–10. Springer, 2010.

[33] Jian Gao, Jin Liu, and Shunping Ji. Rational polynomial camera model warping for deep learning based satellite multi-view stereo matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6148–6157, 2021.

[34] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[35] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M Seitz. Multi-view stereo for community photo collections. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[36] Rafael Gonzalez and Richard Woods. *Digital Image Processing*. Pearson/-Prentice Hall, NY, 2018.

[37] Jacek Grodecki. Ikonos stereo feature extraction-rpc approach. In *ASPRS annual conference St. Louis*, 2001.

[38] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2495–2504, 2020.

[39] Alvaro Gómez, Gregory Randall, Gabriele Facciolo, and Rafael Grompone von Gioi. An experimental comparison of multi-view stereo approaches on satellite images. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 707–716, 2022.

[40] Rostam Affendi Hamzah and Haidi Ibrahim. Literature survey on stereo vision disparity map algorithms. *Journal of Sensors*, 2016.

[41] Heiko Hirschmüller. Stereo Vision in Structured Environments by Consistent Semi-Global Matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2386–2393, 2006.

[42] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

[43] Hiroshi Ishikawa. Exact optimization for Markov Random Fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1333–1336, 2003.

[44] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanaes. Large Scale Multi-view Stereopsis Evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413, Columbus, OH, USA, June 2014. IEEE.

[45] Maurice George Kendall. Rank correlation methods. 1948.

[46] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.

[47] Thomas Krauß, Pablo d'Angelo, Mathias Schneider, and Veronika Gstaiger. The fully automatic optical processing system catena at DLR. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 177–183, 2013.

[48] Hamid Laga. A Survey on Deep Learning Architectures for Image-based Depth Reconstruction. *arXiv:1906.06113 [cs, eess]*, June 2019. arXiv: 1906.06113.

[49] Hamid Laga, Laurent Valentin Jospin, Farid Boussaid, and Mohammed Bennamoun. A survey on deep learning techniques for stereo-based depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[50] Matthew J. Leotta, Chengjiang Long, Bastien Jacquet, Matthieu Zins, Dan Lipsa, Jie Shan, Bo Xu, Zhixin Li, Xu Zhang, Shih-Fu Chang, Matthew Purri, Jia Xue, and Kristin Dana. Urban semantic 3d reconstruction from multiview satellite imagery. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[51] José Lezama, Jean-Michel Morel, Gregory Randall, and Rafael Grompone Von Gioi. A contrario 2D point alignment detection. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):499–512, 2015. `http://dx.doi.org/10.1109/TPAMI.2014.2345389`.

[52] José Lezama, Gregory Randall, Jean-Michel Morel, and Rafael Grompone von Gioi. An Unsupervised Point Alignment Detection Algorithm. *Image Processing On Line*, 5:296–310, 2015. `https://doi.org/10.5201/ipol.2015.126`.

[53] Jin Liu and Shunping Ji. A novel recurrent encoder-decoder structure for large-scale multi-view stereo reconstruction from an open aerial dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6050–6059, 2020.

[54] Charles Loop and Zhengyou Zhang. Computing rectifying homographies for stereo vision. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 1, pages 125–131 Vol. 1, 1999.

[55] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[56] Roger Marí, Carlo de Franchis, Enric Meinhardt-Llopis, and Gabriele Facciolo. To Bundle Adjust or Not: A Comparison of Relative Geolocation Correction Strategies for Satellite Multi-View Stereo. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 2188–2196, Seoul, Korea (South), October 2019. IEEE.

[57] Roger Marí, Gabriele Facciolo, and Thibaud Ehret. Sat-nerf: Learning multi-view satellite photogrammetry with transient objects and shadow modeling using rpc cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1311–1321, 2022.

[58] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. arXiv:1512.02134.

[59] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.

[60] Zachary M Moratto, Michael J Broxton, Ross A Beyer, Mike Lundy, and Kyle Husmann. Ames stereo pipeline, NASA's open source automated stereogrammetry software. *LPI*, (1533):2364, 2010.

[61] Jean-Michel Morel and Guoshen Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM journal on imaging sciences*, 2(2):438–469, 2009.

[62] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application (VISSAPP)*, pages 331–340. INSTICC Press, 2009.

[63] Chukwuma J Okolie and Julian L Smit. A systematic review and meta-analysis of digital elevation model (dem) fusion: pre-processing, methods and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 188:1–29, 2022.

[64] Ozge C Ozcanli, Yi Dong, Joseph L Mundy, Helen Webb, Riad Hammoud, and Victor Tom. A comparison of stereo and multiview 3-D reconstruction using cross-sensor satellite imagery. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 17–25, June 2015.

[65] Sonali Patil, Bharath Comandur, Tanmay Prakash, and Avinash C Kak. A new stereo benchmarking dataset for satellite images. *arXiv preprint arXiv:1907.04404*, 2019.

[66] Daniela Poli and Thierry Toutin. Review of developments in geometric modelling for high resolution satellite pushbroom sensors. *The Photogrammetric Record*, 27(137):58–73, 2012.

[67] Nikolay N Ponomarenko, Vladimir V Lukin, MS Zriakhov, Arto Kaarna, and Jaakko Astola. An automatic approach to lossy compression of aviris images. In *2007 IEEE International Geoscience and Remote Sensing Symposium*, pages 472–475. IEEE, 2007.

[68] Adrian K Preiss. *A Theoretical and Computational Investigation into Aspects of Human Visual Perception: Proximity and Transformations in Pattern Detection and Discrimination*. PhD thesis, University of Adelaide, 2006. `http://hdl.handle.net/2440/37820`.

[69] Rongjun Qin. Rpc stereo processor (rsp)–a software package for digital surface model and orthophoto generation from satellite stereo imagery. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3:77, 2016.

[70] Rongjun Qin. Automated 3d recovery from very high resolution multi-view satellite images. In *Proceedings of the ASPRS Conference (IGTF) 2017, Baltimore, MD, USA, 12–17 March 2017*, pages 12–16, 2017.

[71] Rongjun Qin. A critical analysis of satellite stereo pairs for digital surface model generation and a matching quality prediction model. *ISPRS Journal of Photogrammetry and Remote Sensing*, 154:139–150, 2019.

[72] Sébastien Roy and I J Cox. A Maximum-Flow Formulation of the N-Camera Stereo Correspondence Problem. In *Proceedings of the Sixth International Conference on Computer Vision*, volume 5, page 492, 1998.

[73] Ewelina Rupnik, Mehdi Daakir, and Marc Pierrot Deseilligny. Micmac–a free, open-source solution for photogrammetry. *Open Geospatial Data, Software and Standards*, 2(1):1–9, 2017.

[74] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *Pattern Recognition*, pages 31–42. Springer, 2014.

[75] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[76] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, Las Vegas, NV, USA, June 2016. IEEE.

[77] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pages 501–518. Springer, 2016.

[78] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[79] Pierre Soille. *Morphological image analysis: principles and applications*. Springer Science & Business Media, 2013.

[80] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, pages 839–846. IEEE, 1998.

[81] Charles Toth and Grzegorz Jóźków. Remote sensing platforms and sensors: A survey. *ISPRS Journal of Photogrammetry and Remote Sensing*, 115:22–36, 2016. Theme issue 'State-of-the-art in photogrammetry, remote sensing and spatial information science'.

[82] Srimant P Tripathy, Alexander J Mussap, and Horace B Barlow. Detecting collinear dots in noise. *Vision Research*, 39:4161–4171, 1999. `http://dx.doi.org/10.1016/S0042-6989(99)00125-X`.

[83] William R Uttal. The effect of deviations from linearity on the detection of dotted line patterns. *Vision Research*, 13(11):2155–2163, 1973.

[84] Olga Veksler. Stereo Correspondence by Dynamic Programming on a Tree. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 384–390, 2005.

[85] Johan Wagemans. Perceptual use of nonaccidental properties. *Canadian Journal of Psychology*, 46(2):236–279, 1992. `http://dx.doi.org/10.1037/h0084323`.

[86] Andrew P Witkin and Jay M Tenenbaum. On the role of structure in vision. In J. Beck, B. Hope, and A. Rosenfeld, editors, *Human and Machine Vision*, pages 481–543. Academic Press, 1983. `http://dx.doi.org/10.1016/B978-0-12-084320-6.50022-0`.

[87] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. MVSNet: Depth Inference for Unstructured Multi-view Stereo. *arXiv:1804.02505 [cs]*, July 2018. arXiv: 1804.02505.

[88] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip H.S. Torr. GA-Net: Guided Aggregation Net for End-To-End Stereo Matching. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 185–194, Long Beach, CA, USA, June 2019. IEEE.

[89] Kai Zhang, Noah Snavely, and Jin Sun. Leveraging vision reconstruction pipelines for satellite imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019.

[90] Enliang Zheng, Enrique Dunn, Vladimir Jojic, and Jan-Michael Frahm. PatchMatch Based Joint View Selection and Depthmap Estimation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1517, Columbus, OH, USA, June 2014. IEEE.

Esta es la última página.
Compilado el martes 18 abril, 2023.
http://iie.fing.edu.uy/