Urban 3D Reconstruction in Remote Sensing via Deep Learning and Dataset Enhancement Strategies

DISSERTATION

Zur Erlangung
des Doktorgrades der Naturwissenschaften (Dr. rer. nat.)
des Fachbereichs Mathematik/Informatik
der Universität Osnabrück

Vorgelegt von

Mario Fuentes Reyes

Prüfer der Dissertation:

Prof. Dr. Peter Reinartz, Universität Osnabrück
Prof. Dr. Friedrich Fraundorfer, Technische Universität Graz, Österreich

Tag der mündlichen Prüfung: 14.03.2025

Abstract

Modelling the profile of a city has been widely studied by the research community, particularly in remote sensing. By using sensors located on airborne and satellite platforms, it is possible to retrieve data such as optical/infrared images, radar and laser measurements, etc. Many of these sensors can be used to compute the 3D profile of the scene. Radar and LiDAR are able to measure the distance with high accuracy, but the reconstruction might be sparse, include outliers and uses expensive technology. Images on the contrary are relatively cheaper and capture geometric details, useful for a dense reconstruction. Nonetheless, the reconstruction depends on the matching capabilities of the applied algorithm, as the depth has to be computed from the displacement of corresponding pixels in the images.

Before the deep learning solutions, algorithms such as Semi-Global Matching or those based on Structure from Motion used to lead the reconstruction benchmarks. These conventional algorithms can be implemented on any set of images without any prior knowledge of the scene and the refinement process, which benefit from geometric principles to detect inconsistencies and occlusions, generate an accurate digital surface models with few remaining outliers. However, conventional approaches fail in complicated areas such as those with poor texture, repetitive patters, reflective surfaces, that are common in remote sensing imagery.

In contrast, deep learning approaches deal better with complicated areas and by using contextual information, they are able to reconstruct a smooth 3D profile with few outliers and high accuracy. Yet, learning based algorithms might fail if the differences between the training and testing sets are large. In addition, neural networks require a large amount of quality data for a robust training, which is not easy to collect for remote sensing platforms. What is more, ground truth might still be obtained with laser but for smaller regions, leading to domain shifts.

Hence, the first step to set a reliable framework to evaluate reconstruction algorithms is to provide high quality data. As this is expensive in a real scenario, this study proposes the use of a pipeline to generate large amounts of synthetic data to train stereo matching and multi-view stereo (MVS) networks. Since the data is rendered from software, accurate ground truth is available. Moreover, as the software allows editions of the virtual scene, the urban growth can be simulated, which helps to create data for additional tasks like change detection.

A reliable dataset allows to set up experiments to evaluate the quality of the reconstruction algorithms. This dissertation considers two main research directions to design these experiments. On the one hand, it is important to explore the advantages of both the conventional and the learning based solutions, which are evaluated for the stereo matching case. On the other hand, a comparison between the stereo and MVS algorithms is conducted. Intuitively, using complementary information as MVS does might produce a more robust result, but stereo methods have been more studied and have a simplified matching case. Therefore, conventional and learnable, stereo and MVS algorithms are analysed with reliable datasets to assess how these contribute to the 3D reconstruction task. Furthermore, an alternative case to fuse height values into a final digital surface model is explored, where the confidence for the values predicted by the neural networks is estimated and used to guide the fusion.

Valuable insights into the urban 3D reconstruction were obtained from the carried out experiments. The generation of datasets from real and synthetic scenarios facilitated the analysis of the capabilities of the tested algorithms. Despite the well-known problem of the domain gap, the networks trained on the generated datasets produced good reconstruction results in complex regions. Buildings and man-made structures benefit from the synthetic models, but for vegetation and natural elements the algorithms exhibit a lower performance because such elements are simplified in the 3D modelling.

Among the methods tested, stereo matching approaches computed reconstructions that were less prone to outliers, while the MVS was more robust for edge discontinuities. However, learning algorithms estimate a value for each pixel in the input images, but the reliability of this estimation should still be assessed. By pre-selecting the predicted values based on a confidence estimation, the accuracy of the fusion was improved for the stereo matching case. Yet, this fusion strategy needs to be further explored to generalize to MVS methods as well.

Zusammenfassung

Die Modellierung des Profils einer Stadt ist in der Forschung, insbesondere im Bereich der Fernerkundung, weithin untersucht worden. Durch den Einsatz von Sensoren auf Luft- und Satellitenplattformen ist es möglich, Daten wie beispielsweise optische/Infrarot-Bilder, Radar- und Lasermessungen zu erhalten. Viele dieser Sensoren können verwendet werden, um das 3D-Profil der Szene zu berechnen. Radar und LiDAR sind in der Lage, die Entfernung mit hoher Genauigkeit zu messen, die Rekonstruktion kann jedoch spärlich sein, Ausreißer enthalten und teure Technologie verwenden. Bilder hingegen sind relativ kostengünstig und erfassen geometrische Details, die für eine dichte Rekonstruktion nützlich sind. Dennoch hängt die Rekonstruktion von den Matching-fähigkeiten des verwendeten Algorithmus ab, da die Tiefe aus der Verschiebung der entsprechenden Pixel in den Bildern berechnet werden muss.

Vor den Deep Learning-Lösungen führten Algorithmen wie Semi-Global Matching oder solche, die auf Structure from Motion basieren, die Rekonstruktionsbenchmarks an. Diese konventionellen Algorithmen können ohne Vorkenntnisse der Szene auf jeden Satz von Bildern angewendet werden und der Verfeinerungsprozess, der von geometrischen Prinzipien zur Erkennung von Inkonsistenzen und Verdeckungen profitiert, erzeugt ein genaues digitales Oberflächenmodell mit wenigen verbleibenden Ausreißern. Konventionelle Ansätze versagen jedoch in komplizierten Bereichen, wie z.B. solchen mit schlechter Textur, sich wiederholenden Mustern und reflektierenden Oberflächen, die in Fernerkundungsbildern häufig vorkommen.

Im Gegensatz dazu kommen Deep-Learning-Ansätze besser mit komplizierten Gebieten zurecht. Durch die Verwendung von Kontextinformationen sind sie in der Lage, ein glattes 3D-Profil mit wenigen Ausreißern und hoher Genauigkeit zu rekonstruieren. Allerdings können lernbasierte Algorithmen bei großen Unterschieden zwischen den Trainings- und den Testsätzen versagen. Außerdem benötigen neuronale Netze für ein robustes Training eine große Menge an qualitativ hochwertigen Daten, die für Fernerkundungsplattformen nicht einfach zu sammeln sind. Außerdem können die Ground-Truth Daten zwar mit dem Laser gewonnen werden, aber nur für kleinere Regionen, was zu Bereichsverschiebungen führt.

Der erste Schritt, um einen zuverlässigen Rahmen für die Bewertung von Rekonstruktionsalgorithmen zu schaffen, besteht daher in der Bereitstellung hochwertiger Daten. Da dies in einem realen Szenario teuer ist, schlägt diese Studie die Verwendung einer Pipeline zur Erzeugung großer Mengen synthetischer Daten vor, um Stereo-Matching- und Multi-View-Stereo-Netzwerke (MVS) zu trainieren. Da die Daten von einer Software gerendert werden, ist eine genaue Ground-Truth verfügbar. Da die Software außerdem die Bearbeitung der virtuellen Szene ermöglicht, kann das Wachstum der Stadt simuliert werden, wodurch Daten für zusätzliche Aufgaben, wie die Erkennung von Veränderungen, erzeugt werden können.

Ein zuverlässiger Datensatz ermöglicht die Durchführung von Experimenten, um die Qualität der Rekonstruktionsalgorithmen zu bewerten. Diese Arbeit berücksichtigt zwei Hauptforschungsrichtungen, um diese Experimente zu konzipieren. Einerseits ist es wichtig, die Vorteile sowohl der konventionellen als auch der lernbasierten Lösungen zu erforschen, die für

den Fall des Stereo-Matchings bewertet werden. Andererseits wird ein Vergleich zwischen dem Stereo- und dem MVS-Algorithmus durchgeführt. Intuitiv könnte die Verwendung komplementärer Informationen wie bei MVS zu einem robusteren Ergebnis führen, Stereomethoden sind jedoch besser erforscht und haben ein vereinfachtes Matching. Daher werden konventionelle und erlernbare Stereo- und MVS-Algorithmen mit zuverlässigen Datensätzen analysiert, um zu beurteilen, wie diese zur 3D-Rekonstruktion beitragen. Darüber hinaus wird ein alternativer Fall der Fusion von Höhenwerten zu einem endgültigen digitalen Oberflächenmodell untersucht, bei dem die Konfidenz in die von den neuronalen Netzen vorhergesagten Werte geschätzt und zur Steuerung der Fusion verwendet wird.

Die durchgeführten Experimente lieferten wertvolle Erkenntnisse über die 3D-Rekonstruktion von Städten. Die Erstellung von Datensätzen aus realen und synthetischen Szenarien erleichterte die Fähigkeiten der getesteten Algorithmen zu analysieren. Trotz des bekannten Problems der Domänenlücke lieferten die auf den generierten Datensätzen trainierten Netzwerke gute Rekonstruktionsergebnisse in komplexen Regionen. Gebäude und künstliche Strukturen profitieren von den synthetischen Modellen, aber für Vegetation und natürliche Elemente zeigen die Algorithmen eine geringere Leistung, da solche Elemente bei der 3D-Modellierung vereinfacht werden.

Unter den getesteten Methoden berechneten die Stereo-Matching-Ansätze Rekonstruktionen, die weniger anfällig für Ausreißer waren, während das MVS robuster gegenüber Kantenunterbrechungen war. Allerdings schätzen Lernalgorithmen einen Wert für jedes Pixel in den Eingabebildern, aber die Zuverlässigkeit dieser Schätzung sollte dennoch bewertet werden. Durch die Vorauswahl der vorhergesagten Werte auf der Grundlage einer Konfidenzschätzung wurde die Genauigkeit der Fusion für den Fall des Stereo-Matchings verbessert. Diese Fusionsstrategie muss jedoch noch weiter erforscht werden, um sie auch für MVS-Methoden zu verallgemeinern.

Acknowledgments

Firstly, I would like to thank the German Academic Exchange Service (DAAD) for providing me with a scholarship to pursue my PhD studies at the German Aerospace Center (DLR). I would also like to thank all the institutes and organizations within the DLR that facilitated the tools I needed to conduct my research. I would also like to recognize the friendly communication I received from the staff at the University of Osnabrück.

I would also like to thank Prof. Dr. Peter Reinartz for giving me the opportunity to join the Photogrammetry and Image Analysis Department in recent years. His support and guidance have helped me gain a deeper insight into the remote sensing field and improve the quality of my research outputs. Thank you for your help in making my research topic possible.

Thanks are also given to Prof. Dr. Friedrich Fraundorfer, whose constant supervision offered a more complete understanding of the 3D reconstruction area, and whose constructive feedback led to new experiments that enriched the publications related to this research. Thank you for our regular meetings and the very pleasant conversations we have had.

My gratitude is extended to Dr. Pablo D'Angelo, my technical supervisor, who has helped me with many aspects of my PhD research, from coding ideas to releasing datasets. His supportive, friendly and professional attitude has resulted in more robust and scientifically sound outputs. This thesis would not have been possible without his help and enthusiasm.

To all my colleagues at DLR: thank you for the collaborative atmosphere, the shared ideas and the memories made at conferences and during regular coffee breaks. I found not only good research at the office, but also a great team of people.

I would like to thank the friends I have made since living in Munich for enriching my time in this city and for providing reliable support during the challenging times many of us experienced when settling in a new country.

To my girlfriend, Julia, thank you for the company and love all these years. Thanks to your patience and support it was always possible to do the research work and maintain a healthy and restful life. Thanks also to her family, who have always been there for me and made it possible for me to find a second home in Germany.

To my parents, Mario and Obdulia, who have undoubtedly been a pillar in my education, values and outlook on life. Thank you for your support to study and seek a better future. For always being there at a distance and for the love you have always shown me. To my siblings, Janeth and Oscar thanks also for the good times living away from home, for your support in difficult times and for all the shared laughs. We have grown together.

To my friends in Mexico: Marcial, Rodrigo, Pame and Iván, who have always given their encouragement and kept in touch despite the distance.

Finally, I would like to dedicate this work to my grandmother, Tila, who blessed me when I left home to study. Even you are no longer here to see it, this achievement is for you.

Thanks!, Danke!, ¡Gracias!

Contents

1

Introduction

Contents

The generation of digital surface models (DSMs) is commonly a starting point for other remote sensing tasks such as building detection, semantic segmentation and terrain models creation. However, there are many approaches in the research community for DSM generation and it is difficult to assess which one performs best, particularly for urban reconstruction.

Height can also be estimated by direct measurement using instruments such as LiDAR, which is highly accurate. However, in practice it is very expensive to scan regions with LiDAR and if the measurements are not dense enough, the computed DSM will lack sharp boundaries. Optical imagery on the other hand is relatively cheaper and contains all relevant geometric information to reconstruct sharp edges.

Reconstructing the 3D profile of a scene from images is a widely studied topic in the computer vision community. Based on the matching of features visible in two or more images and the camera parameters, it is possible to estimate how far away are the objects from the camera. Two main solutions have been addressed to tackle this task: stereo matching and multi-view stereo.

Stereo matching takes two stereo rectified images as input and matches the common features, which lay on the same epipolar line as the images are already rectified. The algorithms estimate how many pixels the features have been shifted between the two images, where such a shift is called disparity. If the parameters of the stereo array are known (namely baseline and focal length), the distance to the object can be computed from the disparity.

In the multi-view stereo case (MVS), the input images have different locations and rotations around the scene. Hence, the matching would be a computational demanding task in the image domain. By using the homography matrix, the pixels are translated into the 3D domain and the matching is applied for points in this space. As a result, we obtain directly the depth each pixel represents in the image.

Furthermore, there are two main strategies to develop stereo and MVS algorithms: traditional photogrammetry and deep learning. The former relies in the equations that describe the camera systems and the projection of 3D scenes into the camera plane without previous parameter training or prior knowledge of the scene. This has been the main strategy for many years and is widely studied. Deep learning takes a large set of images as input data and learns through multiple convolutional layers and iterations the relation between input and output. Although subject to the domain similarities, neural networks lead most of the benchmarks.

Nonetheless, considering the fact that images in the remote sensing field are not set in a controlled environment, the matching in any of the cases is a challenging task. Effects such as seasonal changes, illumination conditions, construction works and textureless areas lead to wrong disparity/depth predictions and affect all algorithms.

The particular point of construction works, while being a challenge for matching, it is also an indicator for changes in the urban development. For cadastral databases and urban planning, it is relevant to keep track of changes in buildings, either because of new constructions, demolitions or building renovation works. As 3D data is valuable for this task, we consider it within the scope of this thesis.

1.1 Scope of the Research

In this thesis there are different aspects to be analysed in the task of 3D urban reconstruction. Firstly, the quality of the existing data to assess stereo and MVS algorithms is discussed. Apart from the input images, reference data is needed to evaluate the reconstructed DSMs. LiDAR generated maps are a reliable source, but due to the expensive acquisition costs, only few areas are surveyed with this sensor and where these maps are available, these are not regularly updated. In other cases, the reference data is computed by merging the result from many optical-derived DSMs, so the evaluation of new algorithms is subject to the accuracy of those used to generate the reference data. Hence, developing synthetic data is a viable alternative to compensate for the lack of accurate data and allows to simulate acquisitions that give more insights of the reconstruction algorithms. Besides, it reduces significantly the costs as no real flight campaigns or satellite missions are required. The pipeline to create synthetic data, the generated images and ground truths, and the resemblance to reality of the samples is the starting point for this thesis.

In addition, the 3D software provides a controlled environment which helps in the preparation of data for the analysis of the detection of urban changes. It is difficult in reality to prepare a dataset for change detection, as acquisitions with similar conditions in different times are needed to compare the variations in the scene. In a virtual environment it is easier to modify the scene and render with different conditions. Thus, a similar pipeline as for stereo data can be applied to create data for change detection algorithms.

Once reliable data is available, performance analysis can be carried out. Traditional photogrammetry and deep learning algorithms are compared for the stereo matching task. This highlights the advantages and drawbacks of each case. The performance to reconstruct specific objects of the scene is also studied.

Similarly, stereo and MVS algorithms are evaluated considering deep learning approaches. This is an important evaluation, as the approaches are usually applied separately to study cases and the performance on the same 3D reconstructed area has not been the subject of in-depth research.

Last but not least, confidence estimation is also covered in this dissertation. While the median fusion is a robust solution to merge multiple DSMs, a more sophisticated method could lead to improvements in accuracy. The reconstruction algorithms generate disparity and depth maps, where some pixels are derived from more confident estimations than others. A fusion based on a confidence value assigned to each predicted pixel could be a more accurate solution.

Summarizing, the study objectives of this dissertation are:

- Develop a pipeline to generate synthetic data to be used for stereo and MVS algorithms. It is important that the generated samples resemble features from real images in the remote sensing field.
- Develop a different synthetic dataset oriented to the urban change detection, where the city growth process is simulated.

- Study the advantages and disadvantages of traditional and deep learning methods for the stereo matching.
- Study the performance of stereo and MVS deep learning algorithms to reconstruct the same urban areas.
- Analyse the benefits and limitations of using synthetic data for deep learning algorithms.
- Investigate the role of confidence predictions for the fusion of DSMs.

1.2 Dissertation Structure

This thesis is presented as a cumulative dissertation, which therefore is subject to the content of peer reviewed publications. The chapters are organized as follows:

Chapter 2 provides an introduction to the nature of the images used in remote sensing, the principles for stereo and MVS matching in both traditional photogrammetric and deep learning approaches, and the studies related to confidence estimation.

Chapter 3 focuses on the pipeline to create synthetic data and its application for stereo matching, where traditional and learning methods are compared. This is linked to the paper:

- Fuentes Reyes, M, d'Angelo, P., & Fraundorfer, F. (2022). SyntCities: A large synthetic remote sensing dataset for disparity estimation. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 15, 10087-10098.

Chapter 4 is related to the generation of synthetic data for city change detection. The content is described in the paper:

- Fuentes Reyes, M., Xie, Y., Yuan, X., d'Angelo, P., Kurz, F., Cerra, D., & Tian, J. (2023). A 2D/3D multimodal data simulation approach with applications on urban semantic segmentation, building extraction and change detection. ISPRS Journal of Photogrammetry and Remote Sensing, 205, 74-97.

Chapter 5 analyses the differences between stereo and MVS learnable algorithms. This chapter also deals with the confidence guided fusion for DSMs. This topic is in a paper still under submission process.

Chapter 6 is a complementary study to chapter 5, where a similar set of experiments is designed if only 2 views are available with synthetic data. This study was included in the conference paper:

- Fuentes Reyes, M., d'Angelo, P., & Fraundorfer, F. (2023). An evaluation of stereo and multiview algorithms for 3D reconstruction with synthetic data. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 48, 1021-1028.

Chapter 7 includes general conclusions based on the findings from this thesis and looks into pending research topics for future work.

2

Theoretical Background

Contents

This chapter gives a brief introduction to the topics that will serve as the basis for understanding the following chapters. Firstly, the description of the camera geometry, its parameters and the configuration for stereo systems are addressed. Then, the principles applied for 3D reconstruction algorithms are explained in both stereo matching and MVS with traditional and learnable approaches. Later, some existing datasets for these tasks are reviewed. Finally, basic concepts related to confidence estimation are described.

2.1 Camera and image principles

Images in remote sensing usually come from three different sources: satellite missions, manned flight campaigns and unmanned aerial vehicles (UAVs). The last two sources use cameras that can be simplified by the pinhole camera model to transform the 3D scene into a 2D image. Satellites use a different principle that will be briefly described later in subsection 2.1.1.

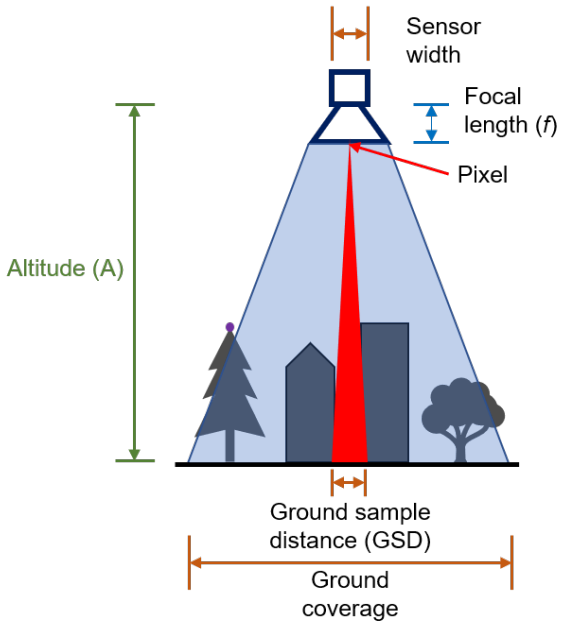


Figure 2.1: Image acquisition and related parameters.

Fig. 2.1 shows a basic diagram to explain how some parameters are related while acquiring the images. The camera is usually mounted on an aerial platform and follows the flight trajectory. The distance between the camera and the ground is the altitude or height of the flight (A) and depending on the focal length (f), the camera will cover a specific area on the ground that is the content of the image. In addition, for digital cameras, the acquisition sensor has a specific amount of pixels that define the image resolution in pixels (I_w). Such a sensor has a physical size defined by its width s_w. Each pixel covers a specific tile on the ground. The side of this tile is known as the ground sample distance (GSD) and defines the resolution in terms of meters. For aerial acquisitions it is common to use resolutions in the range of few centimeters, keeping a lot of details in the images.

These parameters are related as:

$$\text{GSD} = \frac{s_w \times A}{I_w \times f} \tag{2.1}$$

where A, s_w and f are given in centimeters (cm), I_w in pixels and the GSD in $cm/pixels$.

For the present dissertation, it is important to differentiate some concepts. In Fig. 2.2 two types of acquisitions are displayed. On the left, we show a nadir case, where the camera plane is parallel to the ground which is hard to set in reality. On the right, the acquisition is oblique, having the camera plane not parallel to the ground.

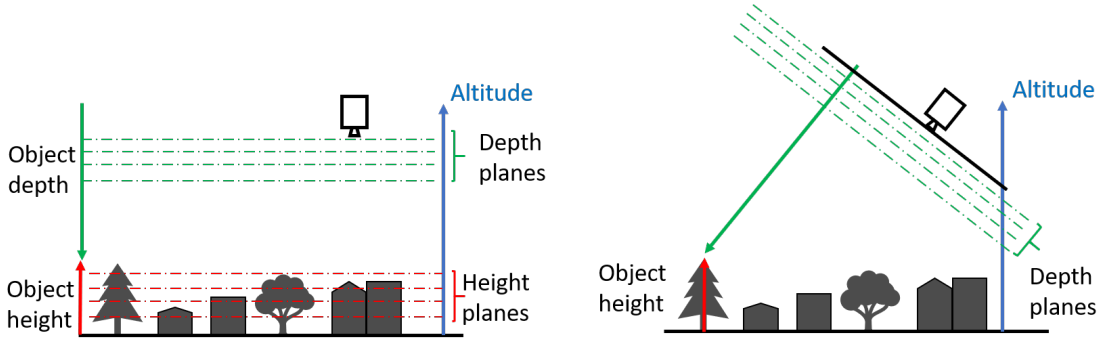


Figure 2.2: Nadir (left) and oblique acquisitions (right) for an aerial acquisition, highlighting the orientation of the camera and depth planes.

The reconstruction algorithms estimate the height of the objects on the scene (indicated by the red arrow) based on the depth, which is the distance from the objects to the camera plane. In a perfect nadir orientation: $A = h + z$ where h is the height of the evaluated point and z its depth, or distance to the camera plane. However, in practice the camera takes oblique images as shown on the right, where the camera parameters are known to properly convert the depth into height. As this is a general case, a 3D point in the real world with coordinates U, V, W is transformed into image coordinates u, v (u as column, and v as row) as:

$$\begin{bmatrix} u' \\ v' \\ w' \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -X \\ 0 & 1 & 0 & -Y \\ 0 & 0 & 1 & -Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix} \tag{2.2}$$

$$\begin{aligned} u &= (f * u'/w') + c_x \\ v &= (f * v'/w') + c_y \end{aligned} \tag{2.3}$$

where X, Y, Z are the coordinates of the camera in the 3D coordinate system and c_x and c_y are the principal point values. In many cases, both principal points are set to 0.

As mentioned before, depth and height can be computed from each other if the camera parameters are known. Assuming that the depth values are known after a 3D reconstruction algorithm was applied, the object coordinates can be found by solving the equation system:

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}^{-1} \left[\begin{bmatrix} \frac{(u-c_x)(-z)}{f} \\ \frac{(v-c_y)(-z)}{f} \\ -z \end{bmatrix} + \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \right] \tag{2.4}$$

where z is the depth for the evaluated pixel to be converted to a 3D point.

It is important to remark that transformations between coordinate systems might be required while transforming points from the real world into 2D images.

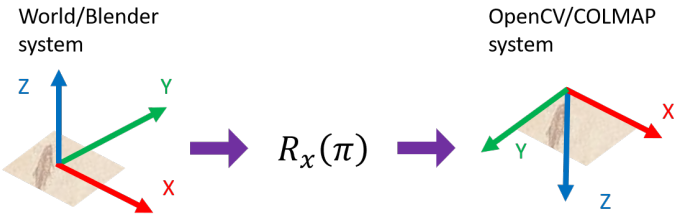


Figure 2.3: Image acquisition and related parameters.

For example, in Fig.2.3 two coordinate systems are displayed. On the left side, the specifications are given for the case of the Blender system and on the right, the one used by OpenCV and by the COLMAP pipeline[1]. A simple rotation, like $R_x(\Pi)$ is enough to go from one to another.

Another important aspect to consider while dealing with remote sensing imagery is the large size of the files and the available bands. Rasters might involve images that have dimensions larger than $10k$ pixels. Cropping into smaller patches helps to reduce the computational costs while processing, which is crucial for deep learning approaches. With respect to the available bands, for stereo and MVS algorithms either panchromatic or RGB images are used.

2.1.1 Satellite imagery

For satellite technologies it is common to use linear pushbroom cameras. Unlike traditional cameras, these sensors take one array (or line) of pixels at a time. Therefore, the image is formed by many adjacent sensor acquisitions. Although the use of linear pushbroom cameras might look more complicated, it is useful considering the orbit of a satellite, where the Earth and the satellite are moving at the same time [2].

Satellite images taken as input for stereo or MVS algorithms are usually preprocessed and aligned to cover neighbouring areas. Moreover, translating 3D point into images is not computed with the previous equations, but with rational polynomial coefficients (RPC). Some satellite images are even processed in a way to resemble the features of pinhole cameras as in [3]. For the present dissertation, the preprocessing of satellite data will not be addressed, but some satellite images will be used for the experiments.

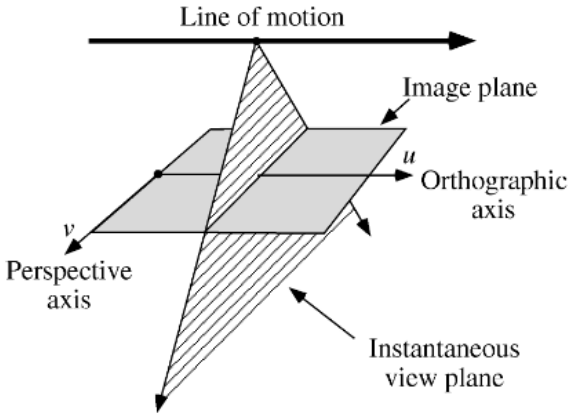


Figure 2.4: Geometry of a pushbroom sensor acquisition. Image taken from [2]

2.2 Stereo reconstruction

The stereo array configuration assumes that two cameras with the same parameters but separated by a distance B (called baseline) take a picture simultaneously, simulating the human vision process. As elements in the image show a horizontal shift between the two images, the human brain can estimate the distance of such elements.

A similar principle is applied in the computer vision community to estimate the distance from objects to the camera, as illustrated in Fig. 2.5. There, a stereo array is set above the scene, where the cameras are separated by the baseline B and both have a focal length f. To the side of each camera an image is shown that can be acquired from that perspective (see image left and image right, not related to the objects shown in the middle). As mentioned before, it is noticeable that the objects show a shift between the images, which is called disparity (d).

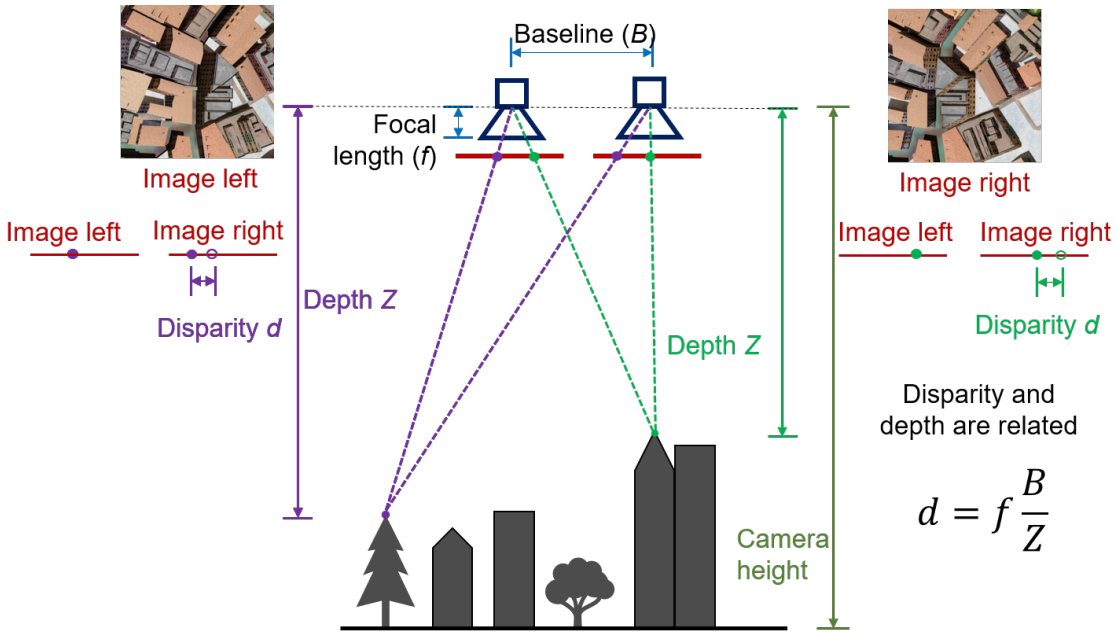


Figure 2.5: Stereo vision principle, where disparity and depth are related.

Considering the top of the pine (marked with a purple point) and tracing a line between this and the center of the cameras, a different position for the pine's top is given in each created image. The purple point at the "Image left/ Image right" diagram on the left illustrates the disparity between the two images. Similarly, an analysis for the top of the pointed tall building is shown in green. The disparity is related directly to the depth (Z) by using the equation:

$$d = f\frac{B}{Z} \tag{2.5}$$

Looking at the disparity-depth relation, the larger the disparity, the closer is the object to the camera. This can also be observed with the green point, which is taller (therefore with smaller depth) than the purple one. Hence, the disparity for the green point is larger.

If the disparity is known, using this value to generate a DSM is relatively easy. However, estimating a reliable disparity value for each pixel in the image is a challenging task. Most of the approaches (traditional and learnable ones) are based on a cost volume computation, as shown in Fig. 2.6.

2.2.1 Traditional stereo matching

Having a set of left-right images, the algorithms look for the position of the same pixel in both images along the same horizontal line. Corners or points with high contrast are easy to match, but in reality, the images might also include occluded areas (not visible in both images), blurry boundaries, textureless regions, illumination differences and noise from the acquisition, among others, which pose a challenge for the matching. By using a matching algorithm that compares the similarities between pixels a cost volume is generated. The cost volume approach first creates a volume with size $H \times W \times D$, where H and W are the height and width of the image and D the disparity range (commonly $[0, 192]$ for learning algorithms), so there is a value for each pixel in the image along the disparity range.

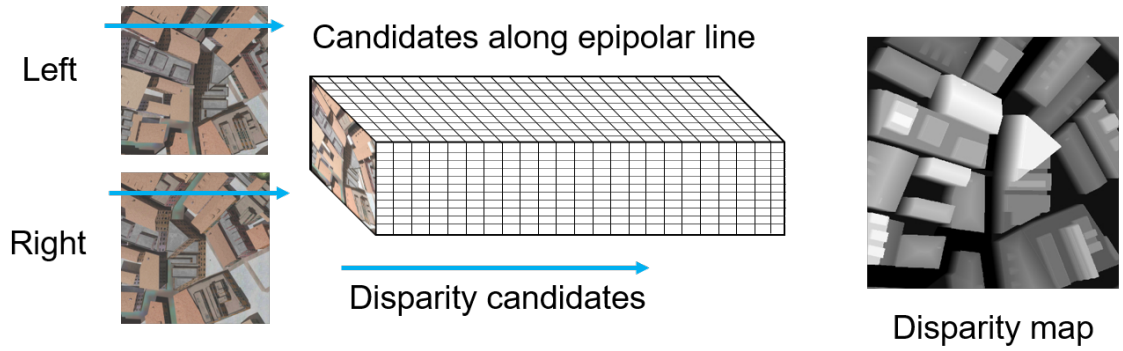


Figure 2.6: Disparity estimation based on a cost volume.

In the case of Semi-Global Matching [4], a cost that penalizes discontinuities in the disparity estimation is applied, benefiting from using context information from neighbouring areas. The SGM function in this algorithm is computed as:

$$E(D) = \sum_{p}(C(p, D_p) + \sum_{q \in N_p} P_1 T[|D_p - D_q| = 1] + \sum_{q \in N_p} P_2 T[|D_p - D_q| > 1]) \tag{2.6}$$

where the objective function $E(D)$ should be minimized for an optimal disparity map. The first term of the sum is the direct cost based on the similarity of the pixels (for pixel p using the disparity D_p) and an algorithm such as Census [5] can be used as suggested in [6]. The second term penalizes small differences in the neighbourhood of p (where q is another pixel in the N_p neighbourhood) with a penalty value P_1. The third term penalizes larger differences with the parameter P_2. T is a function to validate if the argument is true. SGM looks recursively along different directions to reduce the computational burden of analysing the whole image to get context information. Due to the good balance between accuracy and computational cost, SGM is one of the most widely used methods and a reliable option for remote sensing data [7].

Still, SGM is part of a pipeline for stereo matching, which encompasses matching cost computation, cost aggregation, disparity estimation and disparity refinement [8]. Additional steps such as left-right consistency check [9] are applied to remove remaining outliers in the prediction.

2.2.2 Learnable stereo matching

As for many tasks in the computer vision community, deep learning has achieved remarkable results and leads many state of the art solutions. This also applies to remote sensing tasks [10] such as semantic segmentation, object detection, image pansharpening or 3D reconstruction.

For the specific work of stereo matching, there has been also a series of neural networks which outperform traditional methods if the domain gap is not too large, being more robust to textureless areas or occlusions, and generating smoother disparity maps.

One of the first architectures was MC-CNN [11], where the step of matching computation is replaced with convolutional and fully-connected neural networks (CNNs and FCNs respectively). Feature extractors are applied to each of the input images and these are later concatenated to obtain a similarity score. The rest of the pipeline is based on SGM to enforce the smoothness of the result. MC-CNN reduced the error and the presence of outliers significantly, encouraging the research of more sophisticated solutions.

After that, the algorithms focused on including the whole stereo matching pipeline in a learnable way. DispNet [12] did not only improve the training of the networks by releasing the SceneFlow dataset, but developed an end-to-end strategy to estimate directly the disparity maps taking the stereo pair as input. The network has an encoder-decoder architecture with multiple convolutions, so context features can be learned in the coarse resolution and the details are recovered while upsampling. As the image features are not always processed at full resolution in all steps, this reduces the computational cost which makes real time inference possible. Another valuable study is GC-Net [13], where 3D CNNs are added to the architecture. While this type of convolutions improve the smoothness of the disparity map due to the 3D captured context, it also increases the computational cost.

Another significant work was proposed by PSMNet [14], which added a spatial pyramid pooling module that is able to learn the context at different scales, leading to improvements, especially for the occluded areas. EdgeStereo [15] has an additional sub-network that estimates an edge map, whose features are used for the disparity prediction, preserving edges and geometrical details from the input images. AMNet [16] benefited from contextual information by adding atrous convolutions at multiple scales and has an extended architecture that is fore- and background aware.

GA-Net [17] is an interesting approach, as it is based on a principle similar to SGM, where the costs for context are evaluated along different directions. It computes a robust matching cost using 3D CNNs and has an additional layer to refine thin structures. GA-Net is commonly used as a baseline to evaluate the performance of newer architectures. Nonetheless, it is a computationally expensive network and the inference times are slow. A solution to reduce the memory consumption is a coarse to fine architecture as suggested in GA-Net-Pyramid [18], which was tested for large remote sensing images.

AANet [19] follows a different strategy by replacing the heavy 3D convolutions with an architecture based on deformable convolutions instead. The cost volumes are handled in different scales, which helps to reduce the memory demands without missing context information. AANet is able to work on real time and its layers can be plugged into other architectures such as GANet, improving memory and inference time with a slight decrease in accuracy.

Despite the robust architectures for cost matching and disparity refinements, deep learning algorithms are in general affected by the domain gap problem. If the training and test datasets show large differences, many algorithms might fail to generate a disparity map. DSMNet [20] applies a "domain normalization" to the input images and uses a graph-based filter to capture geometrical features which also helps to generalize data. With DSMNet it was possible to train only on synthetic data and evaluate in real images to obtain a good quality disparity map.

Newer approaches use more sophisticated layers or features extractors to reduce the prediction errors and alleviate the domain gap problem. RAFT-Stereo [21] uses gated recurrent units (GRUs) [22], a type of layers that are able to keep or forget some features and are also commonly used for language processing. RAFT shows a more robust result for textureless areas, overexposure or fine structures. Its architecture can be adapted to work on real time without a significant accuracy loss. Other cases use transformers [23], an attention mechanism which transfers a sequence of data into another but without using recurrent networks. The STTR architecture [24] benefits from using transformers, which helps to avoid the limitation of having a fixed disparity range (as most of approaches use), handles better the presence of occlusion and strongly focuses on constraining the matching to a single candidate.

2.3 MVS reconstruction

The Structure-from-Motion (SfM) algorithms are commonly the first step for MVS, as these are able to retrieve the camera parameters for a set of input images. The usual pipeline for SfM algorithms follows these steps: detection of features in images, matching of the features, construction of 2D tracks based on the matching, solving the SfM algorithm with the tracks as input and using bundle adjustment to refine the SfM model [25].

Although the images were constrained to small changes in the early stages of SfM, methods have improved to consider larger distances between the cameras and using data from different acquisition times which is common in remote sensing. They also sort the images based on similarity, a useful input for MVS to select feasible additional views while processing.

However, the reconstruction of the scene itself is sparse in SfM, but its outputs are useful for the MVS algorithms that focus on creating a dense reconstruction. MVS can be seen as a more general case of stereo view, where the images come from various points of view. Assuming that the camera parameters are known, the matching of features between images is reduced from a 2D to a 1D problem, as optical rays can be traced between the camera center and its intersection in another image. Nonetheless, occlusions are a significant issue, since objects are not visible from all directions.

2.3.1 Traditional MVS

A widely used strategy applied for MVS is the plane-sweep algorithm [26]. It starts by assuming one of the views will be considered as the reference (see Fig.2.7). Starting at the camera plane of the reference view, a plane is swept along the camera frustum, creating a set of planes. The additional views are reprojected onto each of the planes via homography and these reprojections are compared to find the plane that defines the right depth for each pixel. This algorithm can be efficiently implemented in a GPU [27], where the depth value is chosen with a "winner-take-all" implementation from the computed family of planes. In this way, the set of depth hypotheses is similar to the cost volume from the stereo matching algorithms.

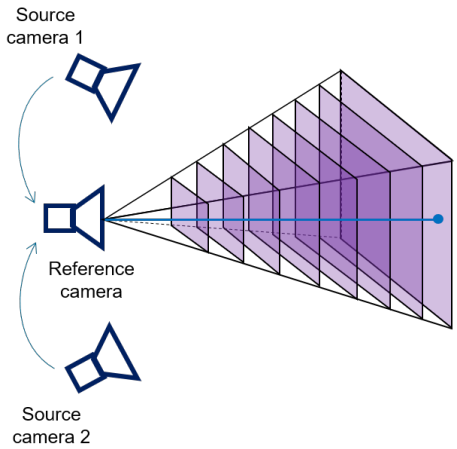


Figure 2.7: Sweep plane algorithm. A set of depth plane hypothesis is defined in the frustum of the reference camera.

COLMAP [1] is an open source pipeline for 3D reconstruction from imagery. Both algorithms SfM and plane sweep are included in the pipeline. The reconstruction is based on geometric principles and robust algorithms are added to handle noise and inaccuracies. Hence, reconstructions made with COLMAP are very accurate despite their relative sparseness. COLMAP is also used to retrieve SfM parameters for other algorithms such as camera extrinsics and it gives a score for the closeness of the additional views.

Another popular open source strategy is GIPUMA [28] which follows the principle of PatchMatch [29] applied for stereo matching. Here, the depth candidate planes are randomly initialized and the best-fitting ones are propagated to refine the estimation. This showed a good performance in terms of accuracy and used low computational resources.

COLMAP and GIPUMA are still widely used in academia and industry as many steps have been refined and designed to be more robust in the last years. As it is based on geometric principles, algorithms for ray tracing, reprojections and occlusion detection help to remove errors and outliers. Despite the success of learnable approaches, the physical interpretation of the learned features is not easy to understand and many outliers might remain in the results as these are hard to identify.

2.3.2 Learnable MVS

As for the stereo matching case, learnable algorithms have topped the MVS benchmarks. The baseline for most of the approaches is MVSNet [30]. The main idea is to make the homography in a differentiable way, so it can work in an end-to-end architecture. Hence, input images along with the camera parameters are given as input and the depth is directly the predicted result.

Although the networks architecture differs in many elements, the creation of a cost volume for depth candidates is usually present. In Fig. 2.8 we show the principle of learnable networks in a very simplified way. A set of images, where one of them is used as reference (in this case view

N) are given as input. As the camera parameters are known, the candidate depth planes are projected along the frustum. The homography (H), which is implemented in a differentiable way is computed as:

$$\mathrm{H}_i(d) = \mathrm{K}_i \cdot \mathrm{R}_i \cdot \left(\mathrm{I} - \frac{(\mathrm{t}_1 - \mathrm{t}_i) \cdot \mathrm{n}_1^T}{d}\right) \cdot \mathrm{R}_1^T \cdot \mathrm{K}_1^T \tag{2.7}$$

where $\mathrm{H}_i(d)$ represents the homography at depth d for the i^{th} view ($i \in [1, N]$). K, R and t are the camera instrinsics, rotations and translations respectively. n and I represent the principle axis and the image of the reference camera (labelled as 1). By using this homography it is possible to set the image features in a cost volume to be regularized. Unlike the usual plane sweep algorithm, a fixed number of planes (D) is set since the beginning. The first and last depth planes are also defined by the depth range which is included in the camera parameters. For a reference image with dimensions $H \times W$ (where H is height and W is width), a volume of depth candidates with dimensions $H \times W \times D$ is created. In practice, many networks use a downsampled version of this volume due to memory constraints. After that, the expectation value is computed from the depth candidates as:

$$\mathbf{D} = \sum_{d=d_{min}}^{d_{max}} d \times \mathrm{P}(d) \tag{2.8}$$

where $\mathrm{P}(d)$ is the probability of the estimation at depth d and $\mathbf{D}$ the output depth map.

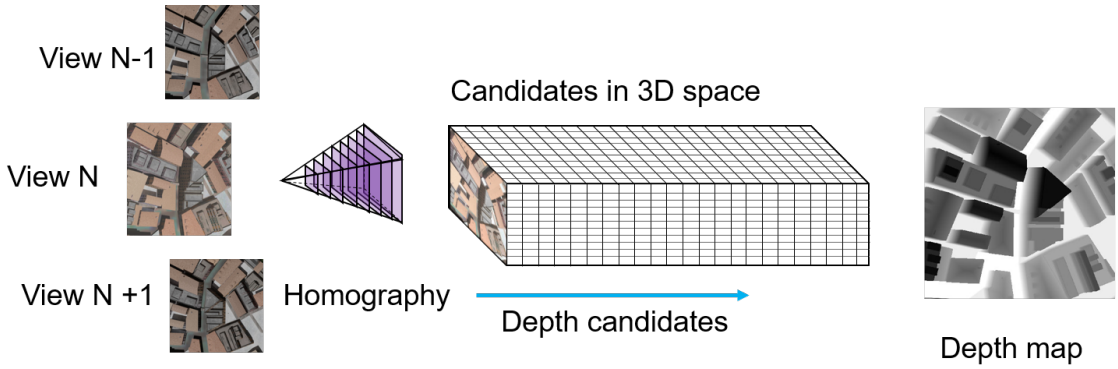


Figure 2.8: Simplified representation of the depth prediction in MVS learnable architectures, where a cost volume is used to match the features from the input images.

Of course, algorithms are not that simple and take many considerations into account, such as robust feature matching or refinement steps for the predicted depth maps. MVSNet shares weights for the feature extractors which are used to build the cost volume according to the variance of the feature inputs. The input images are selected according to the sorting suggested by processing the same images with COLMAP. Then a multi-scale set of 3D CNNs is applied to refine the cost volume and generate an initial depth map. As many outliers are still present and the result is not smooth in some regions, an additional depth refinement guided by probability maps computes an improved final depth map. This network showed significant advances

with respect to COLMAP or GIPUMA, especially in terms of completeness. Still, as for deep learning solutions, it also performs well if the domain gap is not large.

The same authors from M-VSNet proposed R-MVSNet [31] as a design that can handle images with larger sizes without the burden of heavier computational costs. This architecture regularizes 2D cost maps in the depth direction instead of the 3D cost volume by using GRUs. A cross-entropy loss is applied, as the network considers the task as a classification problem. It achieves slightly better accuracy and completeness than MVSNet.

CasMVSNet [32] proposes handling the cost volume in a coarse to fine strategy, where the coarsest volume comprises the whole disparity range with a small block resolution and wider resolutions only search in a narrower depth range for a finer estimation. The design can be applied to existing networks such as MVSNet or R-MVSNet. Moreover, it can also be adapted to enhance GANet or PSMNet for the stereo matching task. Fast-MVSNet [33] focused on an architecture capable of processing the reconstruction in real time situations with a sparse to dense initial depth prediction and a coarse to fine refinement using a Gauss-Newton layer.

AA-RMVSNet [34] proposes adding long short-term memory (LSTM) layers as an adaptive aggregation. Two modules were added: one for improving challenging regions based on the context and the second to prioritize the better-matching pairs from the inputs. A recurrent network is used to refine the cost volume instead of 3D CNNs. Another interesting approach is proposed in Vis-MVSNet [35] which addresses the problem of the pixel-wise occlusions for matching areas. For each input pair, the algorithm does not compute only an initial depth estimation but also an uncertainty map that selects which features are added to the fused cost volume. Hence, this fused cost volume works with features less susceptible to be affected by occlusions.

UniMVSNet [36] is a remarkable work where the depth prediction is considered both a regression and a classification task. The classification part selects the closest depth plane to the expected value and the remaining distance to this plane is estimated by the regression part. To achieve this, the network includes a unified representation of the ground truth which helps during the training to further refine the cost volumes. The results showed higher completeness and can handle large resolution images, as the architecture uses a fine to coarse design.

Newer concepts in deep learning such as transformers can also be applied to the MVS task as is the case for TransMVSNet [37]. The network firstly computes features with a pyramid network which are later refined by an adaptive receptive field (ARF). The ARF modules ensure that the original features can be passed to the robust feature matching transformer. These new features are correlated in a common volume and with the guidance from a probability volume, the depth map is computed. Again, this architecture is designed in a coarse to fine manner.

Recently, GeoMVSNet [38] focused on the geometry and included more structural features. In a coarse to fine architecture, an additional geometry embedding volume is computed after each cost volume refinement which is passed to the next stage to keep relevant geometric information. Besides, a frequency domain filter is applied to reduce the effect of outliers.

One of the latest architectures is IGEV-Stereo [39], which is based on RAFT and encodes additional geometric and contextual information to tackle difficult regions, such as ill-posed

ones. Although its main application is for stereo matching, with some changes it is able to work as a MVS algorithm. What is more, IGEV-Stereo is one of the leading solutions in both stereo and MVS benchmarks.

2.4 Fusion of disparity/depth maps

The prediction of depth and disparity maps is not the final step in 3D reconstruction. The estimated depth values can be converted into a point cloud, a height map or a voxel based representation. For the present dissertation, the fusion of disparity/depth maps to get a DSM is studied. As the camera acquisitions usually cover just a part of the region to be analyzed and the algorithms might require even smaller tiles to work due to computational costs, a stack of small DSMs have to be fused into a large one. Having a set of possible values for each pixel also helps to reduce the influence of outliers and create a denser result.

If the data follows a stereo matching setting, the input images are processed to compute a disparity map. However, this map has a meaning only in the 2D image domain as it defines the relation between the two images. The disparities can be converted into depth as described in equation 2.5, so the distance from the camera to the objects can be estimated. The next step requires converting the depth into height values (the difference is explained in Fig. 2.2) and height values can be computed with the Eq. 2.4.

Still, the height values have to be processed for a DSM and that affects the density of the result on the ground level. To explain this in an easier way, look at Fig. 2.9 (left). When the depth values are converted to height, these are still camera oriented, which means that for each pixel we have a height value. For a DSM instead, the rasters are not in camera pixels, but each pixel represents an area defined by the GSD. In the same figure (right) and focusing on the far left building, we notice that the right side of such building is covered by three pixels, but all these results are mapped into one area cell on the ground. Hence, the density of the DSM on the ground depends on the visibility of the objects from the camera perspective.

If we start with a MVS algorithm and depth prediction, the translation to depth is not needed and the rest of the process is the same as just described, so both stereo and MVS solutions map their results onto a grid on the ground. As the grid is defined by the GSD, it is necessary to merge the measurements (if any) that are above each cell. Facades or walls which are largely visible in the image will have many measurements, while occluded areas get no measurements at all.

For non-occluded areas, the measurements are stored in a raster. Some methods take the maximum, minimum, mean or median from all the measurements over a ground cell to select the value to store. While this is an efficient way, many areas remain with no defined values. It is possible to use interpolation algorithms [40] that fill in the missing measurements reasonably well and create smooth surfaces, such as the widely used inverse distance weighting (IDW) or Kringing (known as Gaussian process regression in some literature). The implementation of interpolation algorithms is a recurrent tool in DSMs generation.

Fortunately, it is also common to have more than one stereo pair or one pair of MVS images to generate a DSM. Flight and satellite campaigns acquire enough data to reduce occlusions and

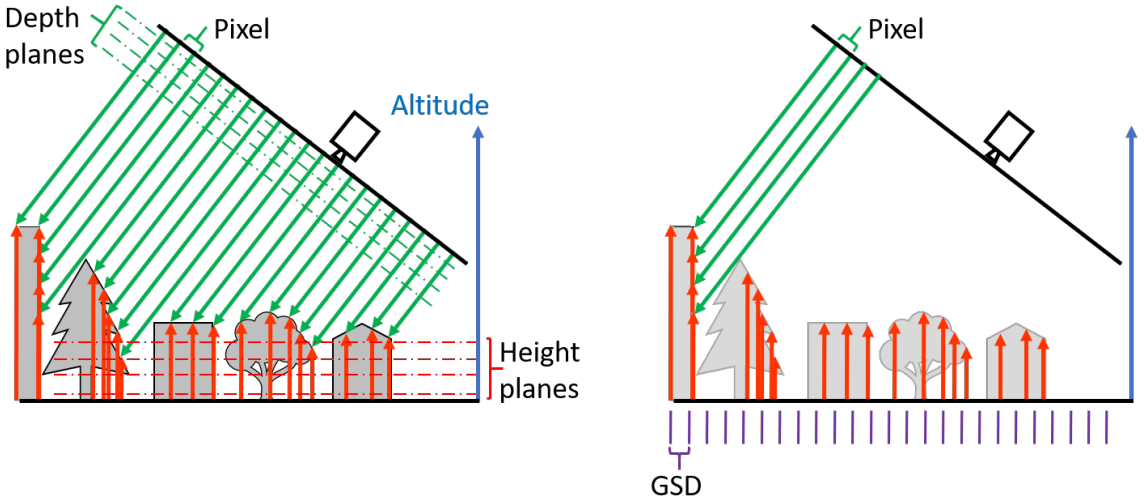


Figure 2.9: Relation between pixel and GSD in an aerial acquisition. On the left side a set of depth planes intersect the objects in the scene, which are defined by height planes. On the right, the sampling discrepancy between camera pixels and GSD is highlighted.

allow the overlapping of measurements in some regions. Hence, measurements for one DSM cell might be present in multiple disparity/depth map estimations. The easiest way to merge multiple measurements is using the average or a weighted average from them. However, the predictions are neither normally distributed nor free of outliers, which causes the averaging to produce significant errors. Fusing the data by computing the median from the available estimations is a more robust option [4], as it is less influenced by outliers. Median fusion has already been applied to satellite data [41]. An additional weight average of values close to the median can also be helpful. Strategies for a more sophisticated and robust fusion have also been studied by the research community.

An extended survey in these methods can be found in [42], where the authors analyse the main steps of Digital Elevation Models (DEM) fusion, namely: selection, pre-processing, fusion, post-processing and quality assessment. For this dissertation, the object of study is only the fusion process.

The most basic case is just applying the average of the measurements as in [43], where DEMs are created for urban flood assessment. This might work well if the inputs are from the same sensor or if no significant differences are expected in the measurements. Unfortunately, those conditions are not common in practice and a weighted averaging is a more optimal solution. DEMs are fused in [44] with weights related to the accuracy of the measurements and the method is applied to different satellite sources. The weight averaging can be computed as:

$$h = \frac{\sum h_i \cdot w_i}{\sum w_i} \tag{2.9}$$

where h_i are the height values from the i-available DSMs and w_i the weight assigned to each of them. In the case proposed by [44] each weight is computed using the given accuracy a_i as $w_i = 1/a_i$.

Fusion with sparse representations was introduced in [45], where the main idea is to get information about the surface profile, such that values that are inconsistent with such shapes are discarded. In another study [46], weighted averaging and fusion with sparse representations are compared, showing a similar performance on the tested satellite data.

A medmean fusion is proposed in [47]. Here, values that are within a threshold (set empirically to 2m) of the median are averaged to reduce the amount of outliers. The author also suggest using Total (Generalization) Variation based methods (TGV and TV) which are denoising models. These last algorithms can produce smoother surfaces, especially for rooftops, but medmean is still very competitive in the metrics with a simpler operation.

A different fusion method was described in [48]. In this case, the reference and target DEMs are swapped. Only the estimations that are similar remain, while the others are considered false predictions. The method additionally gives insights of uncertainty in the geospatial domain.

The weighted fusion can also be benefited from learnable architectures if the weights themselves can be learned as suggested in [49], where optical and SAR satellite data were fused. The learned weights helped to improve the accuracy in comparison to normal weight averaging and total variation methods. A fusion with k-means clustering was applied by [50], but it showed that depending in the clustering parameters some noise can be generated, degrading the quality of the final DEM.

A fusion with an adaptive 3D median filter was suggested in [51] to remove the outliers in flat regions. It uses the color information from the images and the height values within a window which is centered in the pixel to compute. The images help to to get spatial information for consistency and smoothness.

2.5 Confidence/uncertainty estimation

Regardless of the advances in deep learning approaches, there is one more point that has an impact on these algorithms: the confidence of the prediction. To understand this issue, observe Fig. 2.10 where a pair of stereo images is given as input. Some regions in the image such as the textureless sea and the parts of the facades that suffer from occlusions represent a challenging matching task. The first presented disparity map, where a conventional method was applied (SGM to be exact) produces a good result as most of the urban elements are reconstructed and values were similar to the ground truth. Besides, the algorithm is able to discard areas where the matching/refinement process is not reliable enough, although that reduces the amount of result pixels.

On the other hand, deep learning solutions working in an end-to-end manner (which means the refinement process is also learned) estimate a value for each input pixel, even if the values in the cost volume do not show a clear best candidate. Hence, the produced disparity map has no issues in terms of density, but many outliers remain. It is noticeable that estimated values for the sea region are not consistent and values in the occluded areas may include significant errors, despite the interpolation capabilities of the networks. Therefore, an algorithm able to tell whether the predicted result is reliable or not would be helpful to remove outliers before fusing the data for a final DSM.

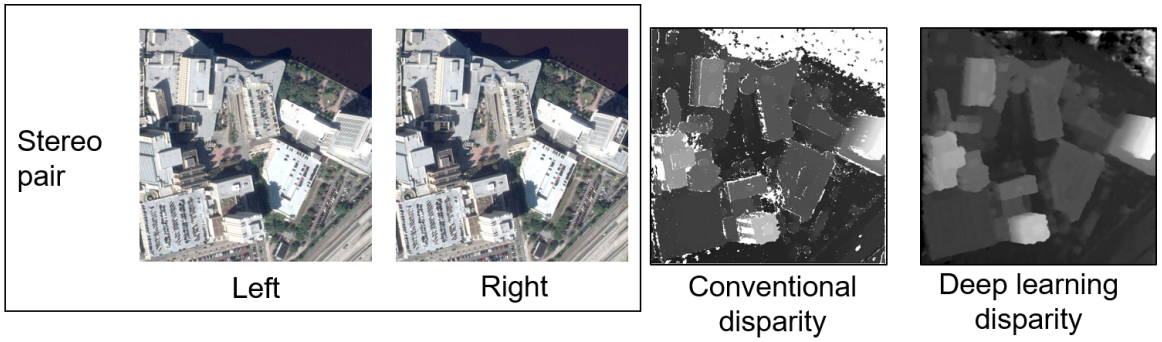


Figure 2.10: Differences between conventional and deep learning disparity estimation, where the latter computes a result for all pixels.

This topic has been addressed more for stereo than for MVS algorithms. As for most of methods nowadays, this can be computed with conventional and learnable algorithms. A review of conventional cases can be found in [52], where six main categories of estimation algorithms are discussed:

- Matching cost. These algorithms invert the cost to estimate the confidence. The idea behind is that the lower the cost, the higher the confidence. As single values in the cost volumes do not represent much information, this method performs poorly.
- Cost curve around the minimum. If the cost curvature is flat or sharp around the selected candidate, this can be used as a confidence criteria.
- Presence of other minima in the local cost curve. If other strong candidates exist around the chosen one, it has low confidence. Noise-ratio values are also commonly used.
- Behaviour of the whole cost curve. For these cases the cost curve is analysed as a probability function. The position of the selected candidate within the function and the number of inflection points are used among other criteria to set a confidence value.
- The consistence between the left and the right predicted disparity maps. One of these algorithms is based on the left right consistency check (LRCC). If the two disparity maps (left to right and right to left) are available, it is possible to warp a pixel to the other image and back, observing the difference with respect to the original location. The lower the discrepancy, the higher the confidence.
- Measures based on Distinctiveness. In an ideal case, salient points are easier to match and thus more confident. Hence, pixels that are dissimilar to the neighborhood are more confident. Input images, cost volumes and the predicted disparity maps can be given as input for these algorithms.

SGM pipelines usually include the left right consistency check (LRCC) as a refinement step to discard bad predictions. By warping the pixels with both disparity maps and comparing the new positions with the original ones, it is easy to spot some of the big outliers present in the result. Occlusion and false matches are thus removed from the predicted disparity maps, which is the reasons why the pixel density is reduced in the final result.

Learnable approaches for confidence estimation are also being developed by the research community and a detailed review about them can be found in [53]. Similar to the non-learnable cases, inputs are usually reference RGB images, predicted disparity maps, ground truth disparity maps (required for the learning iterations) and cost volumes. Nonetheless, strategies based only on the cost volume are not addressed, as deep learning benefits more from multiple convolutions on 2D inputs and 3D inputs are computationally expensive. Still, parts of the cost volume can be used as a complementary source of information.

One of the first proposed architectures was CCNN [54]. It used the predicted disparity map and the ground truth to estimate the confidence, a strategy that is easy to apply as the cost volume is not required. A set of CNNs and FCNs process the disparity map and the training works with a Binary Cross Entropy loss (BCE), where there are two possible values (or labels): 1 for confident and 0 for non-confident. The difference between the ground truth and the predicted map is used to set the confidence condition. If the difference is larger than 1 pixel, the value is non-confident and labelled with 0. This criteria to decide whether a pixel is confident or not was a basis for newer networks. CCNN got better results than the conventional methods which encouraged more sophisticated architectures.

A different architecture is used in PBCP, which even applied the results from MC-CNN. Thus, a learnable algorithm was implemented to evaluate the confidence of the result of another learnable algorithm. The predicted confidence map was added to the SGM pipeline and helped to reduce the error prediction up to $1/3$. It is important to mention that the detection of non-confident pixels in the results from MC-CNN is easier than in more complex networks, since wrongly predicted values from non-smooth regions are easy to spot.

PKRN+[55] proposed a CNN that further enhances the confidence estimation. By using the results from any confidence algorithm (conventional and learnable ones), these are used as input for a network that captures more local context and an enhanced confidence map is computed as output. The purpose of this network is to generate a smoother confidence map and reduce noisy predictions. Such architecture leveraged the performance of CCNN.

A more elaborated approach is UCN [56] where the cost volume is used as an additional input. UCN includes two sub-networks, the first one processes the cost volume and the second one aims to predict the confidence map. In the first part, the raw cost volume passes through a series of convolutional layers with skip connections. However, the cost volume is a large size block to be processed simultaneously with the disparity map for the second part. Therefore, just the most relevant values of the cost volumes are selected with a top-K pooling layer which is implemented in the most popular DL frameworks. The top-K volume along with the predicted disparity map are used in the second part to estimate the confidence with another set of CNNs. The test cost volumes were processed with Census and MC-CNN, showing a good performance in both cases.

LAFNet [57] considered a more robust estimation by taking the predicted disparity map, the reference image and the cost volume as inputs. The network extracts firstly the features of the three inputs separately with a set of CCNs, the weights are not shared as each branch has a different type of input data. After that, an attention inference network assigns a weight to each feature block for a better concatenation than simply adding all of them with the same

importance. A scale inference network helps to set the optimal receptive fields for better context learning and at the last stage, a recursive network is used to refine the predicted confidence map. This network outperformed the other approaches, but since it requires the cost volumes as inputs, not all approaches for stereo matching can be used as a previous step if the cost volume is not available. For example GANet creates a cost volume with multiple channels $H \times W \times C \times D$ (where C are the channels) that cannot be directly used unlike MC-CNN that outputs a regular one $H \times W \times D$.

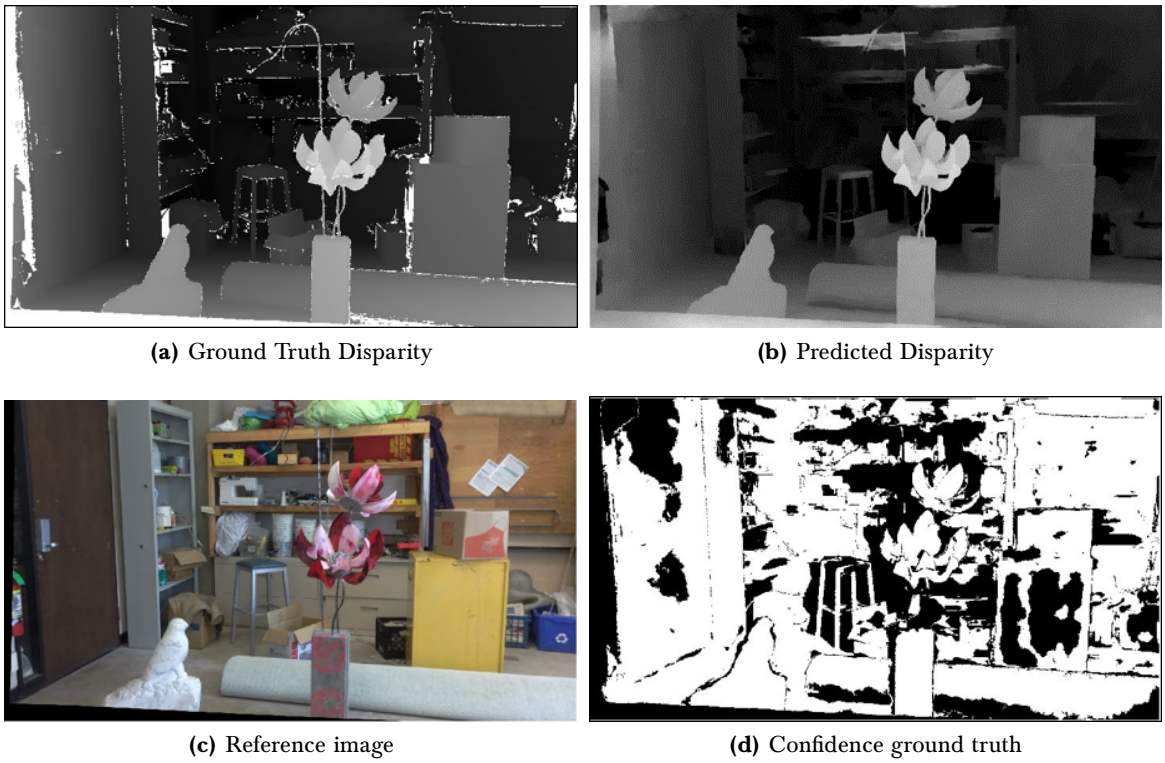
The newer SEDNet [58] estimates both the disparity and the confidence map simultaneously at multiple resolutions. The network uses GwcNet [59] as the algorithm for stereo matching and this generates the cost volumes and disparity maps. Additional layers compute the uncertainty (it can be seen as the inverse of the confidence) based on the error of the disparity prediction. The result shows how SEDNet is able to detect non-confident pixels better than LAFNet trained with GwcNet outputs.

In the case of MVS, some networks use the uncertainty or confidence estimation within the network to generate a more robust depth map, although it has not been deeply studied yet. VisMVSNet uses a uncertainty estimation as an indication for visibility . The depth map is usually computed from a probability volume in MVS approaches and for this architecture, the probability is considered an indication of the matching quality. An entropy map is computed from the probability map and passed by a shallow CNN to estimate the uncertainty. The training loss considers both the disparity estimation and the uncertainty, which are then fused to create the volume that computes the final depth map. For UniMVSNet, the uncertainty is also computed from the probability volume but was not used for the cost volume refinement.

In Fig. 2.11 a confidence map is shown to understand what it represents. For a stereo pair, where the left image is 2.11c with corresponding disparity ground truth in 2.11a, a disparity map is computed. Using AANet with a model trained on the SceneFlow datasets, a predicted disparity map is obtained as shown in 2.11b. The confidence requires a predicted disparity map to estimate which pixels have a better probability to be correctly computed. Hence, by computing the difference between the ground truth and predicted disparity maps, a confidence map can be created. In 2.11d the white areas show a confident region, where the difference is less than 1 pixel. Regions in black have a difference larger than 1 pixel instead and are labelled as non-confident. The displayed disparity maps have a scale of 50 to 120 pixels and the reference image is "Artroom1" from the Middlebury dataset in its 2021 version [60].

2.6 Existing datasets and limitations

Since deep learning algorithms became a popular way to tackle computer vision algorithms, the demand for large amounts of quality data has been increasing. Because of that, in this section the available datasets for stereo and MVS are addressed, as well as the pros and cons of their usage. Besides, a brief summary of datasets for change detection is included.



(a) Ground Truth Disparity

(b) Predicted Disparity

(c) Reference image

(d) Confidence ground truth

Figure 2.11: Confidence ground truth generation. For the reference image 2.11c, the disparity ground truth is shown in 2.11a. Using a stereo matching algorithm, the predicted disparity map in 2.11b is computed. From the difference between 2.11a and 2.11b, the confidence map 2.11d is created. White is confident, which means the difference was less than 1 pixel. The reference image belongs to the Middlebury 2021 dataset.

2.6.1 Datasets for stereo matching

Even before learnable approaches became the trend for stereo matching, some datasets were used as a common reference to evaluate the performance of conventional algorithms. One of the popular initial datasets is the Middlebury dataset 2001 [8]. This is a small but challenging dataset, as the subsets of images are not similar to each other, include regions with repetitive patterns, textureless areas and planar objects in oblique perspective.

The Middlebury dataset was enhanced in 2003 [61] with samples that are photographed using a structured lighting technique to provide accurate ground truth disparity maps. More complex 3D objects were incorporated in the captured scene. Later on, the dataset was expanded in 2005 [62] and 2006 [63]. These datasets were provided in a format and size that for the first time suited learnable algorithms. Their publication was also used for testing a Conditional Random Field (CRF) to estimate disparity maps and to test different matching costs.

The Middlebury dataset was further expanded in 2014 [60] with an improved acquisition technique for sub-pixel accuracy ground truth generation and a new approach for self-calibration. Containing large resolution images and capturing more complex scenes, the dataset is more challenging than previous versions. Finally, it was lastly updated in 2021 with samples taken with the same technique as the 2014 version but the camera was replaced by a mobile device.

The "torch" device light and its flash were used to create a variety of lightning conditions.

Despite the accurate ground truth and good quality images, Middlebury is usually used for finetuning instead of training. The main reasons are its relatively small size (comparing to other datasets with $> 1k$ samples) and the domain gap when testing, as few real datasets resemble the features in the included samples. Interestingly, RAFT-Stereo suggests using its model finetuned in Middlebury for unconstrained samples.

Another significant dataset is KITTI in their 2012 and 2015 versions [64, 65]. KITTI is a dataset created for autonomous driving tasks and includes images taken from a car perspective around the German city of Karlsruhe. A stereo camera array was set on top of the car to generate the stereo imagery and a LiDAR sensor to measure the distance to the objects. The size of the dataset is 400 image pairs, which is enough to train deep learning architectures, but it is also common to use it for finetuning.

One of the main features of KITTI is the complexity of their images because the scene is not constrained. Difficult aspects for stereo matching such as illumination differences, occlusions, reflective and transparent surfaces, wide disparity ranges, moving objects and repetitive patterns are present, which made the dataset a challenging benchmark for many years. Nonetheless, due to its relative small size, some algorithms might easily overfit to the training samples. The 2015 version included a denser ground truth, where 3D cars models were incorporated to fill in the missing surfaces and moving objects were added. One of the disadvantages of this dataset is the lack of a denser ground truth, which for moving objects is just an approximation, not a direct measurement from the laser.

As an alternative to compensate the lack of large databases and considering the struggle of creating more samples in a real scenario, synthetic datasets emerged as a feasible solution to easily generate thousands of samples. Furthermore, in synthetic datasets it is easier to manipulate the scene and retrieve an accurate ground truth, seeing that the 3D software has all the geometric definitions required for it. Diversity of textures and geometrical shapes, and modelling real camera parameters help to enhance the quality of synthetic data [66].

The Sintel dataset [67] is generated from the short film with the same name. Diverse conditions were applied when rendering the same scenes, adding effects such as blurring, specular reflections, illumination and atmospheric effects, which enhance the variety of the dataset. A total of 1628 frames are provided as ground truth, larger than previous datasets, and it was used as a benchmark for some years. However, models trained on Sintel showed a limited performance in some real datasets. This raised awareness that the domain gap problem also was significant for learning stereo matching architectures.

The release of the SceneFlow datasets [12] is known to have contributed to the development of stereo matching architectures. With 35k training frames, this synthetic dataset is much larger than earlier datasets. Besides, it includes a wide variety of textures and geometries. There are three subsets included: 1) Driving, resembling the point-view of a car on a street level, similar to the KITTI tiles with forward and backward driving perspective, 2) Monkaa, rendered from the short film with the same name; this might not resemble natural scenes but the motion of the objects is hard to track with optical flow; and 3) FlyingThings3D takes objects from many categories (like chairs, cars, planes, furniture, etc., following random 3D trajectories. A large

pool of objects and textures is used to generate the 3D scenes, creating also a large diversity of content for the samples.

SceneFlow datasets are widely used either to train a model and test directly in unseen samples, or as a strategy to pre-train models that are finetuned with smaller real datasets. This last strategy benefits from the complex geometry and texture cases present in the dataset aside from the accurate available ground truth.

A demanding dataset for stereo matching algorithms is the ETH3D [68] in its low resolution two-view case. Stereo rectified gray scale image pairs are provided together with the laser scan ground truth. The images were taken in real-world scenarios with non-constrained laboratory environments.

The previous datasets are oriented for general stereo matching or the autonomous driving task. Yet, for the scope of this dissertation the remote sensing imagery is the final application. Since not all the areas where imagery is available include LiDAR measurements, there are few large datasets to train deep learning approaches.

One of the first remote sensing datasets covers the area above Catalonia in Spain [69]. Data from the Cartosat-1, Worldview-1 and ALOS/PRISM satellites is processed to generate the stereo pairs. Three different regions including urban areas, hills, steep mountains and forests are included. The ground truth is derived from an airborne laser scanning campaign. Due to the resolution of the satellite images (0.5m-2.5m) the reconstruction of the valleys and hills can be achieved with good quality, but buildings face difficulties to get sharp edges.

The EuroSDR/ISPRS benchmark [70] was released to encourage the research for DSM generation. Two main areas are included: Vaihingen and Munich. The former includes a rural landscape with small houses, crop fields and forests with a GSD of 20cm at a height of 2900m. The images have an overlapping of 63% and 62% for flight and cross direction.The Munich set is oriented to urban reconstruction with higher buildings, streets and complex man-made structures with a GSD of 10cm and an overlapping of 80% and 80% for flight and cross directions. While the terrain in this case is mostly flat, the dense of the buildings is very high. In the Vaihingen case few buildings are present but the terrain height differences are up to 200m. An enhanced version of the Vaihingen dataset for stereo matching learning architectures was presented in [71], where an additional study for the impact of the B/A is included.

Another known benchmark is the US3D dataset [72], which was used for the Data Fusion Contest (DFC) 2019 [73]. The images of the dataset cover an area of about 100km^2 over the cities of Jacksonville, Florida and Omaha, Nebraska. The images are captured by the WorldView-3 satellite in panchromatic, and visible and near infrared (VNIR) formats. The GSD for panchromatic is 30cm and 1.3m for VNIR. The ground truth disparity maps are computed from an airborne LiDAR source with a pulse spacing of 80cm. The imagery also include semantic labels used for the contest task which aimed for a semantic stereo reconstruction. With 4292 images included, it has been used to train deep learning models.

The WHU-Stereo dataset [74] provides 1757 images in panchromatic mode obtained with the GF-7 satellite and covering parts of 6 Chinese cities. In total, the images represent an approximate area of 900km^2. The GF-7 satellite mission is equipped with a duel-line stereoscopic

camera that samples a GSD of 0.8m and 2.6m for panchromatic and multispectral imagery respectively. The ground truth is computed from airborne LiDAR measurements with a nominal pulse spacing of 25cm. The released dataset contains the already epipolar rectified image pairs with the corresponding ground truth, where urban and rural areas are represented.

2.6.2 Datasets for MVS

In a similar situation to stereo matching, some datasets have been created in the last years to address the MVS task, few even considered benchmarks. As already mentioned, the format of these datasets is different to the stereo ones, as they are designed for another kind of algorithms and architectures.

One of the most widely used datasets is DTU [75]. It comprises 80 scenes, which are captured from 49 or 64 positions each. By using an industrial robot arm and a light scanner, camera positions and ground truth are both very accurate. The scanned objects are defined by dense point clouds, with around 13.4 million points. However, the scans do not cover the whole objects due to self-occlusions. The distances between the center of the scene and the cameras are 35cm and 65cm. DTU has been used as a benchmark that evaluates two main parameters:

- Accuracy. Measures the distance between the estimated points and those from the reference, it is an index of the quality of the reconstruction.
- Completeness. It also measures the distance between estimation and reference but focusing on how much of the surface was reconstructed.

Many algorithms can achieve a very high accuracy by keeping just those points where the estimation is considered reliable, but it has an impact on the completeness. A good performing algorithm should get good results for both metrics.

Another dataset used as benchmark is Tanks and Temples [76], which includes indoor and outdoor imagery with real conditions. The ground truth is generated with an industrial laser scanner with a range up to 330m with high accuracy, having a noise of only 0.1mm at a distance of 10.2m. The images come from high resolution video sequences. Two subsets are released: intermediate and advanced. The intermediate one contains smaller objects with outside-looking-in camera paths and the advanced is oriented to larger scenes, with a more complex geometry and camera paths.

The ETH3D dataset comprises also a pair of MVS sets. The high resolution one includes relatively few samples (454) that were recorded by a digital single-lens reflex (DSLR) camera and the low resolution set has a large number of images (4796) taken with a multi-camera rig. In the same way as for stereo matching, complex indoor and outdoor scenes are included. The authors also evaluated conventional methods for MVS reconstruction observing a good performance for COLMAP, where completeness and accuracy were the evaluated metrics.

With the advancement of deep learning solutions, the demand for data increased and some of the presented datasets might be small to train a robust network. Therefore, and similarly to the SceneFlow datasets, synthetic data was generated to create a large pool of MVS imagery with accurate ground truth.

The BlendedMVS dataset [77] contains 17k training samples which is enough to train architectures in a robust way. As for other synthetic datasets, a common strategy is to train in the large amount of synthetic samples and then to finetune in the domain of application to have a robust, yet domain adapted model. BlendedMVS is based on 113 images that resemble small objects, buildings, residential areas, indoor environments, etc. The authors focused on a right alignment between images and depth maps while rendering, and on simulating the effects of lighting conditions that the scenes would have in a real setting. The released dataset provides images with a resolution of 1536×2048 pixels, files with camera parameters and a suggested depth range so it can be directly be used as input to train learnable MVS algorithms. Despite its synthetic nature, R-MVSNet models trained on BlendedMVS showed better results when tested on the Tanks and Temple dataset than models trained with DTU or ETH3D.

The remote sensing community has also created MVS datasets as such as WHU-MVS [78]. The original images were acquired with an oblique five-view camera rig mounted on an UAV. These images were processed via software to generate a 3D DSM. The covered area is located in the Guizhou province in China, where elements such as forests, urban areas, industrial sectors and nature landscapes are present. From the generated DSM, the released dataset is generated in a synthetic way simulating the acquisition from a single-lens camera and retrieving the 3D measurements for the same area. A total of 1776 images with the respective ground truth are provided with a resolution of 5376×5376 pixels.

The US3D dataset was also adapted for MVS reconstruction by setting multiple stereo rectified pairs for the same areas. Additional unrectified images are also given with the corresponding RPC metadata and a normalized ground truth in height coordinates. All the required metadata for rectified and unrectified images was shared by the authors. This data was used for the track 3 in the DFC 2019. A sorting of suggested additional views was implicitly required for a better performance of the tested algorithms.

2.6.3 Datasets for change detection

A remote sensing application case where the 3D information is a valuable resource is the change detection. However, the acquisition of real data for this task is difficult, as taking images from a scene at two (or more) different times imposes significant challenges that make it hard to define what change is, such as seasonal changes, construction works, moving objects, illumination conditions, atmospheric conditions, point of view, etc. As this dissertation later discusses the generation of synthetic data for change detection, we describe here some existing datasets that have been designed/applied for this task.

One of the widely used datasets for semantic segmentation and building extraction is the ISPRS Potsdam dataset[i]. It includes aerial orthoimages over the region of Potsdam (Germany) with a GSD of ~ 5cm. Apart from the imagery, DSMs generated via dense matching are provided with the same GSD. 24 tiles covering each an urban area of 300m × 300m are included. The dataset is taken from a real area with remote sensing sensors and subsequently it has been used as a benchmark, where the metrics to evaluate are precision, recall and the F1-score.

[i] https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-potsdam.aspx

Similar to Potsdam is the ISPRS Vaihingen dataset[ii] which is also a benchmark for semantic segmentation. This covers a smaller area over the city of Vaihingen (Germany) with a GSD of 9cm. for both orthoimages and DSMs. Unlike Potsdam, this datset does not include the blue band but green, red and infrared. The availability of bands is useful for vegetation detection but affects many pipelines based on RGB imagery, like those for matching.

An alternative source is the LEVIR-CD dataset [79] which obtained images from Google Earth with 0.5m GSD and a resolution of 1024×1024 pixels. The images were acquired between 2002 and 2018 and correspond to USA cities such as Austin or Lakeway. The images were taken in different seasons to add this factor as an additional challenge to evaluate algorithms. Nonetheless, the ground truth was manually annotated as this can not be retrieved directly and requires expert knowledge to differentiate where the changes happened. The full dataset contains 31333 annotated individual change buildings. Suburban areas, crop fields, parks and warehouses are examples of the diverse content of the scenes.

The S2Looking dataset [80] focuses on side looking satellite images which were collected between 2017 and 2020 with the GaoFEN, SuperView and Beijing-2 satellites A total of 5000 registered image pairs are available with a resolution of 1024×1024 pixels and a GSD of $0.5 \sim 0.8$m. The number of present building changes is 64920, making it one of the largest datasets. By including rural areas and side view images, the dataset represents a challenge to new developed algorithms.

Another dataset is proposed in [81], named DSIFN-CD with 3940 pairs of images. This is a complex dataset collected from Google Earth and taken over distinct cities, especially in China. 394 original images are augmented to obtain the total 3940 ones with a size of 512×512 pixels. The authors randomly selected 90% of the samples for training.

A different case is the GVLM dataset [82] which stands for global VHR landslide mapping and includes images from 17 landslides around the world, with a surface of 163.77km^2. Such landslides were caused by various factors such as earthquakes, rainfall or glacier melting. Considering the illumination conditions and the effects caused by landslides, this is a very challenging dataset. The labels were manually annotated by experts.

A different strategy to deal with the difficulties of acquiring real data is the usage of synthetic data. The Synthinel1 dataset [83] starts from 3D models created in the CityEngine suite, a software that creates virtual cities by following the Computer-generating Architecture (CGA) programming language. Cameras with a nadir and oblique view are simulated to create optical imagery. Since the content of the scene is known by the software, masks for building/no-building are also accurately generated. The authors used the dataset to augment the training data in the conducted experiments showing an improvement in the performance. However, the ground truth is limited only to two categories and no 3D additional information is provided.

The ParallelEye dataset [84] includes a pipeline using OpenStreetMap, CityEngine and Unity3D to get the map, generate the buildings and rendering respectively. With this procedure, it is easy to generate a large dataset resembling the street distribution from real places, although the buildings do not match with the real ones. Nonetheless, this dataset is not oriented to tasks

[ii] https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-vaihingen.aspx

from aerial/satellite perspective but for autonomous driving, rendering images from the driver point of view, similar to KITTI.

The authors of ChangeAnyhwere [85] proposed a distinct idea, where a dataset with no changes is used as input for a diffusion model that learns how to simulate the change and generate new samples where the content has been modified, creating in that way a paired dataset that resembles two acquisition times. The networks learns that a change can be interpreted as an assignation to a new semantic category, but that the nong-change might present small differences that do not mean a new semantic label has to be assigned, as in a more realistic scenario, where a building might have a different color but is still a building. With this approach the authors created a dataset of about 100k samples. The original samples were extracted from the OpenEarthMap dataset [86], where 5000 images are included from aerial and satellite sources with 8 different semantic classes. As the original dataset covers 97 regions across all continents, it is a robust input to avoid a large domain gap.

Additional datasets and experiments for change detection algorithms can be found in some review papers. A collection for satellite data is described in [87] which provides a large overview of existing datasets classified according to the application task. It also includes some relevant solutions to address each of these tasks. Besides, a description of current satellite mission with their sensor capabilities is supplied.

The MLCNet architecture [88] which is designed for semantic ground truth detection applies the Levir-CD, S2Looking and the later described SMARS [89] datasets in its evaluations. The method has modules to keep edge details and achieves a good performance in all datasets.

3

SyntCities: A Large Synthetic Remote Sensing Dataset for Disparity Estimation

Contents

In this chapter, the content related to the journal paper [90] is addressed. A pipeline to generate synthetic data for stereo matching was designed to create a large dataset resembling remote sensing aerial imagery. Experiments to show the benefit of using this dataset for deep learning approaches are conducted, and a comparison between deep learning and conventional algorithms for stereo matching is also discussed.

3.1 Background

Disparity estimation algorithms aim to find the correspondence between two rectified images and retrieve the shift for the pixels location along the epipolar line. From this shift, it is possible to compute depth values for the captured objects and reconstruct the 3D scene. Generally, the pipeline for conventional algorithms include: matching cost computation, cost aggregation, disparity estimation and disparity refinement [8].

3D reconstruction is also relevant in the remote sensing community, where the input images are processed to generate data such as Digital Surface Models (DSM). Seasonal changes, atmospheric and illumination conditions, urban redevelopment, among others, modify the appearance and content of the captured scenes, making the matching a challenging task. Additional difficulties for a successful matching are imposed by the presence of texture-less, patterned and non-Lambertian surfaces. Moreover, a large range of disparity values might be required depending on the height profile of the scene, which can include mountains or tall buildings.

Conventional approaches like Semi Global Matching (SGM) [4] perform well to estimate disparities for many scenes, but recent deep learning algorithms are now the state of the art, outperforming in complicated areas [11, 17].

Nevertheless, the improved performance offered by deep learning algorithms demands a large amount of samples for training, which is sometimes limited or incomplete in remote sensing. Due to its nature, aerial/satellite-borne data is expensive and its acquisition requires planning to avoid bad weather conditions. Also, the ground truth for disparity estimation is usually obtained from LiDAR, that produces a sparse result and makes it difficult to define sharp boundaries or detect small objects. Additionally, LiDAR shows different behaviour in vegetated areas, especially trees, and needs to be captured simultaneously to avoid systematic differences due to scene changes such as vegetation growth and building activities. 4D light fields and plenoptic cameras are also a resource to generate high quality 3D models [91], but this technology cannot be used during aerial and satellite data acquisition.

Because of all these difficulties to collect large amounts of real data, we propose a new synthetic dataset for disparity estimation. Since the rendering is obtained via software, dense ground truths with sharp boundaries and sub-pixel accuracy are generated. Additionally, we simulate different illumination conditions, ground sample distances and baselines for the stereo system. One of the novelties of the proposed dataset is its remote sensing oriented application by using models that resemble urban areas to reduce the domain gap.

We train different state of the art networks on our generated samples and test the models on real satellite and airborne data. Besides, we compare the results by training with the widely used SceneFlow [12] datasets, where the disparity maps are oriented on close-range applications.

Our main contributions in this paper are the following:

- We present SyntCities, the first (to the best of our knowledge) large synthetic dataset to train disparity estimation focused on remote sensing imagery. Ground truth maps are dense and offer sub-pixel accuracy.
- We conducted a set of experiments on recent neural networks to analyse the advantage of performing data augmentation with our generated samples.
- By comparing with other datasets, we reduce the estimation error and improve the 1-pixel accuracy, which is of crucial importance for the generation of DSMs.
- We show how SyntCities has good generalization capabilities to be used even on unseen data for inference of disparity maps.
- We share the data in formats that can be further processed (like point for cloud generation) and include multi-class semantic maps.

SyntCities can be downloaded at: `https://tinyurl.com/77e3n6m9`

3.2 Related Work

In this section we discuss firstly the existing work oriented to the generation of synthetic datasets, its applications and limitations. Secondly, we mention some studies related to possible usage of both disparity estimation and semantics segmentation, since we provide these maps in our dataset and might encourage the research community to conduct further experiments in this direction. For our own experiments, we focus only on the disparity estimation part.

3.2.1 Synthetic datasets

Deep learning has helped to outperform many algorithms related to computer vision recently, but it also demands a large amount of data to train models that can generalize for testing on images from different sources. However, such large amount of information is not always available or is expensive to acquire. Therefore, the application of synthetic datasets is an option that can compensate the lack of real data for the training process. In many cases, these datasets are used for pretraining stages and smaller sets of real data are applied to finetune the models and reduce the domain gap.

One of the first available synthetic options was the MPI Sintel Dataset [67], where frames are taken from an open source movie and rendered to evaluate optical flow algorithms. The samples were extended to facilitate other tasks such as semantic segmentation, camera motion and stereo matching. In the same way, the SceneFlow datasets [12] were proposed to train neural networks for optical flow, but increasing the number of samples to 34K (instead of 1K as Sintel). Due to its large size and variety of objects and textures, SceneFlow has been one of

the main references to pre-train networks for different tasks. It includes scenes from a movie, random objects and resembling a car perspective on the streets.

Autonomous driving has also benefited from the synthetic imagery. While real images are part of available datasets, these are limited in size and might lead to the overfitting of the models. The KITTI 2012 [64] and KITTI 2015 [65] datasets include images from cameras with a driver's perspective, where elements like streets, cars, houses or vegetation are part of the scene. They also include a ground truth from a laser scanner, providing accurate values for depth. Additionally, files for odometry or semantics ease their application for other tasks. However, the number of samples (around 400 pairs) limits its implementation for deep learning architectures and the sparse measures from the depth sensor provide an incomplete disparity map. As a feasible solution to balance the amount of required data, SceneFlow can be used to pretrain the models for disparity estimation, while the SYNTHIA dataset [92] is a suitable option for the semantic part. SYNTHIA also focuses on autonomous driving and is similar in terms of content and geometry to the KITTI datasets. In contrast, it consists of more than 13K samples and dense ground truth maps. Another similar approach is the ParallelEye dataset [84] based on a pipeline of the CityEngine and Unity3D software suites. It also includes information for object detection and tracking.

Nonetheless, the alternatives described above are oriented to close range applications, which is not suitable for remote sensing, where large areas are covered and small errors in the disparity estimation lead to significant inaccuracies in the DSMs. The Urban Semantic 3D (US3D) dataset [72] was proposed for the Data Fusion Contest 2019 (referenced as "grss_dfc_2019" for the contest itself, but we keep it as US3D in the current paper) and included a stereo matching track. Although the number of samples enables the training of deep learning architectures, the disparity maps are not complete (with a default value assigned to many pixels) and do not archive sub-pixel accuracy, which imposes a significant error when computing the depth. Additionally, a multi-year difference between image and ground truth LiDAR acquisition causes many inconsistencies due to vegetation, building and infrastructure changes. Using multidate imagery also affects the vegetation measurements, since it has visible seasonal changes in terms of color and density. Despite the fact that training with this data might affect the performance of the networks, testing on such imagery is still one of the few options for real large areas.

Developing synthetic datasets within the remote sensing environment has also been studied, although only few publications deal with it. The WHU dataset [78] is based on real aerial images and then merged on a DSM. After that, images are rendered via software from the generated DSM and it produces a synthetic output in form of disparity maps. Ground truth is obtained as dense maps, but the accuracy of the DSM is constrained by the algorithms of the ContextCapture software.

Under these circumstances, we have developed a new synthetic dataset. Considering that the 3D software has detailed information of the geometric content of the scenes, dense and accurate ground truths can be achieved. Furthermore, expanding the dataset for additional views or different simulated conditions can be easily done, reducing costs and time.

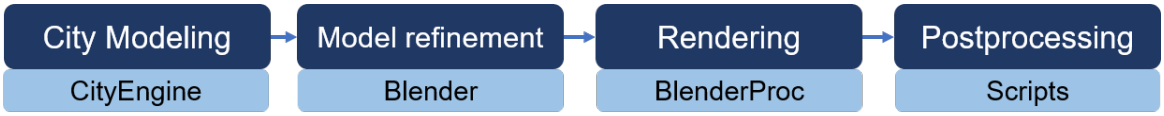


Figure 3.1: Simplified pipeline used for the proposed dataset generation

3.2.2 Approaches to use both disparity and semantic maps

Although the present work focuses on the disparity estimation, the provided semantic maps can be a helpful resource for research making use of both data sources, since these exploit the geometric information from the scene. This idea was recently addressed on the Data Fusion Contest 2019 [93–95], where semantic and disparity maps are predicted and evaluated for the same regions on one of the tracks.

Real datasets for semantic segmentation such as US3D have incomplete semantic maps, with noisy buildings and many elements without an assigned category. On the contrary, synthetic datasets avoid expensive manual annotations and provide sharp dense maps. An existing synthetic example is the Synthinel-1 dataset [83], where models from the CityEngine software are rendered to create segmentation maps with the labels building/no-building. While the pipeline is an efficient way to generate the data and resembles real imagery, the ground truth is limited to two classes and depth information is not included.

Some publications have already studied the usage of both input sources. In SegStereo [96] the semantic information is embedded in the network and also being learned as an intermediate step to refine the disparity map. GIO-Ada [97] learns to reduce the domain gap by creating intermediate samples with a more realistic appearance and later estimates both semantic and depth maps. DispSegNet [98] proposed an architecture similar to SegStereo but using the semantic embedding for the disparity loss and created an enhanced cost volume to improve the accuracy. RTS^{2}Net [99] focused on real-time efficiency and followed a coarse to fine design. SSPCV-Net [100] considered pyramid cost volumes to describe semantics and geometry. CorDA [101] used the depth estimation as an intermediate step to retrieve the disparity maps and with this information reduced the domain gap.

Many of these methods achieve good quality results, but at least for the disparity estimation, they do not compete with the state of the art solutions in terms of accuracy. By releasing this dataset, we intent to facilitate further research in the integration of 3D reconstruction and semantics.

3.3 Dataset Generation and Description

The generation of the dataset makes use of different 3D software suites and scripts for modelling, rendering and postprocessing. In figure 3.1, a simplified description of the adopted pipeline is shown. The detailed steps are explained in the following paragraphs.

3.3.1 City Modelling

CityEngine is a software that allows to build cities in a 3D environment and follows the CGA Shape Grammar Language. Large models can be created from Open Street Map (OSM) and user defined rules for the city architecture and its distribution. In the current paper, we started from the example models for New York, Paris and Venice that are publicly available on the Esri platform.

Empty areas from the examples were replaced with parks and buildings to set content in all the regions of the scene. Vegetation was changed to textured ellipsoid models instead of the intersected planes to have a more natural distribution of depth values. Additionally, we used the script option within the CityEngine environment to separate the buildings according to the rooftop type, this is done to provide the additional semantic maps.

A model including only the buildings belonging to each roof type and a full model including all elements in the scene are exported. All cases were exported in Wavefront (.obj) format. CityEngine consumes approximately 17GB of RAM memory to manipulate the full models, and requires few minutes to export the whole scene.

3.3.2 Model refinement

The models were later imported in the Blender software, which is an open source for 3D modelling, animation and rendering. Here, the objects were split into different categories, which are represented in the ground truth segmentation maps. The objects were created by separating the faces of the complete 3D scene according to the image file used as texture. This does not apply to the buildings, which were previously separated by roof type in CityEngine. A single file in COLLADA (.dae) format is exported with the merging of all objects.

Illumination conditions and camera properties are studied in the 3D environment to set the appropriate values for each city. The light is set to the Sun mode to have a homogeneous brightness in the whole area. A vertex located close to the center of each model is used as a reference to set the camera positions. Apart from that, changes are applied on the reflection properties of the surfaces as well as on the noise distribution for textures. Minor editing was also conducted to avoid empty regions that might lead to the presence of outliers. Furthermore, we set a 3D plane below the models as background, which avoids infinite depth while rendering for not defined regions.The manipulation and edition of the models in Blender requires approximately 8GB of RAM memory.

3.3.3 Rendering

Once the models were complete, we utilized the BlenderProc pipeline [102] to render within the Blender environment. BlenderProc requires a detailed configuration file to set properties such as camera positions, camera parameters, stereo configuration, illumination conditions, output resolution, etc.

Our approach wraps BlenderProc, so we can define externally the main parameters for our dataset. Here we also set the camera positions according to the size of the city model and the desired overlapping between samples. The stereo rig configuration is computed from the base to height ratio and allows different baselines. The configuration file required by BlenderProc is then built with the specified parameters.

Additionally, we manipulated the antialiasing filters to produce smooth borders in the RGB samples but sharp edges for the depth maps. For each camera position we rendered a pair of RGB images, their depth maps and their segmentation maps.

We also experimented the option to produce instance maps (where each building would be assigned a label), but the computational cost is too high even for one camera position. Rendering a pair in instance segmentation mode requires around 200x longer than the semantic case. The rendering process for SyntCities takes a bit more than 5 days using a NVIDIA Quadro P1000 graphics card with 4GB memory and Blender 2.93.

3.3.4 Postprocessing

RGB images and semantic maps were directly obtained from the rendering process. In contrast, the depth map has to be translated into a disparity map. Since the depth is measured in a radial way from the center of the camera, we transformed it into distance to the camera plane first. After that, the distance is used with the known camera parameters to compute the disparity as follows:

$$d = \frac{f \cdot b}{z} \tag{3.1}$$

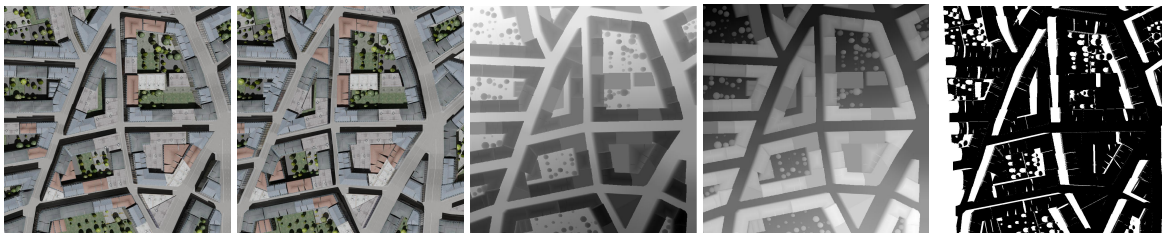
where d is the disparity, f the focal length, b the baseline of the stereo rig and z the distance to the plane. The disparity values are then transformed into pixels. As a result of the different baselines applied to create the dataset, occlusions are present in many samples. Therefore, we also created left-right check consistency maps to mask pixels that are not visible in both views. The threshold for consistency is set to 1 pixel.

Homogenization of categories between different models and rendering conditions is also applied, so the labels remain coherent amid all the samples. For users requiring the camera extrinsic and intrinsic matrices, we also provide these in separate files for each camera position and view. Such matrices are usually expected for multi-view stereo (MVS) neural networks.

3.3.5 Description

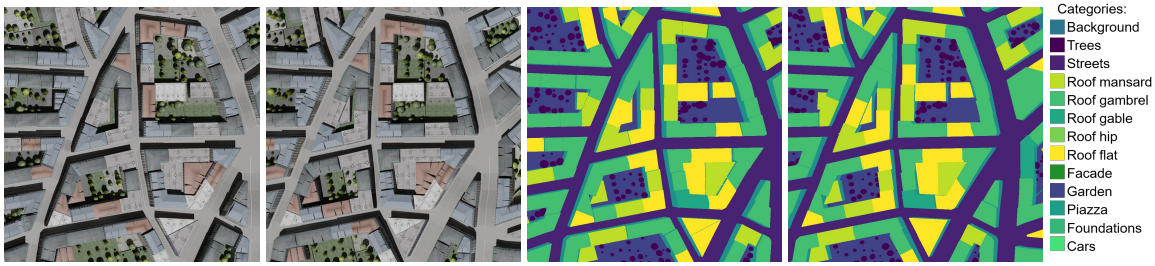
The presented dataset includes a total of 8100 image pairs with the following features:

- 3 city models: New York, Paris, Venice.
- 3 ground sampling distances (GSDs): 10cm, 30cm and 1m.
- 3 azimuth angles (150°, 180°and 210°) and 3 elevation angles (20°, 50°and 70°) for the simulated Sun light.
- 4 base to height ratios (BH) per city: 0.1, 0.3, 0.5 and 0.9 for Paris and Venice; 0.03, 0.07, 0.10 and 0.12 for New York.



(a) RGB - left view **(b)** RGB - right view **(c)** Depth map **(d)** Disparity map **(e)** Consistency mask

Figure 3.2: Samples from the SyntCities dataset. Optical imagery used for input is shown in (a) for the left and (b) for the right view, with the respective depth and disparity maps for the left view in (c) and (d) (Samples for the right view are also available, but not shown in this image). In (e) we illustrate the left-right consistency masks, where the region in white is not visible in both views.



(a) RGB - left view **(b)** RGB - right view **(c)** Segmentation map left **(d)** Segmentation map right

Figure 3.3: Additional samples from SyntCities. Optical imagery used for input is shown in (a) for the left and (b) for the right view, with the respective segmentation maps in (c) and (d). Colors for each category are displayed in the list at the right.

- For each combination of the previous parameters 20 pairs are available for training and 5 for testing. This split is fixed for all cases.
- Disparity values are mainly in the range of [0,192]. This facilitates its direct usage in deep learning frameworks, where the cost volumes usually use such range to estimate the disparities.

On the figure 3.2, we show samples from the dataset for a small region on the simulated Paris model. The 8100 pairs include a similar subset of images, camera parameters and rendering conditions. All images have a resolution of 1024x1024 pixels.

3.3.6 Semantic categories

As mentioned before, semantic maps are also included. There are 13 categories available: vegetation, streets, rooftops (mansard, gambrel, gable, hip and flat styles), facades, gardens, landmarks, cars and background. The figure 3.3 shows an example of the semantic maps for the same patches represented in figure 3.2. Samples for both left and right view are available.

3.3.7 Data for point cloud generation

Taking advantage of the available rendered maps and known camera parameters in SyntCities, we explored the possibility of generating point clouds based on the depth and semantic maps. We utilized the Open3D library [103] for this purpose.

Due to their large file sizes, we do not include these outputs in the dataset, but this can be easily generated from the provided images.

Although we did not conduct any experiments in this direction, we consider this would be helpful for deep learning strategies applied to point clouds, specially because the type of rooftops and other geometries can be learned.

3.4 Disparity Estimation Experiments

We have conducted a series of experiments to analyse the advantages of training architectures on SyntCities for the disparity estimation. Aside from our proposed dataset, we also worked with samples from SceneFlow, US3D and an aerial 4K dataset processed by DLR [104].

SceneFlow is the main reference to train networks for disparity estimation due to its large size, but as we have previously mentioned it is oriented to close-range applications. Hence, we want to compare how networks perform while training with both synthetic options SceneFlow and SyntCities, to investigate if the domain gap with respect to real satellite/aerial imagery is reduced.

On the other hand we also consider two real datasets. First, we take samples from US3D covering areas above Jacksonville, Florida and Omaha, Nebraska. The images are captured by the WorldView3 satellite with 30cm GSD for the panchromatic case. The ground truth is obtained from an aerial LiDAR sensor and almost 4000 pairs are available for training.

Secondly, we use a 4K collection of aerial imagery covering the area of Gilching, Germany with 6.9cm GSD. The reference disparity map for these samples is obtained by an SGM implementation for multi-view stereo matching, where a high-quality DSM is cropped to match the location of the images. Because of the size of this dataset (we consider only 16 images where urban and semiurban areas are covered), we use the samples only to test the algorithms.

3.4.1 Stereo Matching Algorithms

Semi-Global Matching (SGM) has been the main algorithm for stereo matching in the last decades. Its compromise between accuracy and computational cost makes it a feasible option for many applications and is used in open source pipelines for 3D reconstruction like S2P [1]. Unlike deep learning architectures, SGM does not need to be trained on the target domain. Nevertheless, the computation of the aggregated cost requires parameters that are set empirically and have to be adapted to the features of the input images. Those parameters limit the performance of the algorithm and might lead to incomplete disparity maps as outputs.

Deep learning approaches on the other hand require large volumes of data. Even when recent state of the art architectures outperform SGM and traditional methods, the models are not able to handle easily changes in the target domain. For example, a network that has been trained on data for autonomous driving might have a poor performance when applied for remote sensing imagery. Moreover, the training process frequently takes days and a high computational cost in terms of memory and GPU usage.

Despite the drawbacks mentioned above for deep learning, it performs better than traditional algorithms having enough data and a reliable ground truth. Since the publication of MC-CNN [11], where a cost volume is generated with convolutional neural networks, many architectures have achieved outstanding performance for benchmarks like KITTI or Middlebury [8].

Some other remarkable approaches include the first end-to-end architectures Disp-Net [12] and GC-Net [13], where postprocessing steps such as SGM are removed and the refinement of the disparity maps is embedded in the learning process. A significant improvement was later presented with the design of PSMNet [14], which includes a pyramid pooling model to recover more context information and makes use of 3D convolutions to regularize the cost model, a strategy used in many further architectures. Based on a similar principle to SGM, GANet [17] evaluates the costs along different directions to refine the cost volume and avoid discontinuities. To reduce the domain gap presented in the previous networks, DSMNet [20] applies a domain-invariant normalization which benefits of the synthetic imagery. Nevertheless, its performance is not as good as GANet when using the same training dataset. A different concept is presented in AANet [19] to reduce both memory consumption and inference times, while slightly decreasing the accuracy. More recently, strategies consisting of gated recurrent units (GRUs) have been introduced to computer vision tasks with an outstanding performance. This has been applied to the disparity estimation problem, where RAFT-Stereo [21] includes a series of GRUs to estimate maps at full resolution and with high accuracy. In a different strategy, SMAR-Net [105] includes a GAN to compensate for sparse ground truths by warping the left image with the disparity map.

For this paper, we train our models in two networks: GANet and AANet. The reason to select these networks is the accuracy for GANet and the reduced computational cost of AANet, being both also a common framework to compare other architectures.

GANet includes two types of novel layers named Semi-Global Guided Aggregation (SGA) and Local Guided Aggregation (LGA). SGA is based on a principle similar to SGM by considering four directions for the cost aggregation step and LGA recovers information from thin structures. The parameters that are empirically set in SGM are adapted in the model to be learned while training. GANet outperformed the PSMNet (which had the best result for KITTI back then) and generates accurate results on subpixel level. However, the training process might take many days and is computationally demanding.

To reduce the memory and time consumption we conduct experiments with AANet as well. AANet introduces two adaptive aggregation approaches in an intra- and cross-scale manner. The intra-scale aggregation is similar to deformable convolution [106, 107] and adds an offset to the convolutional filters to improve the quality of the result around boundaries and thin structures. The cross-scale aggregation shares information between different scales. Its based

on the idea that correspondences in the coarsest scale are more discriminative in textureless regions and this can guide the algorithm in the finer scales.

3.4.2 GANet experiments

We trained the GANet network with different samples and tested on real aerial and satellite data. The configurations for training are listed in table 3.1. For each training, we show the percentage of each available dataset that was used as input. From this point on, we use SF and SC as acronyms for SceneFlow and SyntCities respectively, specially to describe the experiments and results based on this data.

Table 3.1: Composition of the input data for the proposed experiments with GANet. The GA-SCd case corresponds to the "deeper" version in the GANet paper. Values are expressed as percentages.

	Datasets		
Training model	**SF**	**SC**	**US3D**
GA-SF	100	0	0
GA-SC	0	100	0
GA-SCd	0	100	0
GA-US3D	0	0	100
GA-95SC	0	95	5

The SceneFlow model was trained only for 10 epochs due to its very large size (more than 35K pairs are included) and took more than 6 days. For the other cases we trained for 27 epochs, resulting on 2 days of training time and 4 days in the GA-SCd case. GA-SCd corresponds to the "GANet deep" model presented by the authors in the original paper and includes more layers than the basic model. Here, 6480 image pairs are taken as input, corresponding to all the training samples (80% out of the 8100 available). For the GA-95SC instance, we want to observe the performance of the training when a real but small dataset is available and we can mix the samples with the synthetic ones to compensate the lack of data. We used 4750 samples from SyntCities and 250 from US3D. The GA-US3D model had 4000 samples for training.

Training was conducted on 4 GeForceRTX 2080 GPUs with 12GB memory each, a batch size of 4, patches with 432x432 pixels size, a disparity range of $[0, 192]$ and the other parameters have the default values of the GANet implementation.

3.4.3 AANet experiments

Similarly, we trained AANet with different configurations. Because of the reduced memory consumption and faster training, we conducted an extensive set of experiments. The table 3.2 shows the configurations for the different training models, following the same description system as explained for table 3.1. The AA-SF model was trained for 64 epochs as suggested in the AANet paper. For the other models we adapted accordingly the number of epochs to have a similar training time (around 48 hours each). AA-SF is trained again with more than 35K pairs, AA-SC is trained with 6480 pairs for 350 epochs, AA-US3D with 4000 pairs for 560 epochs and the other models with 5000 samples for 450 epochs. Many cases with mixed sources

Table 3.2: Composition of the input data for the proposed experiments with AANet. Values are expressed as percentages.

	Datasets		
Training model	**SF**	**SC**	**US3D**
AA-SF	100	0	0
AA-SC	0	100	0
AA-US3D	0	0	100
AA-80SF	80	0	20
AA-80SC	0	80	20
AA-95SF	95	0	5
AA-95SC	0	95	5
AA-99SF	99	0	1
AA-99SC	0	99	1

are trained to observe the advantages of data augmentation from synthetic imagery. For all these options we used both SceneFlow and SyntCities. Again, we trained on 4 GeForceRTX 2080 GPUs with 12 GB memory each, a batch size of 24, patches with 288x576 patch size and a disparity range of $[0, 192]$. Other parameters are kept with the default values.

3.5 Disparity Estimation Results

The trained models were tested on the US3D and the aerial 4K datasets. We evaluated four metrics to assess the quality of the results. For the statistical metrics, we use the median-based values instead of the mean-based ones because of their robustness to outliers and their capabilities to summarize skew distributions better [108]. First, we compute the median of the difference between the ground truth and the generated disparity maps. For this metric we did not consider the median of the absolute difference to use it as an indicator of a possible bias. This is computed as:

$$\text{Median}_{\text{diff}} = \text{median}(X_{\text{diff}}), \quad X_{\text{diff}} = X - \bar{X} \tag{3.2}$$

where X is the ground truth, $\bar{X}$ is the generated result and X_{diff} is the difference between both. Second we compute the median absolute deviation (MAD) of the difference as:

$$\text{MAD}_{\text{diff}} = \text{median}(\mid X_{\text{diff}} - \widetilde{X}_{\text{diff}} \mid), \quad \widetilde{X}_{\text{diff}} = \text{median}(X_{\text{diff}}) \tag{3.3}$$

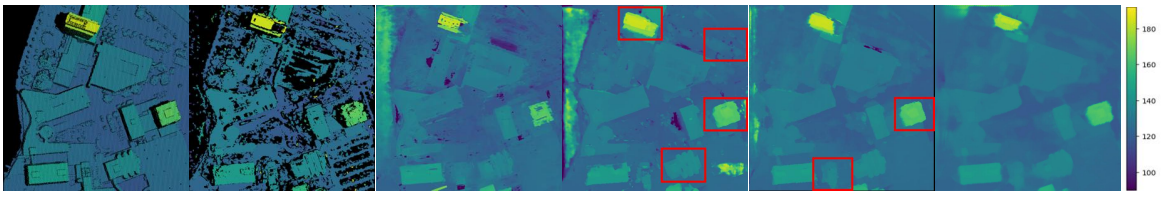
The absolute value is used in this occasion to analyze the precision of the disparity values. We also consider the 3 pixel accuracy, where the percentage of pixels whose difference with respect to the ground truth is below or equal to 3. Likewise, we estimate the 1 pixel accuracy. While a margin of 3 pixels for errors in the disparity map is acceptable for applications like autonomous driving, it would represent a large error when the depth is estimated from the aerial/satellite camera. For that reason, we also consider pertinent to analyse how this metric performs.

Table 3.3: Results of GANet for the US3D dataset. $\text{Median}_{\text{diff}}$ and MAD_{diff} are defined in terms of pixels, while the accuracy is expressed as the percentage of all the pixels below the specified error threshold. Best result is indicated in bold font and underlined, second best just underlined.

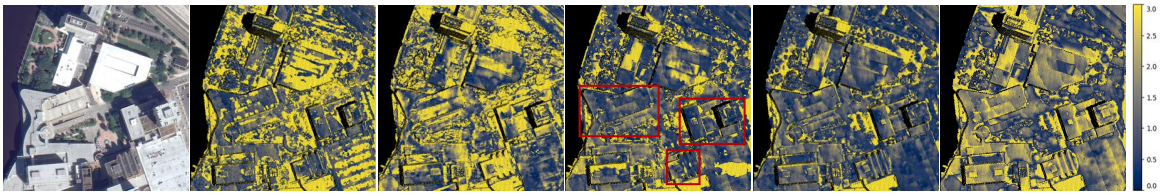
	Algorithms					
Metrics	**SGM**	**GA-SF**	**GA-SC**	**GA-SCd**	**GA-US3D**	**GA-95SC**
$\text{Median}_{\text{diff}}$	1.40	0.94	0.33	0.28	0.61	**-0.05**
MAD_{diff}	3.27	1.92	1.45	1.27	1.10	**0.98**
3pix-acc(%)	57.0	62.3	69.3	72.3	78.3	**79.7**
1pix-acc(%)	32.3	28.9	36.9	38.7	36.3	**43.8**

Table 3.4: Results of GANet for the 4K aerial dataset. $\text{Median}_{\text{diff}}$ and MAD_{diff} are in defined terms of pixels, while the accuracy is expressed as the percentage of all the pixels below the specified error threshold. Best result is indicated in bold font and underlined, second best just underlined.

	Algorithms				
Metrics	**SGM**	**GA-SF**	**GA-SC**	**GA-SCd**	**GA-US3D**
$\text{Median}_{\text{diff}}$	-0.02	**-0.01**	-0.13	-0.12	0.56
MAD_{diff}	0.29	0.33	0.31	**0.28**	0.60
3pix-acc(%)	86.3	92.2	94.1	**94.3**	84.1
1pix-acc(%)	80.4	80.5	83.2	**84.0**	55.1



(a) Disp. reference **(b)** Disp. SGM **(c)** Disp. GA-SF **(d)** Disp. GA-SC **(e)** Disp. GA-95SC **(f)** Disp. GA-US3D

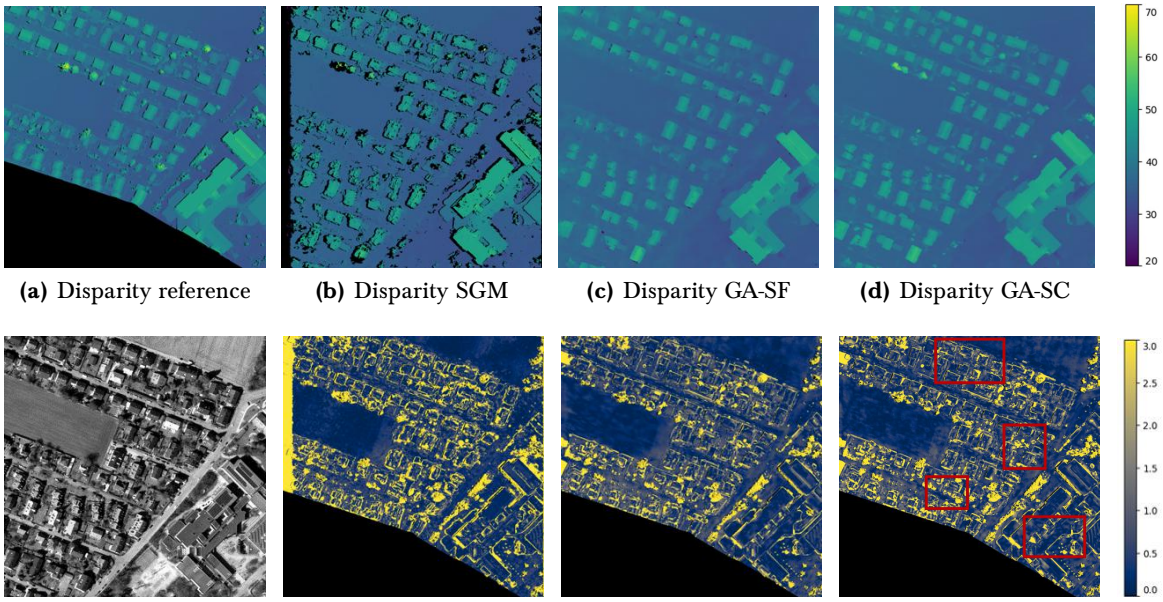


(g) RGB - Left view **(h)** 3pix err. SGM **(i)** 3pix err. GA-SF **(j)** 3pix err. GA-SC **(k)** 3pix errGA95SC **(l)** 3pix err.GAUS3D

Figure 3.4: Results from the GANet for the US3D dataset. The disparity reference (a) is compared to the disparity maps obtained by SGM (b) and the models GA-SF (c), GA-SC (d), GA-95SC (e) and GA-US3D (f). The range for the disparities is set from 90 to 192. Error maps for the reference RGB image (g) are shown for the same models SGM (h), GA-SF (i), GA-SC (j), GA-95SC (k) and GA-US3D (l). The error range is clipped to 0-3 pixels.

3.5.1 GANet results

In table 3.3 we observe the results of using the GA-SF, GA-SC, GA-SCd, GA-US3D and GA-95SC models for the US3D dataset. Additionally, we also compared the results with the traditional SGM algorithm. It is important to mention that SGM does not produce a complete result, but has values only for those pixels where the estimation achieves the quality accepted by the algorithm. However, we evaluate the metrics in the whole image since completeness is a desired



(a) Disparity reference **(b)** Disparity SGM **(c)** Disparity GA-SF **(d)** Disparity GA-SC

(e) Optical - Left view **(f)** 3pix error SGM **(g)** 3pix error GA-SF **(h)** 3pix error GA-SC

Figure 3.5: Results from the GANet for the 4K aerial dataset. The disparity reference (a) is compared to the disparity maps obtained by SGM (b) and the models GA-SF (c) and GA-SC (d). The range for the disparities is set from 20 to 70. Error maps for the reference optical image (e) are shown for the same models SGM (f), GA-SF (g) and GA-SC (h). The error range is clipped to 0-3 pixels.

feature as well.

Considering the 3 pixel accuracy, we can observe that all the trained models outperform SGM by a significant margin. If we compare only GA-SF and GA-SC we notice already an improvement of 7% despite the shorter time that was used to train on the SyntCities dataset. GA-SCd has even more accurate results, but it also required a larger training and might not be a suitable option if the computational resources are limited. The model GA-US3D is even better by 6%, which is also expected since the domain gap does not play a role for this case. Interestingly, the GA-95SC model is the one that performed best, although it does not rely only on samples from the US3D dataset. While the improvement for the 3 pixel accuracy metric is slightly higher, the case for 1 pixel accuracy increases more than 7%. By comparing the results on the GA-95SC model and GA-US3D, the former had issues to estimate some areas, but produced sharper results than the latter. The training process augmented with the synthetic data seems to benefit from the accurate ground truth available on SyntCities. It is also important to remark that this strategy could work for datasets with reduced volume as well.

Focusing now on the 1 pixel accuracy, SGM has actually a better result than GA-SF but worse than GA-SC. In this way we can notice how SC boosts accuracy to a finer detail. As mentioned before, this metric has special attention from the remote sensing community for a correct 3D reconstruction. The values for $\text{Median}_{\text{diff}}$ and MAD_{diff} follow a similar trend to the accuracy.

Images to show the performance of the algorithms are presented in figure 3.4. The first row illustrates the disparity maps obtained and the respective reference. The second row shows the error maps, where all values ≥ 3 are in yellow. We can observe how completeness is obtained

by all deep learning algorithms, which is not the case for SGM. However, the valid values obtained by SGM show good accuracy. Now, if we compare only the GA-SC and GA-SF cases for the disparity map, we can notice a better estimation for building areas and vegetation on GA-SC as illustrated with the red rectangles. The model GA-95SC is of course the one with the best reconstruction, since it was partially trained on the test domain.

We can also study the performance for the error maps, where the presence of large areas in blue (error ≤ 1 pix) is desired. This is already achieved for many building and street sections on the GA-SC model as shown in the red rectangles. Difficult areas to solve for the model remain mostly for vegetation and vehicles, which in some cases were not present on the right view. In any case, the significant reduction of the error range would lead to a superior quality for DSM generation, crucial for remote sensing.

With regard to the results shown in table 3.4 for the 4K aerial data we have a similar behaviour. All models show a better accuracy for this dataset in comparison to US3D, this might be a result of the quality of the data referenced as a ground truth. Again, the neural networks outperform SGM, also for 1 pixel accuracy in this case. The GA-SCd model has a slightly improvement with respect to the normal GA-SC. We did not compare the GA-95SC model because it would be challenging to evaluate the individual benefit of each of the two sources of the mixed dataset.

Nevertheless, we made inference on the GA-US3D model as this case is trained on real data as well. A 10% decrease in the 3 pixel accuracy of the result shows that training a model only on US3D data can not be used for a different set of images, while the SF and SC datasets have a better generalization to estimate disparities in different domains. Moreover, the accuracy in terms of 1 pixel is lower than any other case, including SGM that is not defined for all the pixels.

Visual results for the experiments on the 4K aerial dataset are displayed in figure 3.5. Similarly to the US3D dataset, we notice more complete buildings and detection of vegetation on the GA-SC model. This is highlighted with the red rectangles. The effect is similar when analysing the error map, where a significant part of the constructions is within 1 error accuracy and a larger number of trees is retrieved.

In all the illustrated cases, vegetation is still a challenging element in part because of seasonal changes, but we also think that a more realistic 3D representation on the synthetic models could improve the performance.

3.5.2 AANet results

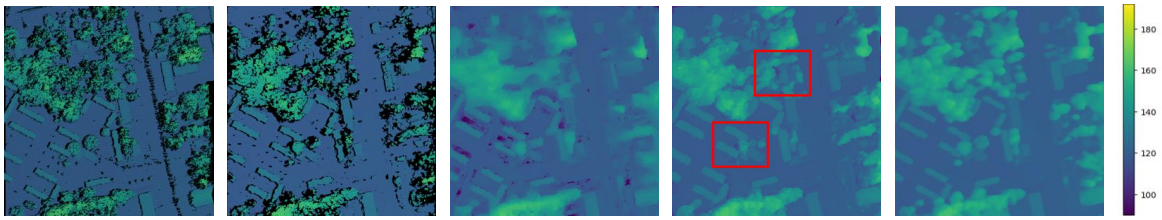
Results from the implementation of the AANet architecture for the US3D dataset are shown in table 3.5. Accordingly to the findings explained for the GANet, the deep learning models also outperform SGM. The highest accuracy is achieved by AA-US3D, which is an expected outcome taking into account that it is trained and tested on images of the same domain. Again, the AA-SC model got a better result than AA-SF and demonstrates the benefits of SyntCities for the training process.

Table 3.5: Results of AANet for the US3D dataset. $\text{Median}_{\text{diff}}$ and MAD_{diff} are defined in terms of pixels, while the accuracy is expressed as the percentage of all the pixels below the specified error threshold. Best result is indicated in bold font and underlined, second best just underlined.

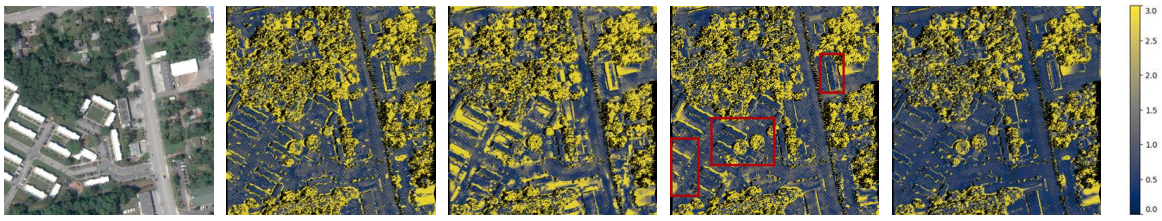
	Algorithms (All but SGM, AA-)									
Metrics	**SGM**	**SF**	**SC**	**US3D**	**80SF**	**80SC**	**95SF**	**95SC**	**99SF**	**99SC**
$\text{Median}_{\text{diff}}$	1.40	0.72	0.08	0.10	0.09	0.10	0.11	**-0.04**	0.09	-0.09
MAD_{diff}	3.27	1.82	1.72	**0.89**	1.08	1.05	1.25	1.09	1.42	1.23
3pix-acc(%)	57.0	63.2	64.3	**85.3**	78.5	79.6	74.8	77.7	70.9	74.7
1pix-acc(%)	32.3	29.3	32.6	**49.8**	41.7	42.4	37.5	41.4	34.6	38.4

Table 3.6: Results of AANet for the 4K aerial dataset. $\text{Median}_{\text{diff}}$ and MAD_{diff} are defined in terms of pixels, while the accuracy is expressed as the percentage of all the pixels below the specified error threshold. Best result is indicated in bold font and underlined, second best just underlined.

	Algorithms			
Metrics	**SGM**	**AA-SF**	**AA-SC**	**AA-US3D**
$\text{Median}_{\text{diff}}$	**-0.02**	-0.07	-0.06	0.29
MAD_{diff}	0.29	0.39	**0.28**	0.50
3pix-acc(%)	86.3	90.5	**92.7**	87.8
1pix-acc(%)	80.4	74.8	**82.6**	66.8



(a) Disparity reference **(b)** Disparity SGM **(c)** Disparity AA-SF **(d)** Disparity AA-SC **(e)** Disparity AAUS3D

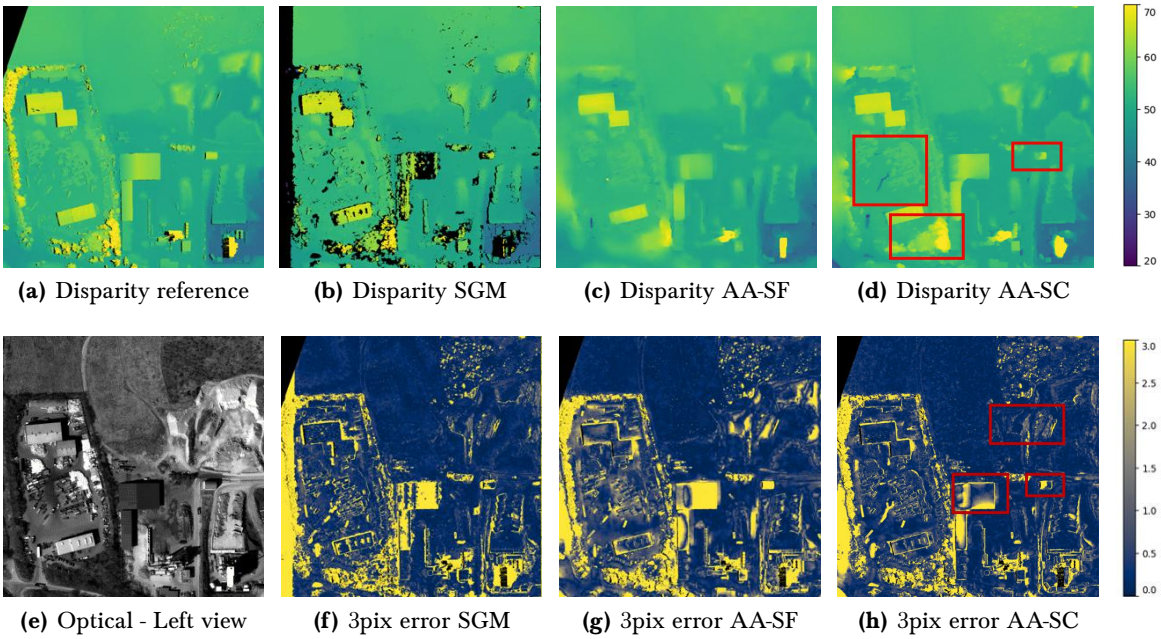


(f) RGB - Left view **(g)** 3pix error SGM **(h)** 3pix error AA-SF **(i)** 3pix error AA-SC **(j)** 3pix error AAUS3D

Figure 3.6: Results from the AANet for the US3D dataset. The disparity reference (a) is compared to the disparity maps obtained by SGM (b) and the models AA-SF (c), AA-SC (d) and AA-US3D (e). The range for the disparities is set from 90 to 192. Error maps for the reference RGB image (f) are shown for the same models SGM (g), AA-SF (h), AA-SC (i) and AA-US3D (j). The error range is clipped to 0-3 pixels.

There are also many cases presented with a mixture from the input data. Models with SceneFlow and SyntCities are compared at different rates of shared data. Nonetheless, the options where SyntCities is involved perform better than those with SceneFlow. This can be noted in both 3 and 1 pixel accuracy. Due to image size limitations not all the cases are illustrated.

Once more we appreciate the advantages of mixing the data with real samples. US3D has enough samples to be trained on its own imagery, but this might not be the case for other



(a) Disparity reference **(b)** Disparity SGM **(c)** Disparity AA-SF **(d)** Disparity AA-SC

(e) Optical - Left view **(f)** 3pix error SGM **(g)** 3pix error AA-SF **(h)** 3pix error AA-SC

Figure 3.7: Results from the AANet for the 4K aerial dataset. The disparity reference (a) is compared to the disparity maps obtained by SGM (b) and the models AA-SF (d) and AA-SC (d). The range for the disparities is set from 20 to 70. Error maps for the reference optical image (e) are shown for the same models SGM (f), AA-SF (g) and AA-SC (h). The error range is clipped to 0-3 pixels.

small datasets. Even by adding only 1% of real data to the training process we can reduce the domain gap as exhibited in the last two columns of the table (comparing only with AA-SF and AA-SC).

Visual results related to these experiments are shown in figure 3.6. AA-SC generates sharper buildings and finer forest sections as remarked in the red rectangles. The range for disparities in the ground level is also more consistent with less generated discontinuities. Similar conclusions can be derived from the error maps displayed in the second row where values for buildings and streets are more uniform on the AA-SC model. This is a congruous result for as much as the 3D models were largely defined for these regions. Although there is room for improvement on the simulated urban scenes, the current quality of the synthetic samples suggests that its usage for training and pre-training is a feasible strategy. The vegetation is still a difficult area to address even for the AA-US3D model.

Turning to the results of the 4K aerial view dataset shown in table 3.6, the AA-SC model performs the best for both 1 and 3 pixels accuracy. An interesting point is the 1 pixel accuracy of SGM, which surpasses the one from AA-SF. This has also been observed in tables 3.3, 3.4. It seems that SyntCities raises the subpixel accuracy.

Images related to this experiment are on display in figure 3.7. In the selected sample vehicles are also present (see the largest red rectangle on the disparity maps) and finely estimated with the AA-SC model, where sharper boundaries are visible. AA-SC also has an improved representation for vegetation areas. The constructions have a similar performance to the other training experiments, exhibiting the benefits of the AA-SC models. Similarly to the results

from GANet, the disparity maps generated on a model trained only on US3D data have larger errors than those trained on the synthetic data. Furthermore, the 1 pixel accuracy is again lower than the other compared methods.

An interesting point to mention for both GANet and AANet is the sensitivity to the disparity distribution of the training dataset. From the conducted experiments, we observed that the large range covered by the synthetic datasets adapts easier for inference in unseen data. On the other hand, US3D has a narrower range and this would lead for a lower performance if the images are not preprocessed before inference on this model. We shifted the left image of the 4K aerial samples to obtain a disparity distribution similar to the one of the US3D dataset to have a fair comparison. Without this preprocessing, a large systematic disparity offset has been observed. However, this behaviour could cause worse results for other experiments if the data is directly feed into the networks without previous knowledge of the disparity distributions used in training. This will especially affect hilly or mountainous areas with larger disparity differences.

3.6 Discussion

A reliable digital surface model (DSM) is a valuable resource for applications such as city planning, updating of cadastral data, transport and flight simulation, autonomous driving or prevention and response to natural disasters, among others. Considering that, we presented in the current paper the SyntCities dataset, which is to the best of our knowledge, the first large synthetic dataset for disparity estimation with focus on remote sensing. The generated samples include different illumination conditions and stereo configurations and benefit from the simulation model to generate a dense and accurate ground truth.

Experiments made for the disparity estimation demonstrate that the accuracy is improved by using our proposed dataset in comparison to models trained on the Scene Flow dataset. This was observed for both aerial and satellite data. A significant outcome is the boost for 1 pixel accuracy, which is desired for remote sensing applications where a single pixel might represent a large distance on the ground.

We also observed that our samples can be used as an augmentation strategy to compensate the lack of data in small real sets. Furthermore, models training on SyntCities without finetuning achieved a good performance on unseen data such as the US3D and the 4K aerial samples.

For future work we want to upgrade the quality of the 3D models by including not only urban areas but features from natural landscapes too, a more realistic vegetation representation and an expanded variety of buildings and architecture. We would also like to conduct some experiments to benefit from both disparity and semantic maps, since their information might be complementary. An algorithm able to create a labelled DSM would enhance many spatial databases.

Apart from that, the dataset could be enhanced with additional viewpoints to allow the training of multi-view-stereo algorithms.

4

Synthetic Data Generation for urban semantic segmentation and change detection

Contents

In this section, a part of the journal paper to release the SMARS dataset [89] is described. The full article was a contribution from few colleagues to tackle the change detection, semantic segmentation and building detection task. Yet, the contribution of this dissertation focuses on the dataset generation part, so the data creation pipeline and features of the generated samples are the part from the article to be discussed.

4.1 Background

Large datasets for computer vision tasks have been created to address many tasks and study fields for years. Some known ones are for object class recognition like PASCAL Visual Object Classes (VOC) 150 [109], or KITTI [110] and Cityscapes [111] for autonomous driving. However, deep learning methods require a large variety of samples to avoid problems such as overfitting, domain gap, incorrect annotations and unbalanced data. The number of datasets that can be used then for deep learning is then reduced.

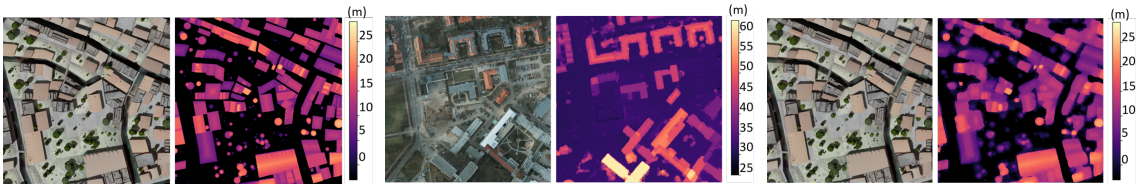
The field of remote sensing and Earth observation has also benefited from deep learning methods for many applications [112, 113]. What is more, it has also been observed that using multimodal data can help to supplement the information from each input sensor, like for the tasks of building detection [114], image segmentation [115] and change detection [116, 117].

Nonetheless, acquiring remote sensing data from many sensors is expensive, and this data is usually not collected simultaneously, leading to differences in illumination conditions, present objects, viewing angles, among others. Apart from that, the resolutions from each sensor vary and the data has to be aligned and/or resampled to be used together as input for processing algorithms. Because of these difficulties and the limited data available for training deep learning approaches, providing a 2D/3D dataset with reliable ground truth is a helpful resource to develop and evaluate newer architectures.

Synthetic datasets have emerged recently to fill this gap and provide a large number of samples for a reduced cost. These have been applied for the medicine field [118], for complex physical models [119, 120] and already remote sensing [78, 84]. However, modelling the complexity of urban and nature environments is a demanding job and the domain gap between the synthetic and real data should be narrow.

Considering the generation of DSMs in a 3D environment, this retrieves highly accurate results as presented in Fig. 4.1 (a), exhibiting sharp edges around the buildings without any occlusions or gaps. Such precise DSM can be hardly achieved using real data with the currently available optical acquisition and stereo matching techniques, as results obtained from photogrammetry pipeline are characterized by blurred boundaries and contain outliers (Fig. 4.1 (b)). In order to reduce the gaps between rendered and real data, we aim at defining a novel approach generating synthetic DSMs with the same limitations of real ones, as for the DSM reported in Fig. 4.1 (c), which more closely resembles the level of detail in Fig. 4.1 (b) with respect to the generation using directly rendered samples.

On the basis of all the above points, we propose a novel synthetic photogrammetric data generation pipeline with a particular emphasis on the use of 2D/3D multimodal urban segmen-



(a) Directly rendered samples **(b)** Real dataset samples-Potsdam **(c)** Proposed synthetic samples

Figure 4.1: Quality differences between synthetic and real data. Elevation scale for the DSM is in meters.

tation, building detection and 3D change detection. The main contributions of the paper that contribute to this dissertation are the following:

- A workflow to produce synthetic data resembling urban areas at different growth stages.
- A 2D/3D multimodal remote sensing dataset, which we name the Simulated Multimodal Aerial Remote Sensing (SMARS).

4.2 State of the art

4.2.1 Existing real 2D/3D multimodal benchmark datasets

Because of the acquisition costs and the aforementioned complications, the number of available 2D/3D multimodal benchmark datasets is limited. The ISPRS Potsdam dataset[i] is a widely used and popular public benchmark for 2D/3D semantic labeling, also applied to test and validate building extraction methods [121]. This dataset includes airborne orthoimages and corresponding DSMs generated via dense image matching. GSD for both images and DSMs is approximately 5cm. The original training set comprises 24 pairs of tiles, each having a size of 6000×6000 pixels (300×300 m).

Another case is the ISPRS Vaihingen[ii] airborne benchmark, which also contains 2D images and DSMs. However, the blue band for RGB images is not available, so it can not be applied for many existing algorithms, but it is useful for vegetation analysis as a near-infrared band is included. DroneDeploy[iii] is a 2D/3D multimodal dataset comprising UAV imagery but provides irregular mosaics and separated training/test subsets. Thus, it is barely applied by the research community.

Regarding the change detection, there are few single modal benchmark datasets available [122–125]. As far as we can tell, 3DCD is presently the only benchmark with 2D/3D multimodal data for remote sensing change detection suitable for deep learning frameworks [126, 127]. Nonetheless, GSD and acquisition times for LiDAR differ from the optical acquisition, potentially affecting their paired use in multimodal algorithms. Aside from undefined pixels in the DSM, changes are for general land use, not only building modifications. Moreover, the dataset focuses on the landscape of Valladolid, Spain, leading to significant domain gaps.

[i] https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-potsdam.aspx
[ii] https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-vaihingen.aspx
[iii] https://github.com/dronedeploy/dd-ml-segmentation-benchmark

4.2.2 Synthetic data in remote sensing

Curating real 2D/3D multimodal datasets is a time consuming task that relies on the costly step of manual annotation. Thus, synthetic data generation is a feasible solution if real imagery is unavailable or hard to collect. Hyperspectral data was simulated in [128] and the authors of [129] replicated SAR for change detection. A different case was presented in [130] to model the geometrical shapes present in the forest canopies, like for conifers. For the creation of multi-temporal datasets there are additional obstacles, such as a complex annotation, having limited public benchmarks. One of the few cases is presented in [131], which is rather small and with low resolution images.

The described constraints in curating the mentioned multi-temporal datasets can be mitigated by reliance on synthetic data. For instance, [132] simulated a set of varying mis-registrations degrees to study their impact on vegetation change detection. Another proposed dataset can be found in [133] but the scene includes a simplified geometry with artificial generated noise.

A real LiDAR point cloud is used in [134] to generate a Level of Detail 2 (LoD2) model as a pre-event dataset. By manually adding and removing buildings in the model, the city growth process can be reproduced. Yet, as only buildings were model in the 3D scene, the results have a large domain gap with real urban 3D models.

To produce more realistic samples [135] proposed an artificial data generation pipeline guided by expert knowledge to control the automatic image and label generation. However, with more complex background information, urban change detection is difficult to simulate and control.

Other studies explore the radiative transfer models as an option to simulate remote sensing data [136, 137]. A remarkable work is the Discrete Anisotroipc Radiative Transfer(DART) model [138], which accurately simulates vegetation properties such as chlorophyll fluorescence. It can also reproduce the vegetation reflectance even for complex canopy geometries [139]. Nevertheless, urban scenarios are not easily modelled with DART because of the complexity of the parameters that have to be added.

On the other hand, software for 3D rendering like Blender, Unity or Unreal Engine offer an interface where the modelling of more complex geometry and retrieval of parameters such as distance to a simulated cameras are easier to process. It is also possible to import 3D models from other suites, render additional images for ground truth and add material properties [140–142]. Besides, a detailed and realistic model for urban elements such as buildings and vegetation is needed, which is feasible with 3D rendering software, where even the simulation of physical processes is possible.

4.2.3 Virtual city synthetic data

Generating data from a virtual model is currently becoming more popular in computer vision due to the capabilities of modeling software and the reduced cost compared to using sensors for real scenes. However, the application of synthetic data is rather limited if the domain gap with the real data is too large.

A virtual model can contain anything from a small object to a large city. For example, building models can be used to create indoor based point clouds [143], or depth and semantics, as in Hypersim [144]. Autonomous driving algorithms have also benefited from the developments of synthetic data creation. A widely known example is the SYNTHIA dataset [92] that provides synthetic images of urban scenes labeled for semantic segmentation. Such scenes are rendered from a virtual New York City 3D model with the Unity game engine. The dataset includes segmentation annotations for 13 classes including pedestrians, cyclists, buildings, and roads. Another approach is used in CARLA [145], an open source simulator that supports the training, prototyping, and validating of autonomous driving models. CARLA facilitates the data acquisition from street view for the generation of segmentation and depth maps. Similarly, the ParallelEye dataset [84] generates images from the CityEngine software with depth and optical flow as part of the ground truth.

A similar setting can be considered for the simulation of aerial or satellite imagery. The Synthinel-1 dataset [83], also based on CityEngine, targets the building/no-building classification from an airplane perspective. The authors also addressed the advantages of synthetic imagery by ablation studies. The VALID dataset [146], focuses on panoptic segmentation and depth estimation for urban infrastructure. Furthermore, the SyntCities dataset [90] provides semantics and disparity maps, making the data suitable for stereo reconstruction. The STPLS3D dataset [147] provides point clouds, and semantic and instance maps built on open geospatial data sources. Authors in [148] simulated LiDAR acquisition for an urban environment and delivered the dataset as point clouds.

However, further applications of synthetic data are limited by the large differences with respect to the real testing data. A remarkable example is the much higher quality of the DSM obtained from the virtual 3D models in comparison with the one generated from photogrammetric matching. Edges are usually sharper in the simulated data, and the occlusions are absent in the generated ground truth. In addition, images from real scenarios show imperfect textures, light reflection, seasonal changes, the presence of temporary objects (cars, pedestrians, street advertisements, etc.), atmospheric effects, and other elements that cannot be easily modeled in software. Hence, the simulation is mostly limited to the geometry of the scene, textures, and camera properties. Still, the rendered images can visually resemble real cases and help to compensate for the limits of real sensors (such as sparsity) and reduce the costs to generate ground truth.

4.3 Methodology on synthetic data generation

To close the gap between synthetic data collection and remote sensing applications we combine two techniques, airborne data collection from virtual cities and photogrammetric stereo data preparation. In this section, we propose a novel workflow to generate a 2D-3D multimodal dataset. A diagram to summarize it is shown in Fig. 4.2. It consists of three parts: 3D virtual city design, imagery simulation, and data processing.

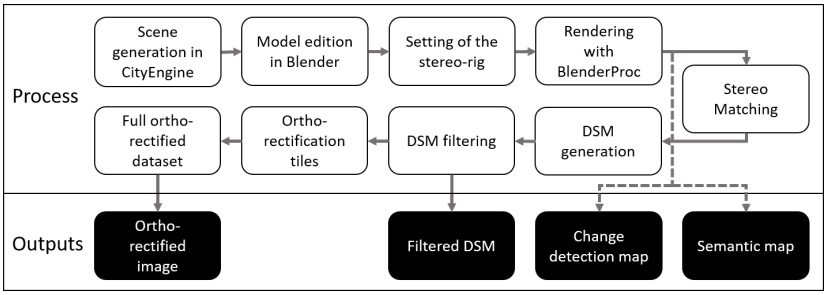


Figure 4.2: Basic description of the pipeline used to generate the SMARS dataset.

4.3.1 3D virtual city design

In order to produce a realistic change scenario we used a 3D virtual city as a starting point to simulate the scene growth process, instead of directly generating artificial images. We built the 3D scenes based on the CityEngine software[iv], a suite facilitating the modelling of urban environments based on the computer-generated Architecture (CGA) shape grammar language. The software was also used to develop the above-mentioned ParallelEye and Synthinel-1 datasets [83, 84]. CityEngine supports building a city model from land cover maps, such as Open Street Map, or a manually designed base map. However, designing a virtual world with carefully customized features would require relevant expert knowledge and would be time-consuming. Therefore, we selected two predefined city models from ESRI and further refined them accordingly.

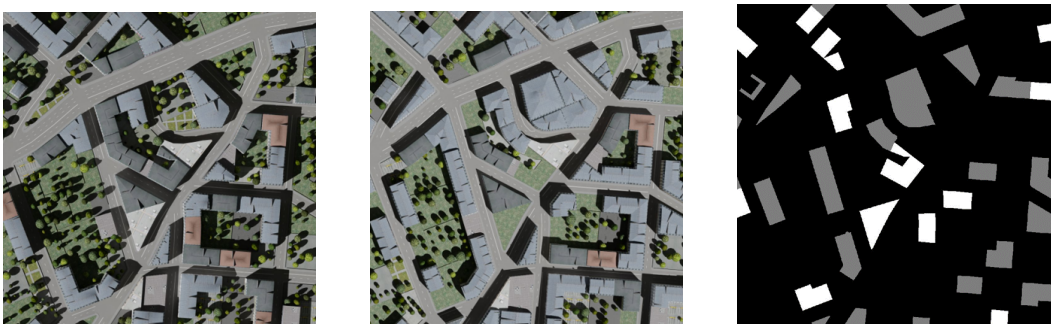
In this paper, we chose two typical European cities: Paris and Venice. Subsequently we refer to them as SParis and SVenice, respectively. The selected city models have a variety of textures and architectures resembling the original cities, as well as a large surface allowing the inclusion of many buildings in the subsequent rendered images. The buildings are defined in terms of roof type, roof angle (if any), height, number of floors, floor height, and size of the parcel. In order to have a lifelike view, we further edited the 3D model of the cities by modifying the streets in order to have a more realistic topography, as the original version had streets with the shape of letters. The trees were replaced with textured ellipsoids instead of the original ones represented with a uniform color. Additionally, some areas were manually corrected in order to ensure that any parcel in the area included urban content.

A large pool of 219 textures has been used in the provided models for buildings (rooftops and facades) and 87 for vegetation. For the latter ones, we edited the default textures of the ellipsoids by creating a dense representation of leaves in order to resemble canopies. While still limited in terms of the full diversity of the real world, these refinements helped to create a scene with sufficient variability.

As the dataset is mainly intended for change detection applications in urban areas, each city model was generated with two versions, simulating the city's growth:

- A case where around 50% of the parcels are covered by buildings. This is seen as the model before changes happen and we call it pre-model in the remainder of the paper.

[iv] https://www.esri.com/en-us/arcgis/products/esri-cityengine/overview



(a) Pre-model **(b)** Post-model **(c)** Change ground truth

Figure 4.3: Rendered samples from the pre- and post-models with associated ground truth for change detection. The pre-model has lower building density and different illumination conditions. Black regions in the ground truth exhibit no change, gray indicates new buildings and white removed buildings.

- A case with approximately 70% of the parcels covered by buildings. Some areas defined previously as gardens are replaced by constructions, while some buildings have been removed and substituted with green areas. This model contains the changes to be detected, and is therefore named post-model.

In Fig. 4.3, we show samples for both the pre- and post-model, respectively 4.3a and 4.3b. The central image exhibits a higher number of buildings and less vegetation cover. Also, some of the original buildings have been replaced with lawns or vegetation.

According to the requirements described above, we adapted a total of four city models (two cities, two epochs) and exported all cases in Wavefront (with extension .obj) format for further editing. The edition of the scenes in CityEngine demands about 17GB of RAM memory.

Subsequently, we loaded the Wavefront files in Blender, an open source tool for modeling, simulation, and rendering. We applied the BlenderProc pipeline [102] to render the images. Our rendering approach is based on the one described in SyntCities [90] and we created for this case the colored images (we refer hereafter to them as "optical") and the semantic maps.

Within Blender we split the geometry of the scenes according to their textures, separating all the surfaces into the required semantic labels. The available categories include: vegetation, streets, rooftops (mansard, gambrel, gable, hip and flat styles), facades, grass, landmarks, cars, and background. We combined them into five typical land cover classes used for urban mapping, including buildings (all rooftops, facades and landmarks), streets, high vegetation (trees), grass (lawns) and others (cars, water, bare soil or background).

We simulate different illumination conditions by setting an artificial Sun in two specific positions for the pre-/after-event models, reproducing two different times for data acquisition. The selected angles were 70°for elevation, and 217°(pre-model) and 160°(post-model) for azimuth. The same conditions were applied to both cities. Finally, we added a homogeneous plane under the ground level of each scene to avoid undefined regions (no value pixels) in the rendering process, which is assigned to the "other" category. Without it, distance would be considered to be infinite if there is an empty region in the objects. This plane guarantees a color and depth value for each rendered pixel.

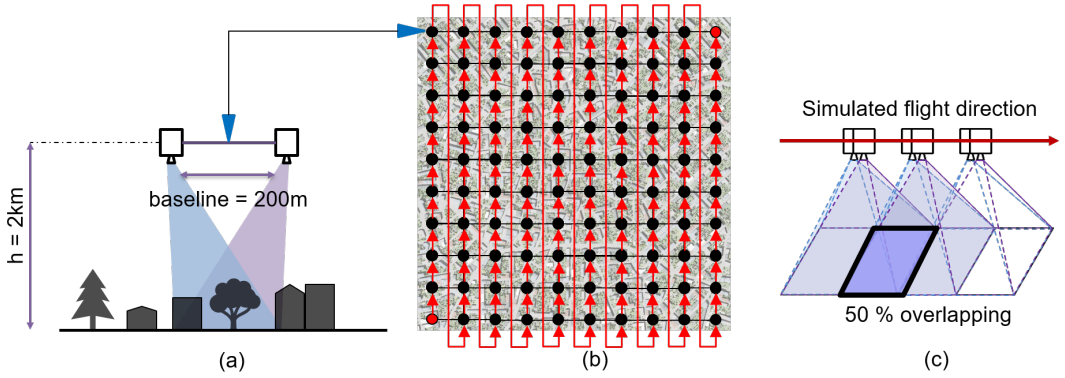


Figure 4.4: Simulated stereo configuration. (a) Stereo rig, where the converge distance and baseline have been adapted to cover the same area on the ground. (b) The path of the simulated camera above the scene. (c) Overlapping between adjacent samples is 50% for both horizontal and vertical directions.

4.3.2 Airborne stereo imagery simulation

SMARS is designed to resemble aerial imagery and the simulated camera is constrained by a stereo rig, which helps to later generate a digital surface model (DSM). In this part, we provide more details on the simulated data acquisition and camera parameters.

Firstly, the simulated camera is located 2km above the origin of the scenes. Since we used synthetic models that are not georeferenced, the origin of the coordinate system assigned by City Engine is used by default. An arbitrary point located at the center of the model and on the terrain level is taken as a reference for the rendering process.

In Fig. 4.4(a), we show the configuration of the stereo rig. In order to simulate the stereo imagery acquisition procedure, two cameras are located at the same distance from the rig center with a baseline of 200m in all cases. Both cameras follow the pinhole model and have the same focal length. As image resolution plays an essential role in transfer learning, we aim to provide this image dataset in two GSDs, namely 30cm and 50cm. Following Eq. 4.1, we set the focal length of the cameras to 234.37mm and 140.62mm, respectively.

$$f = \frac{height \cdot sensor\_width}{covered\_area} \tag{4.1}$$

where f is the focal length, $height$ = 2000m as described above, $sensor\_width$ = 36mm for the simulated camera and $covered\_area = 1024 * \text{GSD}$, being 1024 the size in pixels of the output image. The converge distance is set to 2km (same as the height) with an off-axis camera, which allows us to cover the same area on the ground from two different points of view. This configuration is modeled with the offset of the principal point in the camera intrinsic matrix.

In Fig. 4.4(b) we illustrate the trajectory of the simulated camera above the scene. We rendered images at 100 positions within a regular square grid, with strides set as 153.6m and 256m for 30cm and 50cm GSD, respectively. The center of the grid is set to be close to the one of the scenes, so most of the content is included. In order to simulate a real-world airborne data acquisition campaign, the pair of stereo-cameras are moved from the lower-left to the upper-right corner with a constant stride. The points belonging to the grid represent the location of the

center of the stereo rig (see the arrow with blue extremes). This means that the cameras are located symmetrically to the left and right side of each point.

Overlapping between adjacent samples is set to 50% in both the horizontal and vertical directions of the grid. A visual representation of the overlapping is given in Fig. 4.4(c), where the camera pairs along the simulated flight direction are also included. The images are rendered with a size of 1024×1024 pixels.

After rendering, a semantic segmentation map to be used as ground truth (GT) is delivered with the categories described previously (buildings, streets, vegetation, lawns and others). For the building extraction GT map, we combine all categories except building to no-building, enabling binary semantic segmentation. With the pre-/post-event building extraction GT maps, we calculate the building change detection map by taking only the building class for comparison. Three change classes are included:

- No change: buildings or no-buildings have the same semantic label pre/post-event images.
- Construction: pixels labelled as building in the post event images are no-building in the pre-event images.
- Demolition: pixels labelled as building in the pre-event images are replaced by the no-building label.

The change detection ground truth is directly rendered from the 3D model with an orthographic view. Labels for the semantic categories are also directly rendered from Blender, as BlenderProc generates a category for each object in the scene. The building masks are a simplified version of the category maps considering a binary building/non-building case. For the change detection mask, building masks are compared and labelled according to their difference. In this case, all generated ground truth is generated in the rendering step, and therefore perfectly matches the original images. Due to the orthorectification process described in subsection 4.3.4, the alignment will not be perfect as this simulates the quality obtained from a photogrammetric pipeline.

4.3.3 Stereo matching and DSM generation

Although very precise 3D point clouds and DSMs can be directly delivered with the rendering software, the quality of these data for all cases will be higher than the real-world 3D point clouds generated by stereo matching techniques, where many mismatching errors and occlusions occur. Thus, in this work we only take the synthetic stereo image pairs and generate the orthophotos and 3D point clouds with a traditional approach. First, we assign a fake UTM projection to all synthetic airborne stereo images, in order to enable the photogrammetric processing. Concretely, we assign the tiles to the UTM zone 31N coordinate system (EPSG:32631), even though the simulated model does not match any region on a real map, this area corresponds to the city of Paris. Additionally, for the photogrammetric pipeline we enter the camera extrinsic and intrinsic matrices, including focal length, principal points, and camera rotation and translation parameters. The extrinsic and intrinsic parameters of the synthetic data are precise and there was no artificial noise added. We assume that the deviation of the positional

accuracy is negligible, as the relative accuracy of real-world aerial images used for stereo matching is better than 0.2 pixels.

A DSM is generated from all tiles by using the CATENA pipeline [149], which is used for multiple tasks related to the processing of satellite imagery. The disparity estimation, which is the first step, is computed via Semi-Global Matching (SGM) [4], an algorithm widely used for stereo matching due to its good balance between accuracy and computational costs. SGM takes a rectified stereo image pair as input and estimates a disparity map. We apply the implementation of SGM described in [150], which takes satellite data as input, set the penalty parameters $P_1 = 400$, $P_2 = 800$ and the window size for the Census transform [5] to 7×9.

After the matching and the use of the camera parameters to determine the 3D location of each pixel, we retrieve a georeferenced DSM for each stereo pair. We subsequently merge all the stereo pair DSMs by using the median of all values belonging to the same location, resulting in one final DSM for each virtual city.

As a real DSM generation procedure, gaps are present due to matching failures or occlusions. We apply an inverse distance weighted interpolation in order to fill the remaining holes [151].

4.3.4 Orthophoto and reference data

The orthorectification process for the rendered optical tiles is implemented in a GPU as described in [152], considering as input the generated DSM, and the intrinsic and extrinsic parameters of the optical images. The outputs are take into account occlusions by buildings and vegetation. Bilinear interpolation is used to resample the orthorectified images to a given ground sampling distance.

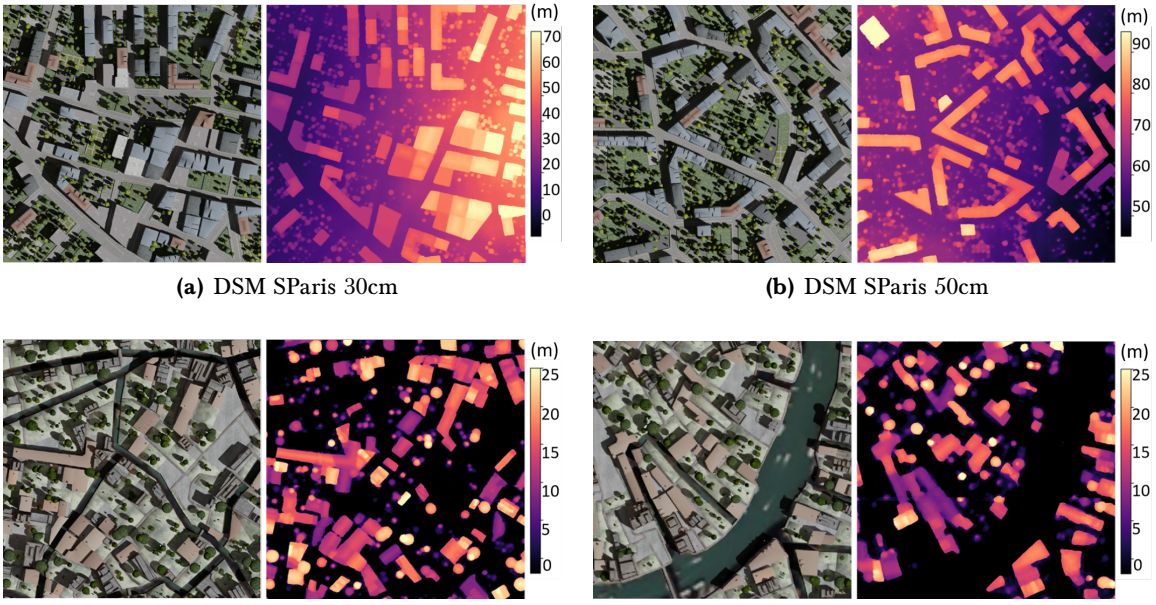
We merge all the tiles into a single large image with the warp utility from the GDAL library [153], getting as result a complete orthorectified optical image, aligned to the DSMs at pixel level.

4.4 Experimental Design

In this section we describe some additional details of the generated SMARS dataset and the delimitation of the regions used for training and testing in the deep learning algorithms for both cities. Additionally, we explain the tasks to be addressed with our generated data to show the advantages and constraints of SMARS. The details of the experiments and their results are the main work of the coauthors an therefore out of the scope of this dissertation.

4.4.1 SParis and SVenice multimodal data structure

The pre- and post-event DSMs and orthophotos are generated using the workflow described in Section 4.3. All the datasets are projected to the UTM zone 31N coordinate system and cropped in order to cover the same regions. Fig. 4.5 reports examples of the generated DSMs. Buildings appear well delimited and easy to identify in most cases, while other elements such as streets or vegetation appear incomplete or blurred. There is a clear difference between the



(a) DSM SParis 30cm

(b) DSM SParis 50cm

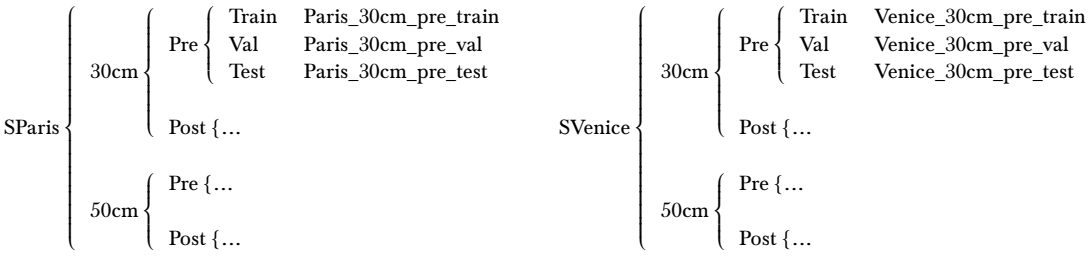
(c) DSM SVenice 30cm

(d) DSM SVenice 50cm

Figure 4.5: Example regions of the DSMs generated for SMARS besides the paired orthorectified images. All samples are taken from the pre-event models. Elevation scale for the DSM is in meters.

models obtained using 30cm and 50cm GSD respectively, as the former exhibits sharper edges with individual trees easy to identify, while the latter exhibits some blobs merging different objects. Despite some artifacts or the presence of outliers, the DSMs still have a high quality in all cases due to their synthetic nature.

The final dataset splittings are summarized in the diagram below. We list all possible subsets but report the names for only three of them for each city in order to simplify the diagram, with the remaining cases following the same nomenclature. For each subset, we have available optical images, DSMs, semantic maps, and building masks for both pre- and post-event scenarios. Additionally, we have building change detection masks for the difference between pre- and post-images. All these cases are shown in Fig. 4.6.



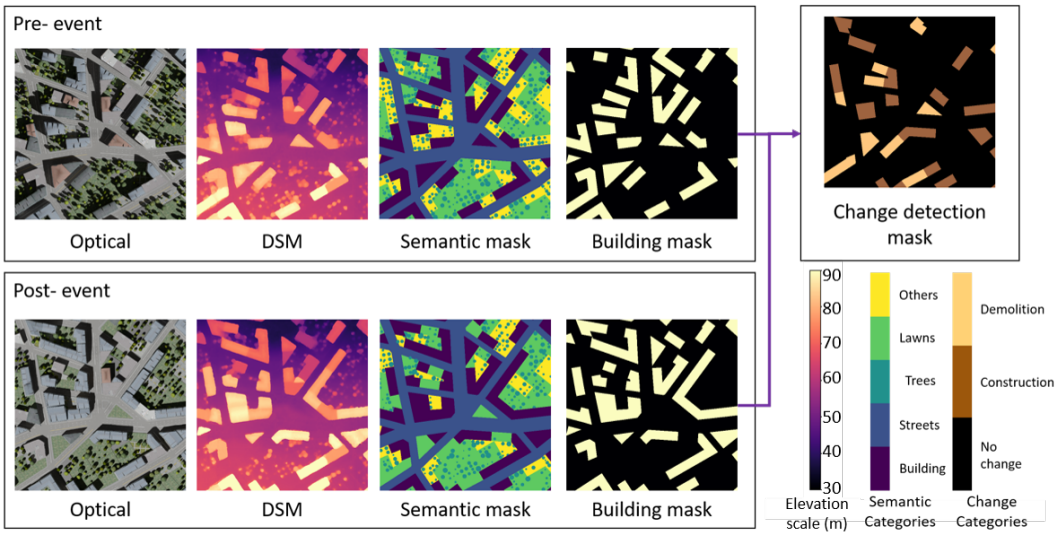


Figure 4.6: Available information for each tile in pre and post-events scenarios. For each case, an optical image, a DSM and semantic and building masks are included. For the change detection, the difference between the two events is used for the ground truth mask. Scales are given as a reference for displayed information. The elevation scale for the DSM is in meters.

4.5 Discussion

This paper proposed a novel workflow for synthetic data generation filling the gaps in the available 2D/3D multimodal data for building extraction, multi-class semantic segmentation and 3D change detection. Our data analysis looks at the effects of the domain gap when the models trained on our synthetic data are tested on real data. The discussion is limited here to the parts related to the dataset creation and observations from experiments. For a detailed explanation of the experiments and results themselves, please read the full paper.

4.5.1 Quality of the synthetic dataset

This subsection discusses the main advantages and disadvantages of the rendered images described in section 4.3. The proposed SMARS dataset meets our expectations in most of the reported experiments. Nevertheless, it also presents some limitations. Both will be discussed below for each of the available semantic categories in SMARS.

4.5.1.1 Buildings

The buildings generated by CityEngine exhibit good quality in terms of geometry, architectural appearance, and textures. They can be favorably compared to models with LoD2 and LoD3, as some rooftops have additional features such as chimneys. Moreover, the buildings resemble the expected distribution of a city in terms of size and arrangement and contribute to creating realistic scenarios. Taking into account the options to manipulate the building properties, it is easy to simulate the city growth as required for the change detection task. Furthermore, as

buildings achieve a very good reconstruction in the DSMs, they can be easily detected by the algorithms considered in this article.

Nonetheless, the pool of textures to generate the *buildings* is limited and might lead to overfitting in the learning process. Besides, no construction sites are part of the dataset, as would be the case for real images; these regions represent a challenge for change detection depending on the progress of the constructions. Another constraint is given by the generation of mostly residential buildings, as facilities such as commercial buildings, parks, sports centers, or transport stations are not included in our dataset.

In the experiments, we notice that the discrepancy in height between the two city models leads to errors for prediction in the learning models, as the DSMs values have different ranges. With traditional approaches, the similarity in height between *trees* and *buildings* can also increase the challenges of classification, especially when they are close to each other. In the SMARS dataset, the building roofs are generally well visible and do not suffer from occlusion problems as in real data, making the task of building extraction easier.

4.5.1.2 Street

A major difference between the two models is the street category. In SParis the streets match the common design with sidewalks, concrete material, and broken and solid lines. Besides, streets in this model are wide and have a height profile different from all other elements, with the exception of lawns.

SVenice is more difficult in this category. In the same way as the real Venice, the streets are designed for pedestrians, and are therefore narrow, causing sidewalks to be absent and are not marked either by broken or solid lines. Additionally, the width of the streets is comparable to the one of the multiple canals crossing the city. This problem is aggravated by the similarities in terms of height between the "others" (where canals and sea are included) and street categories. Because of that, it was observed the semantic segmentation task that cross-domain experiments drop significantly in performance for this category. For learning models trained with SParis, the canals of SVenice are considered streets and the lawns are predicted as "others". Likewise, for learning models trained with SVenice, the streets of SParis are many times wrongly labeled as "others" and only a few streets are actually detected.

As width and height are within the expected ranges for streets, a suitable solution would be to enhance the available categories in order to incorporate canals, squares, roundabouts, alleys, and other elements that could be confused with roads.

4.5.1.3 Vegetation and lawns

Representation of shapes and structures of trees and bushes in 3D is a critical issue. A detailed representation requires a complicated geometric definition leading to high computational costs. A common simplified case with only two intersected vertical planes reduces substantially the memory requirements but exhibits poor visual quality in the models. Due to the trade-off between memory and appearance, we used textured ellipsoids. This allows the inclusion of a

large number of trees and bushes in the virtual scenes. We include many textures, but these are limited to a specific number of plant species.

Yet, the *vegetation* regions largely suffer from the domain gaps between synthetic and real data. Real scenes have no simplified geometry (with the exception of man-trimmed trees) and cannot be easily modeled. Using only ellipsoids makes the learning biased towards this shape, and cannot adequately lead to correct predictions of other types of vegetation. Also, seasonal effects (such as leave colors, snow covering, or fallen leaves) are not considered.

On top of that, the *lawns* category has been simplified too. Actual grass has a non-negligible height (even if this is relatively small in comparison to the other objects), no uniform texture, and can include small vegetation such as low bushes. For the simulated cities, the *lawns* are simplified by a flat area with grass-like texture, which appears realistic enough in the orthophotos. Without the texture, the *lawns* would be similar to the *roads* or *bare soil* category, as the height information of *lawns* is set close to 0.

In DART, *trees* are defined by tree species, various attributes of trunk and crown, and are simulated using turbid voxels or isosceles triangles [138]. Tree crown shapes can be chosen from ellipsoidal, ellipsoid-composed, truncated cone, trapezoid, and cylinder with truncated cone. In addition, branches and twigs can be added. However, the tree modeling requires many manual input and is still not realistic as desired. Nevertheless, there is still potential to improve the quality of the *trees* class by using existing detailed 3D tree models. For example, the RAdiation transfer Model Intercomparison (RAMI) experiments derived detailed and realistic 3D models of various tree species by in situ measurements. The 3D models have been exported to DART, and can be edited in Blender as well. But those tree models do not include enough typical urban tree species to represent the urban tree scenario. For the reasons described above, we did not adopt these accurate tree 3D models.

4.5.1.4 Water

Water is not an annotated category in our SMARS dataset. However, it is an important land cover type in the SVenice scene. In the provided Venice city model of CityEngine, the water bodies are actually covered by a real low-resolution satellite image, exhibiting shadows that might not correspond to the simulated sun conditions. In addition, elements present in the water (such as boats and bridges) do not have an above ground height, so the captured multi-view images do not present a meaningful disparity in the epipolar image pairs. Therefore, in the generated DSMs the surface of water bodies is rather flat and smooth. In reality, the elements present in the water would have a height value larger than zero.

On the other hand, the SParis model has no *water*, so these are absent in the ground truth for either city, an aspect which can lead to errors in the semantic segmentation task, especially for cross domain experiments. It is particularly complex for the algorithms to separate water from streets in the SVenice model, where the canals have similar contextual features as the *streets* in SParis. The collection of a larger number of samples with labeled *water* coverage might help solve this issue.

Finally, since we use an aerial photo as the source for the water areas, these do not change between the pre- and post-models and remain also constant within the simulated flight campaigns. In reality, the waves and tides produce an irregular surface, causing the matching algorithms to yield poor results. Usually, the DSM pipeline would fail to reconstruct such regions, while our DSM has a constant value. As discussed above, a physical simulation of water would lead to enhanced realism in the scenarios. Since our work focuses mainly on buildings, this is currently left out of our studies.

4.5.2 General observations

In this chapter we introduced SMARS, a synthetic large and accurately annotated 2D/3D multi-temporal earth observation dataset, as an effort to meet the demand for multimodal benchmark data suitable for change detection applications in urban areas. In addition to 3D change detection, we provide orthorectified images, DSMs and ground truth for semantic segmentation, along with a pipeline to generate similar synthetic images resembling the characteristics of real aerial acquisitions, including their limitations. By modifying the scenes within the pipeline, it is easy to set and adjust the changes between two simulated acquisition times, which is a difficult task when using real data. As a result, the pipeline has the potential to create larger samples with high variability.

The ground truth associated to the dataset is free from wrongly annotated labels or confusion between classes, being generated during the rendering process. This aspect propagates its advantages to the change detection applications, where a large number of modifications can be handled and are ensured to be correct in the change mask to be used as reference. The quality of the presented synthetic data has been investigated in several experiments, which yielded results similar to what would be expected using real data. The quality of SMARS data is high in terms of coregistration, orthorectification and ground truth quality.

In addition to testing segmentation and change detection approaches, the presented synthetic data can be adapted to train a valid building extraction or semantic segmentation model that can be applied to real datasets. For instance, building extraction shows a good performance on the ISPRS Potsdam dataset, even without a fine-tuning step. In general, the synthetic data represent a feasible option for training neural networks for building detection, semantic segmentation, and change detection tasks, despite expected constraints due to domain gaps.

5

Generation of urban DSMs using stereo and multi-view deep learning algorithms

Contents

In this chapter, we refer to the third article contribution (under review process), where the stereo matching and MVS strategies are compared in a similar setting to analyse the benefits of each one to reconstruct a 3D urban scene. We describe qualitatively and quantitatively the performance of the compared cases. Data from synthetic and real acquisitions is used in the experiments to get more insights. The used real data, which corresponds to the city of Dublin, has been released to be used by the research community.

5.1 Background

The task of generating DSMs is a first step in many remote sensing pipelines. Data from different sensors and platforms (usually aerial or satellite) can be used as input for this task, like images from traditional cameras, LiDAR or synthetic aperture radar (SAR). For this chapter, we focused on the case where a DSM is created from optical imagery only, as this is often cheaper than the other sensors and offers sharp geometry for the reconstruction.

Currently deep learning based algorithms are state-of-the-art, however, many of these depend on supervised learning methods and a requirement for that is the availability of ground truth for training, which is still measured with LiDAR. This data acquisition is expensive and the quality of the ground truth depends on the density of the generated point cloud. Despite this issue, learning models have the advantage of being trained on a subset of data and tested on many other samples, so the ground truth is just required for the training step, allowing the model to predict in many unseen samples, as long as the domain gap is not too large.

After obtaining a good dataset capable of training deep learning models, most existing network architectures are oriented towards either stereo matching or MVS approaches. While both are suitable for generating a DSM, they are based on different principles and therefore require different input data and network architectures.

The stereo algorithms expect data that has undergone epipolar rectification, which means that the points to be matched are along the same epipolar line and we only consider candidates in one dimension. To calculate the height of objects in the scene, the baseline between the two images, the focal length of the camera, the position/orientation of the stereo array and the computed disparity map are needed.

MVS on the other hand does not need stereo rectified images, as it supports images from different points of view. Nonetheless, the correct relative position/orientation between the cameras is required for a homography warping. The algorithms estimate a depth map that can be converted into a height map based also on the reference view position and rotation.

As deep learning architectures have evolved and achieved the best performance in the benchmarks, the differences between the two algorithms have become more pronounced. Datasets are designed separately for each case, as well as metrics and benchmarks. We already set the first experiments to evaluate both stereo and MVS algorithms in stereo paired images in our previous work [154], but we now explore multiple views and also test all the algorithms on real data. We use the available datasets SyntCities [90] and Dublin 2015 [155], where synthetic and LiDAR ground truth is available respectively. The aim was to make the comparison as fair as

possible. This would highlight the differences between the algorithms. Metrics for all cases and discussions are presented for all the obtained results.

In the traditional pipelines for DSM generation, a set of candidate values is available for each pixel/location, which are later fused by using the median to determine a robust final value [41]. In practice, stereo methods are more widely used for remote sensing data as they have been studied longer, just few pre-processing steps are needed and the matching works only along one dimension. MVS methods require less pre-processing steps and might benefit from the information provided by additional views, but they have been less studied.

We explored beyond the traditional fusion, by using a confidence estimation which could help to pre-select the best candidate values before fusion. The confidence estimation responds to one of the remaining issues of deep learning, the fact that there is a prediction for each pixel, whether this is a reliable one or not. The confidence estimation aims to give a value related to this certainty, which we use to sort the available height values used to be fused in the DSM. Although the improvement in the DSM accuracy is small, the experiments show that there is potential for further research in this direction.

Summarizing, our main contributions are:

- A fair comparison of learning-based stereo and MVS methods while using multiple views/stereo-pairs for the same region.
- We evaluate the algorithms in synthetic data, where the ground truth is highly accurate and on the real images, as an application case with challenging regions.
- We explore an alternative way to fuse the height values into a DSM by using the confidence associated to each prediction made by the neural networks.
- We share the processed Dublin dataset [155] to have a large dataset compatible with stereo and MVS algorithms [i].

5.2 Related Work

In this part we describe some of the main algorithms and neural networks applied to the tasks of stereo matching and MVS highlighting also their differences. Besides, we introduce some available algorithms for the confidence estimation in the stereo matching case.

5.2.1 Stereo Methods

Prior to deep learning solutions, stereo algorithms were mostly based on a cost volume generation pipeline and its refinement to produce smooth results. Usually the steps for stereo estimation are matching cost computation, cost aggregation, disparity estimation and disparity refinement [8]. A widely used algorithm for stereo matching is Semi-Global Matching (SGM) [4], which can be implemented also to work in real-time due to its compromise between efficiency and accuracy. As it is the case with non-learning algorithms, it can be applied to any pair of

[i]The processed Dublin dataset can be downloaded at: `https://tinyurl.com/2hmmc4z2`

images without prior knowledge and produce a good quality result. Nonetheless, the tuning of the penalty parameters has a strong influence on the performance of the algorithm.

Recently, deep learning solutions have been the leading approaches for stereo matching. MC-CNN [11] replaced the cost matching part with a neural network and included the refinement of SGM, showing a good performance. Later on, end-to-end networks were developed to predict the disparity maps from the stereo images, learning also the refinement steps. The first approaches were DispNet [12] with an encoder-decoder architecture and GC-Net [13] that incorporated 3D convolutions. Among the architectures that are widely known and used as a baseline to compare performance, we can mention GANet [17], AANet [19] and DSMNet [20]. GANet is a learning-based implementation similar to SGM, where the penalty parameters are learned and 3D convolutions are used to refine thin structures. AANet produces smooth results and avoids the expensive 3D convolutions using less memory than GANet with a slight loss in accuracy. DSMNet on the other hand, tried to reduce the domain gap by using a domain normalization.

Newer architectures benefit from more complex architectures. RAFT-Stereo [21] adds gated recurrent units (GRUs) for a robust result in difficult areas, like textureless sections. Besides, it is less affected by the domain gap problem. A different strategy is STTR [24], where transformers are included and the network also alleviates the constraint of a fixed disparity range.

In our study, we will use only AANet as it requires less time for training/inference than other networks, produces a good quality result, and is a common baseline to compare new methods.

5.2.2 MVS Methods

The multi-view networks do not require the input images to be on the same epipolar line and therefore allow the reconstruction to be based on multiple points of view. Such a reconstruction takes place directly in the 3D space, so the predictions represent the distance from the camera plane to the objects as in the traditional sweep plane algorithms. In contrast to stereo methods, the MVS approaches require a estimated depth range as well as the relative camera positions and rotations values.

Non-learnable photogrammetric algorithms have been developed for this task. COLMAP [156] reconstruction benefits from multi-view geometric consistency, and its algorithm to sort the additional views (with respect to a reference view) is used also by deep learning solutions as a starting point. GIPUMA [28] applies an iterative process in the 3D space which is computed efficiently by using GPU resources.

Deep learning architectures have also been leading the MVS benchmarks in the last years, especially in terms of completeness. MVSNet [30] is a pioneering work that implements the plane sweep algorithm in a learnable way. R-MVSNet [31] includes GRUs which help to slightly improve the results. Another strategy is CasMVSNet [32], that follows a coarse-to-fine architecture reducing the memory consumption and allowing higher image resolutions. VisMVSNet [35] incorporates information related to the occluded pixels to rely on visible pixels for a more robust reconstruction. UniMVSNet [36] has a depth representation that allows the network to consider both a classification and a regression task simultaneously, leading to

significant improvements in the performance. On top of that, computational resources are less demanding than for other networks. Therefore, we select UniMVSNet for the experiments in this chapter.

5.2.3 Confidence estimation

The confidence estimation is a research area that has already been explored in the task of stereo matching. Given a disparity map, which is predicted with a neural network (or a photogrammetric algorithm), the confidence estimation aims to give a value that is related to the certainty of the prediction for each pixel in the result. This would be similar to some post-processing steps applied in the stereo matching, such as left-right check consistency, where according to the bilateral reprojection of the images using the disparity maps, some disparity predictions are discarded due to inconsistencies.

As with the disparity and depth estimation tasks, the confidence can also be estimated by learnable and non-learnable algorithms. Regarding the latter ones, one of the first quantitative evaluations is shown in [52]. Most of the evaluated algorithms are based on the cost volume used to estimate the disparity values. Confidence for each pixel can be computed directly from the cost, by evaluating the curvature of the cost curve, analysing the presence and distribution of the local minima, the behaviour of the whole cost curve or by using the left-right consistency as already mentioned.

With respect to learned-based algorithms, a quantitative evaluation can be found in [53]. These algorithms take as input the input reference image, the predicted disparity maps and/or the cost volume, although the latter increases significantly the memory consumption in the implementations. CCNN [54] was one of the first architectures designed to predicted confidence maps by using Convolutional Neural Networks (CNNs) and Fully Connected Networks (FCNs). Since this method did not use the cost volume as input, it is more flexible to test in other stereo matching algorithms. PBCP [157] used a patch based solution on maps predicted by SGM and significantly reduced the confidence prediction error. PKRN+ [55] included layers able to capture not only the information for the computed pixel, but local context to estimate the confidence. In this way, regions with similar confidence values are smoother. A different architecture [56] proposed to use not only the disparity map, but the cost volume as input for the network. To reduce the high computational cost required to handle the whole cost volume, only the highest costs are selected using the "top-k" operation from PyTorch. Finally, LAFNet [57] takes reference image, disparity map and cost volume (with the same preprocessing as [56]) as inputs and includes convolutional spatial transformers in the architecture, leading to a remarkable performance between the state of the art solutions. Hence. we selected LAFNet for our experiments related to the confidence-based estimation.

Since LAFNet requires the cost volume as input, we had to select a neural networks that are based on a cost volume approach. The previously selected networks for disparity and depth estimation, namely AANet and UniMVSNet were also chosen because their cost volumes can be exported to be used as input for LAFNet. Although LAFNet has been designed exclusively for disparity maps and no for the MVS case, we explored using the depth maps with their respective cost volumes as input a in a similar manner to stereo data.

5.3 Methodology

In the following paragraphs we describe the process used to fuse the data (with and without confidence guidance), the nature of the data used for our 3D reconstruction experiments including preprocessing steps as well as the training conditions of the applied stereo and MVS networks. For the MVS network, we considered two cases, applying it as a stereo matching algorithm (which means many input stereo pairs) and as a full multi-view algorithm (where many views are taken simultaneously as input). Hence, we analysed three cases, namely: Stereo, MVS_Stereo and MVS_Full. The datasets to be used are SyntCities for the evaluation on synthetic data and Dublin for real applications.

5.3.1 Predicted maps fusion

Different methods can be used to estimate the disparity/depth maps as a first step to generate a DSM. Since the images are usually cropped into tiles due to memory and computation restrictions, we end up with a stack of smaller DSMs to be fused. To merge these results into a DSM, steps are different for stereo and multi-view cases.

The pipeline to fuse predicted disparity and depth maps is shown in the Fig. 5.1. We represent here a case to fuse 6 images of SyntCities, but the principle is the same for the Dublin data.

Starting from the stereo cases, which are Stereo and MVS_Stereo, we have a total of 15 possible combinations, and we always consider the disparity map from left to right to get positive values, which is a restriction for the estimation of the networks. The 15 disparity maps are then converted into height using the camera parameters along with the baseline and subsequently georeferenced using the camera positions. Nonetheless, the transformation of the disparity maps to height maps is still influenced by the acquisition perspective, having an oblique view. Hence, it is necessary to orthorectify the images to have the geometry required for the DSM.

We also have the MVS_Full case. Using the algorithms for MVS estimates the depth for only one of the views at a time, which is considered to be the reference view while the additional views provide complementary information. This means, we obtain 6 depth maps as a result of giving the same number of input images if we use the rest as additional views. Although the number of results may seem smaller than in the stereo case, the same number of images is used within the algorithms. After estimating the depth for each view, we transformed this into height using also the camera parameters. Similarly to the stereo case, the height map is still oriented to match the camera perspective and required orthorectification as well.

Having all the results as orthorectified height maps, it is now possible to fuse the results into a single DSM, benefiting from all single estimations. We considered two basic yet widely used methods: mean and median for each pixel/location. The former provides insights of the distribution of the predicted results. The latter is more effective and makes a robust fusion by avoiding the influence of outliers, being the most common strategy.

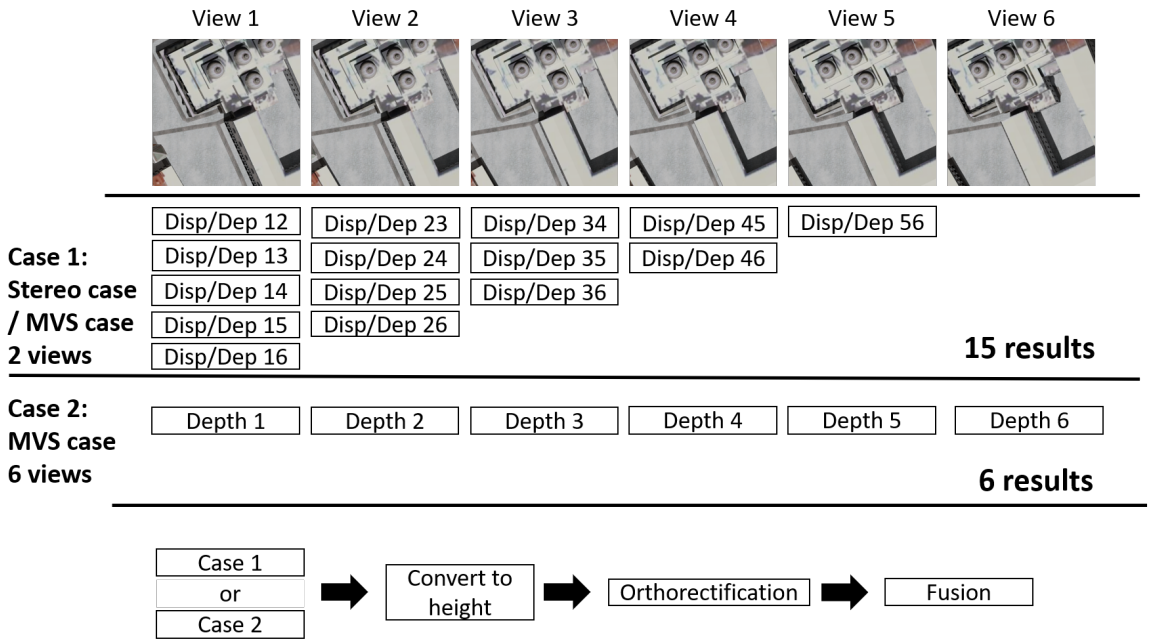


Figure 5.1: Pipeline used to fuse the results of the predicted disparity/depth maps. In the case of the Stereo and MVS_Stereo methods, more results are available but they use the same available information as the MVS_Full case. All results then follow the same steps which include height conversion, orthorectification and fusion.

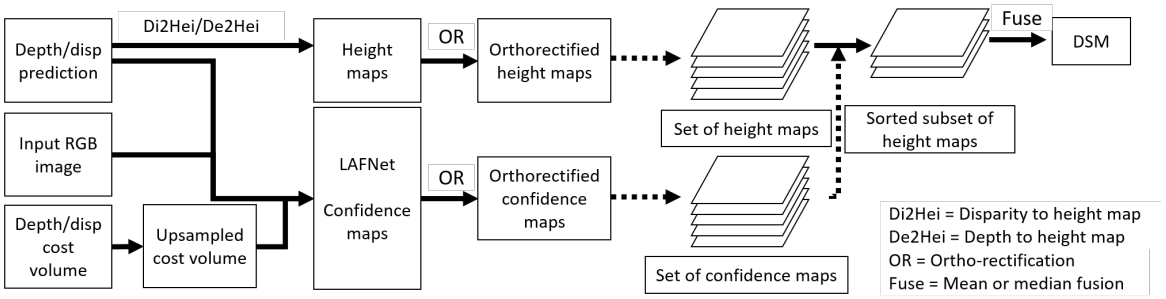


Figure 5.2: Pipeline for confidence-based fusion. After estimating confidence maps along with the height maps obtained from the reconstruction algorithms, a stack of height maps is sorted based on the respective confidence values and then we compute the median to get the final DSM.

5.3.2 Confidence based fusion

We also analysed the case of fusing the depth and disparity maps using a confidence based fusion. A diagram to explain the process is shown in the Fig. 5.2, but we describe here the steps in detail. The confidence maps help to fuse the depth and disparity maps, so we need to process all the data simultaneously.

First, disparity/depth maps are converted to height maps using photogrammetric algorithms. For this step, LAFNet is not required, just the results from the Stereo, MVS_Stereo and MVS_Full algorithms. In parallel, the same depth/disparity maps along with the cost volume (which has

to be upsampled) and the RGB images are used as input to LAFNet, generating a confidence map as a result.

After that, both height and confidence maps are orthorectified. Since these are obtained for the same regions, the orthorectified maps cover the same pixels/areas. If we apply these two steps to all input depth/disparity maps, we end up with a stack of height and confidence maps.

In the above fused cases, we would only apply the median to all the candidate height values for each pixel to obtain the fused height. We do propose a different strategy to fuse the height values by using the corresponding confidence values. We sort the stack of confidence maps according to the values for each pixel from higher to lower, and based on this sorting, we re-arrange the stack of height values as well. Afterwards, we remove the less confident height values according to a removal percentage ($rem_{\%}$). For example, if we have 10 height values for a certain pixel and set $rem_{\%} = 50$, only the 5 candidates with higher confidence remain. We compute the median from the remaining values to generate the DSM.

5.3.3 Data Preparation

As mentioned in the introduction, datasets for stereo and MVS algorithms have been designed separately for each task, making it difficult to establish a common dataset to assess the performance reconstruction of both approaches. To overcome this obstacle, we decided to prepare two datasets for our experiments. First, we used SyntCities as in our previous work [154], but instead of using only two views for all cases, we selected additional views and different baselines. Second, we also evaluated the performance of the algorithms on real data, so we processed the Dublin dataset [155] to be compatible with both approaches and generated the required ground truth. Detailed information is given in the next sections.

5.3.3.1 SyntCities

The SyntCities dataset is a synthetic dataset that was developed to compensate for the lack of stereo paired data in the remote sensing field. Since these images are generated directly from the 3D software Blender by using BlenderProc [102], the ground truth is accurate and dense, which means we have a reliable reference value for all pixels. The images have been rendered at a ground sample distance (GSD) of 10cm, 30cm and 1m. In the original setting, 4 pairs are given for the same area with different baselines. For the new experiments, we benefit from the fact that despite having different baselines, all tiles with the same naming number (based on the SyntCities file organisation) are on the same epipolar line. The SyntCities dataset assumes that the camera follows a flight track over the scene and acquires the images at 25 locations; as those points act as the center for the stereo arrays, we generated the stereo pairs by simply increasing the baselines. Hence, for each location we have 8 images along the epipolar line considering the left and right views (4 baselines x 2 views). The selected testing samples have a GSD of 30cm and 1m and belong to the Venice and Paris samples, as height differences are not so large in these cities.

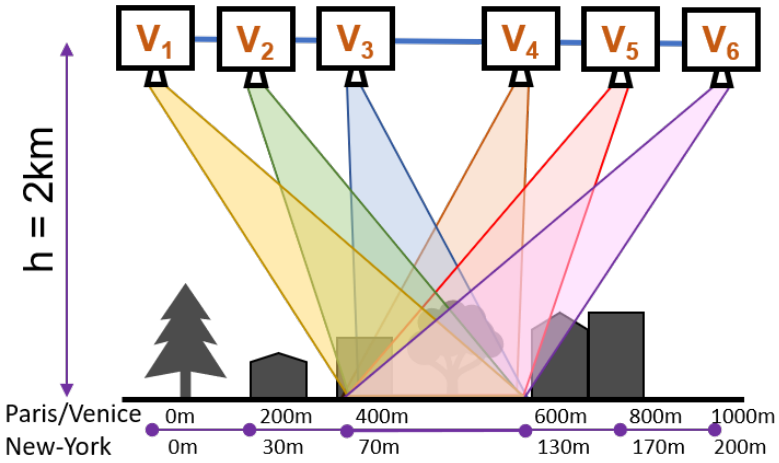


Figure 5.3: Selected geometry for SyntCities samples. All images lie on the same epipolar line with different baselines. There are 6 available views for each region on the surface. Baseline distances are given with respect to V_1.

In our experiments, we used a maximum of 6 views for each location. Due to the camera parameters of the stereo pairs, all images cover approximately the same area on the ground, as shown in the Fig. 5.3, where all the cameras are pointing to a common area. Assuming that we select V_N ($N \in [1, 6]$) as the reference view, we have 5 additional views to help for the reconstruction of V_N. The distance between the cameras is given in the image as baselines with respect to V_1. The cameras were not rotated nor displaced out of the epipolar line. As SyntCities included ground truth only for the default stereo pairs, we generated the missing disparity maps from the depth maps (available for all views) and the camera parameters. Apart from that, no additional data is required.

5.3.3.2 Dublin dataset

The Dublin dataset[ii] is a collection of data acquired on 2015 over the downtown of Dublin, Ireland. The campaign had a flying altitude of 300m and retrieved LiDAR data (as point clouds and full waveform), oblique images, geo-referenced RGB and infrared imagery, and the respective acquisition metadata.

As a first step, we downloaded all the point clouds and merged them to create a single DSM, as the ground truth was later computed from it. The DSM was created with a GSD of 10cm and is shown in Fig. 5.4. Due to the sensor acquisition not all the pixels will have a ground truth, but for those where the value is defined, this is computed from a dense measurement, offering a good quality ground truth.

We selected the georeferenced RGB imagery as input for our experiments. The original images had a size of 9000×6732 pixels with a GSD of 3.4cm. We downsampled the images by $\times 9$, changing the images to a size of 1000×748 pixels with a GSD of 30.6cm, similar to the one in

[ii]The original Dublin dataset can be downloaded at: `https://geo.nyu.edu/?f%5Bdct_isPartOf_sm%5D%5B%5D=2015+Dublin+LiDAR`

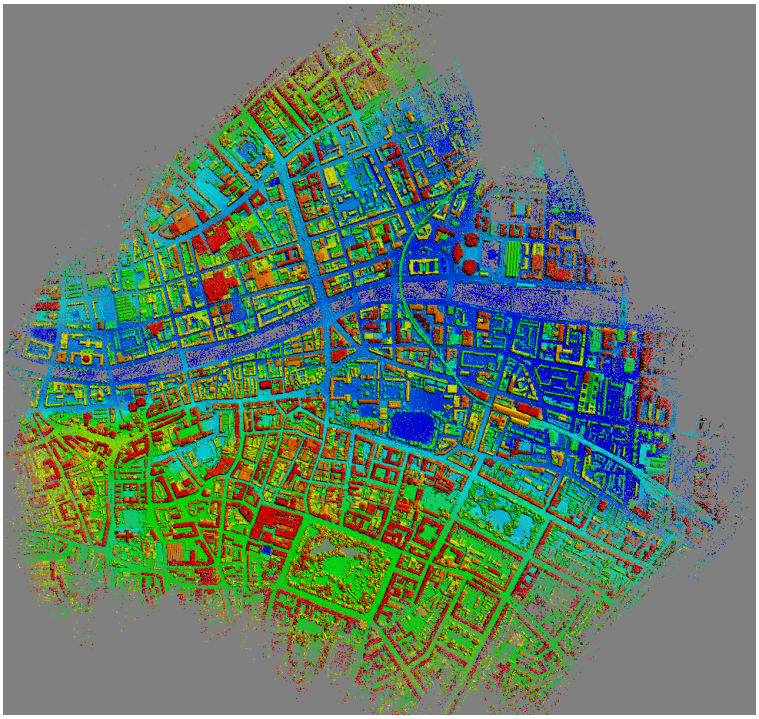


Figure 5.4: Dublin digital surface model obtained by merging all provided point clouds and used as ground truth .

SyntCities. With the downsampled size, it is also easier to use the images as input for the neural networks without additionally cropping and merging the tiles for pre and post processing.

The data was further processed for the two input cases: Dublin_stereo and Dublin_MVS. A diagram for the applied pipeline is shown in Fig. 5.5, where we have K input images. In the case of the Dublin_stereo dataset, we selected a pair N of the K downsampled images, the pair had to be epipolarly rectified for stereo matching. For each image, we selected the 5 closest acquisitions (based on the Euclidean distance of the positions) to set the pairs. The epipolar rectification is done with the compact implementation described in [158]. Once the pair has been rectified, we use a photogrammetric algorithm to convert from the DSM to a disparity map, which is aligned to match the "left" image of the pair (so the disparities have a positive range as required for the networks). Hence, the stereo dataset includes pairs of rectified images with the respective disparity ground truth. A pair examples of the Dublin_stereo dataset are shown in Fig. 5.6.

With respect to the Dublin_MVS dataset, after downsampling the images, we processed the camera values for positions and rotations from the metadata to be compatible with the format required for the camera files in the MVS approaches, which includes camera extrinsics, intrinsics and an estimated depth range where the scene is located. The depth range is computed from the DSM, with a range that includes $\mu \pm 4\sigma$, being μ and σ the mean and standard deviation of the depth values according to the camera parameters. This range is different for each image.

The depth ground truth is obtained in a similar way to the stereo case, where we used the DSM and photogrammetric relations to convert the DSM into the depth map for each image. As the

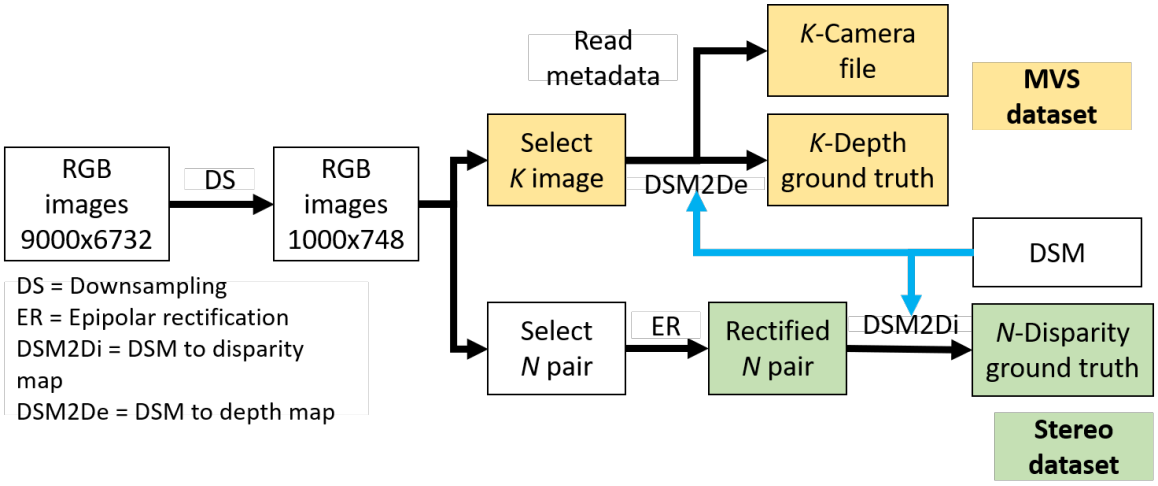
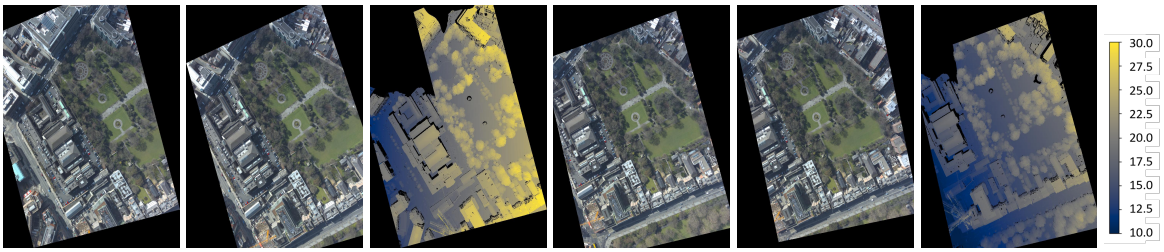


Figure 5.5: Pipeline used to generate the Dublin dataset for both cases: Dublin_stereo and Dublin_MVS.



(a) Left A **(b)** Right A **(c)** GT A **(d)** Left B **(e)** Right B **(f)** GT B

Figure 5.6: Dublin_stereo dataset samples. (a) and (d) are the left views for the corresponding (b) and (e) right views, (c) and (f) are the ground truth aligned with the left views. Bar scale for disparities is in pixels.

depth map does not depend in the additional views, it is always the same for a specific image and we do not need to provide ground truth for different image pairing. Therefore, the MVS dataset includes the RGB images with the respective depth map and camera file. An example of the images included in this dataset are shown in the Fig. 5.7.

(a) Image A **(b)** Depth GT A **(c)** Image B **(d)** Depth GT B

Figure 5.7: Dublin_MVS dataset samples. (a) and (c) are the reference views for the corresponding (b) and (d) ground truth. Bar scale for depth is in meters.

The Dublin dataset acquisition track has a different geometry to the one presented for SyntCities. For the Dublin campaign, images were taking with a single camera along the flight path.

Therefore, the images cover different areas with some overlapping between adjacent acquisitions. In the Fig. 5.8 we show a simplified diagram of the camera positions and ground coverage. A distance of approximately 100m is given between two consecutive images, leading to a side overlapping of ~ 70%.

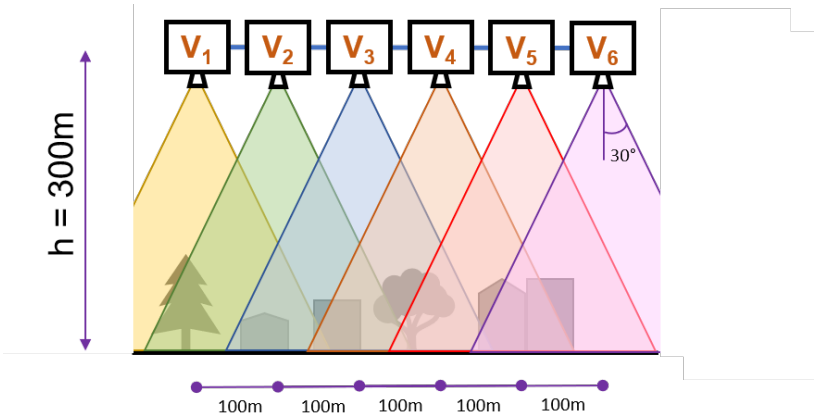


Figure 5.8: Selected geometry for Dublin samples. Images lay on a flight path with an approximate baseline of 100m, but not in the same epipolar line.

Unlike the SyntCities case, in the Dublin dataset some regions are not visible in adjacent input views, which makes the matching more challenging than for the synthetic data. Moreover, the density of objects and textures in the Dublin dataset is larger, posing additional difficulties for the reconstruction algorithms.

5.3.4 Stereo training

We train AANet for stereo matching in both SyntCities and Dublin (stereo dataset), training from scratch with SyntCities and used this model to finetune on the Dublin data. We followed this strategy as the ground truth for SyntCities is dense and accurate, so the finetuning would help to reduce the domain gap for the testing area. For SyntCities, from the original 5400 images from the training subsets, we removed 300 cases with large baselines, keeping 5150 for training. 22 samples from the test subsets with 5 views each, so 110 samples were used for testing. In the case of Dublin, from the available tracks, we selected the subset 150326_122941 for finetuning and the subset 150326_120403 for testing.

The training for SyntCities takes different views along the epipolar line as explained previously for Fig. 5.3. We used a batch size of 20, trained the model for 370 epochs and called this model Stereo_SC. The finetuning is done with the Dublin stereo samples for additional 500 epochs. We reduce the maximum disparity to 96 as this range is enough for these samples. We call this model Stereo_Du. Training was conducted on 4×NVIDIA GeForce RTX 2080 Ti GPUs.

5.3.5 MVS_Stereo and MVS_Full training

Similarly to AANet, we train firstly on SyntCities and then finetuned the model on Dublin samples. However, we apply two different training models for UniMVSNet: as a stereo matching case and full multi-view, which means 2 and 6 views as inputs respectively. The first case will help to study the performance of UniMVSNet with conditions very similar to AANet, and we call this case MVS_Stereo. The full multi view is intended to give data to compare the impact of having more views as input and if this is beneficial for the reconstruction. We named this case simply MVS_full.

In the MVS_Stereo based instance, we train UniMVSNet on SyntCities for 40 epochs with 2 input views, a batch size of 2 and the image pairs are loaded with the same pairing order as for AANet. Afterwards, we finetuned the model for additional 270 epochs. We call these models MVS_Stereo_SC and MVS_Stereo_Du for SyntCities and Dublin respectively.

Similarly, we train the MVS_Full case with UniMVSNet by applying a number of views of 6 for 160 epochs. The number of iterations is larger as there are less possible combinations of input images as for the stereo case. For the finetuning we applied additional 600 epochs. These models are named as MVS_Full_SC and MVS_Full_Du. Finetuning models had more epochs due to the relatively fewer samples in Dublin comparing to SyntCities.

5.3.6 LAFNet training

LAFNet requires the cost volumes as inputs along with the RGB images, the predicted depth/disparity maps and the depth/disparity ground truth maps. While using algorithms such as SGM or MC-CNN, the whole cost volumes are easy to identify and export as additional files, providing also information for each pixel. However, neural networks usually use structures where the volumes are downsampled to reduce computational resources. Moreover, the volumes in the coarsest resolutions generally offer a better overview of the matching, as they take into account the full disparity range. The finer volumes mostly refine around a certain disparity range, not the full one. Hence, we used the coarsest cost volumes from AANet and UniMVSNet, in both cases after the aggregation steps to reduce the presence of outliers.

We adapted both networks to export the cost volumes as described above. Besides, LAFNet applies a pre-processing step to the input cost volumes as mentioned in [56], where the values are normalized to improve the discriminative power of the network and the “top-k” function selects the main cost candidates only. This helps also to reduce the memory demands of the algorithm. In order to also reduce the storage space required for the cost volumes, we apply this processing step before exporting the cost volumes. It also avoids additional processing each time the LAFNet is loading the data.

Nonetheless, using the coarse cost volume makes the input data to be mismatched in terms of resolution. We solved this by interpolating the stored coarse cost volume to match the input image. A more sophisticated upsample strategy based on learning parameters might provide a better result, but we keep that out of scope as our purpose is not to design a new confidence learning network.

We also observed that LAFNet uses a binary cross entropy loss to segment the confidence mask into the ideal case of confident and non-confident pixels. Still, we would like to study the effect of using L1-loss based on the error instead. The confidence estimation is based on an error threshold (common values for disparity threshold errors are 3 and 1 pixels) and is computed from the difference between the predicted and ground truth disparities as:

$$diff = \begin{cases} |disp - disp_{gt}| & \text{if} \quad |disp - disp_{gt}| \leq err_t \\ err_t & \text{if} \quad |disp - disp_{gt}| > err_t \end{cases} \tag{5.1}$$

$$conf = 1 - \frac{diff}{err_t} \tag{5.2}$$

where err_t is the error threshold, $disp$ the predicted disparity value, $disp_{gt}$ the ground-truth disparity value and $conf$ the confidence value used as ground truth for LAFNet. Due to the clipping of the disparity difference ($diff$), the confidence values are restricted to $0 \leq conf \leq 1$.

Since the real data is more challenging and the confidence can help to distinguish bad predicted areas, we trained only on the Dublin dataset. We trained LAFNet for 250 epochs, with patches of 494×494 pixels and a batch size of 4. The LAFNet models were trained on one NVIDIA GeForce RTX 2080 Ti GPU and we call this model Conf_Stereo. The original input cost volumes, which were obtained with AANet, were upsampled by $\times 3$ to match the images input size. For the results coming from UniMVSNet, we upsampled $\times 4$ the input cost volumes, and these models were trained for 350 and 1000 epochs for the MVS_Stereo and MVS_Full cases respectively, naming them as Conf_MVS_Stereo and Conf_MVS_Full. The latter had more epochs as the number of input depth maps is lower than the former.

5.4 Results

In this section we present the qualitative and quantitative evaluation of the fused models in comparison to the ground truth DSM. For the three applied algorithms (Stereo, MVS_Stereo and MVS_Full) we used both SyntCities and Dublin sets, with a total of 6 DSMs to be evaluated.

5.4.1 Metrics

We consider three metrics to evaluate the accuracy of the fused models, which are:

- Median Absolute Deviation (MAD). Since the median based metrics are more robust to outliers [108] we apply MAD, which can be derived from the median of the difference (Med_{diff}). The median of the difference is computed between the ground truth and the fused DSMs. This is computed as:

$$\text{Med}_{\text{diff}} = \text{median}(X_{\text{diff}}), \quad X_{\text{diff}} = X - \bar{X} \tag{5.3}$$

where X is the ground truth, $\bar{X}$ is the compared DSM and X_{diff} is the difference between both. Second we compute the MAD as:

$$\text{MAD}_{\text{diff}} = \text{median}(\mid X_{\text{diff}} - \widetilde{X}_{\text{diff}} \mid) \tag{5.4}$$

where $\widetilde{X}_{\text{diff}} = \text{median}(X_{\text{diff}})$

- Error rate 3 meters (e3m). This metric is similar to the error rates for stereo matching algorithms, but using meters instead of pixels. From all evaluated pixels, we compute the percentage of them where the error is larger than 3 meters.
- Error rate 1 meter (e1m). This metric works the same way then e3m, but for a stricter margin of 1 meter.

5.4.2 Results SyntCities

We do analyse first the results for the SyntCities. As the data has a synthetic nature, the networks faced a simplified case where a controlled environment was used to render the scenes. Nonetheless, as the ground truth is very accurate, these experiments provided insights about the matching capabilities of the algorithms.

We evaluate the models Stereo_SC, MVS_Stereo_SC and MVS_Full_SC, which were trained on SyntCities and applied the median to fuse all height maps into the final DSM. The results are shown in Table 5.1. A total of 22 scenes were evaluated and the results are averaged from individual results.

Table 5.1: DSM generation metrics, based on the fusion of stereo and MVS results for the SyntCities dataset

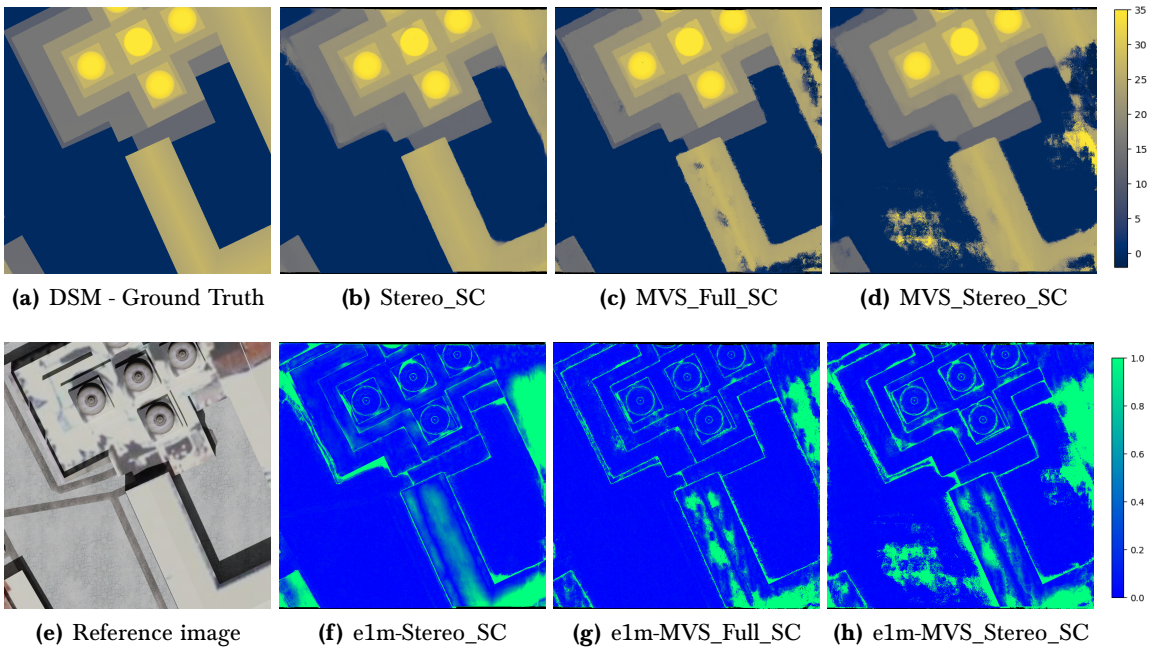
		Metrics		
Network	**Fusion**	**MAD (↓)**	**e3m (↓)**	**e1m (↓)**
Stereo_SC	Mean	1.553	11.385	26.224
	Median	0.390	9.385	22.128
MVS_Full_SC	Mean	0.320	13.049	26.022
	Median	0.299	10.558	22.308
MVS_Stereo_SC	Mean	0.395	21.233	37.992
	Median	0.294	12.270	24.477

From the presented metrics, we can observe the algorithms achieve a similar performance in the reconstructed DSMs. We show both mean and median in the results, as the mean provide information about the presence of outliers in the estimated heights and the median provides a more robust result. The best performing of the three selected algorithms is Stereo_SC, which is based on AANet. If we analyze e3m, Stereo_SC shows an error rate of 9.385%, which is 1.2% and 2.9% less than MVS_Full_SC and MVS_Stereo_SC respectively, containing less outliers. For the stricter e1m rate, Stereo_SC is again best, with differences of 0.2% and 2.3% in comparison to MVS_Full_SC and MVS_Stereo_SC respectively, showing that MVS_Full_SC has a competitive performance in this metric. With respect to the MAD metric, the results benefit the MVS

algorithms. This shows that MVS can achieve a more accurate result for a well matched pixel but the outliers are larger than in the stereo method for areas difficult to match.

In the Fig. 5.9 there is a visualization for the performance of the evaluated cases. In the upper row, the generated DSMs are compared along with the ground truth, while the lower row shows the absolute error map clipped to a threshold of 3 pixels. The RGB image helps to visualize the texture and geometry of the features to match.

As mentioned for the table analysis, the MVS methods present more outliers in areas difficult to match like the texture less areas in the rooftop and ground of the shown building. The Stereo_SC method has less error regions and performs better for the difficult areas. However, around the church domes, the Stereo_SC method is less accurate, especially around boundaries. It is also noticeable how the error regions vary smoothly in the stereo case, whereas for the MVS cases the values vary significantly from one pixel to another. Focusing only on the two MVS results, MVS_Full_SC is better than MVS_Stereo_SC, with a small difference in MAD but a better performance in e3m and e1m.

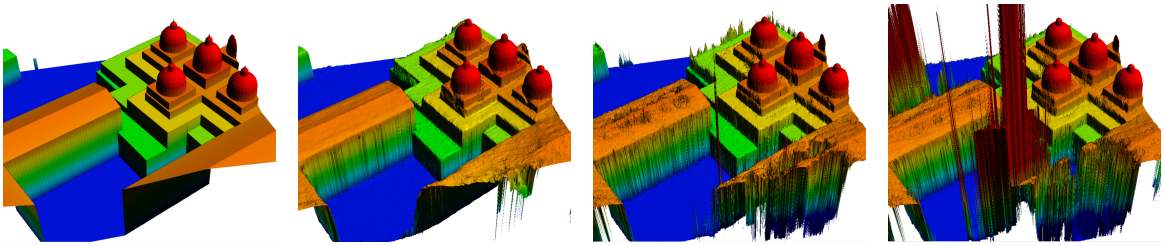


(a) DSM - Ground Truth **(b)** Stereo_SC **(c)** MVS_Full_SC **(d)** MVS_Stereo_SC

(e) Reference image **(f)** e1m-Stereo_SC **(g)** e1m-MVS_Full_SC **(h)** e1m-MVS_Stereo_SC

Figure 5.9: DSMs and error maps for a SyntCities sample. For the reference image (e) with ground truth (a), we show the DSMs computed by using the models Stereo_SC (b), MVS_Full_SC (c) and MVS_Stereo_SC (d). The respective 1m-error maps (e1m) for the same models are shown in (f), (g) and (h). Scale bars for the DSMs and error maps are given as a reference and use meters as unit. Errors are clipped to a maximum of 1m. Regions in black correspond to undefined pixels by the algorithms.

A 3D visualization of the computed DSMs is shown in Fig. 5.10 for the same area as Fig. 5.9. There we can observe how the Stereo_SC method produces smooth areas and the MVS cases suffer from outliers, especially MVS_Stereo_SC, where the values are not even similar to the height range of the scene.



(a) GT **(b)** Stereo_SC **(c)** MVS_Full_SC **(d)** MVS_Stereo_SC

Figure 5.10: SyntCities computed DSMs, 3D view. For the same perspective given for the ground truth (a), we show the results for the models Stereo_SC (b), MVS_Full_SC (c) and MVS_Stereo_SC(d). It covers the same area as the Fig. 5.9.

5.4.3 Results Dublin

For the experiments applied to the Dublin dataset, we show the obtained results in Table 5.2. We compare now the models Stereo_Du, MVS_Full_Du and MVS_Stereo_Du, which were finetuned with the Dublin dataset. As this dataset reflect the complexity of real-world scenes, the performance is lower than the one observed for SyntCities.

Again we observe the results to be in a similar range, demonstrating that all alternatives have reasonable capabilities for the 3D reconstruction. Nonetheless, there are differences to show which one performs best in real data. We observe here that in this case MVS_Full_Du is the leading algorithm followed by Stereo_Du and finally MVS_Stereo_Du. The change about Stereo not leading these results might come from the dataset configuration, as SyntCities was designed to work in a stereo matching framework, rendered already with epipolar geometry.

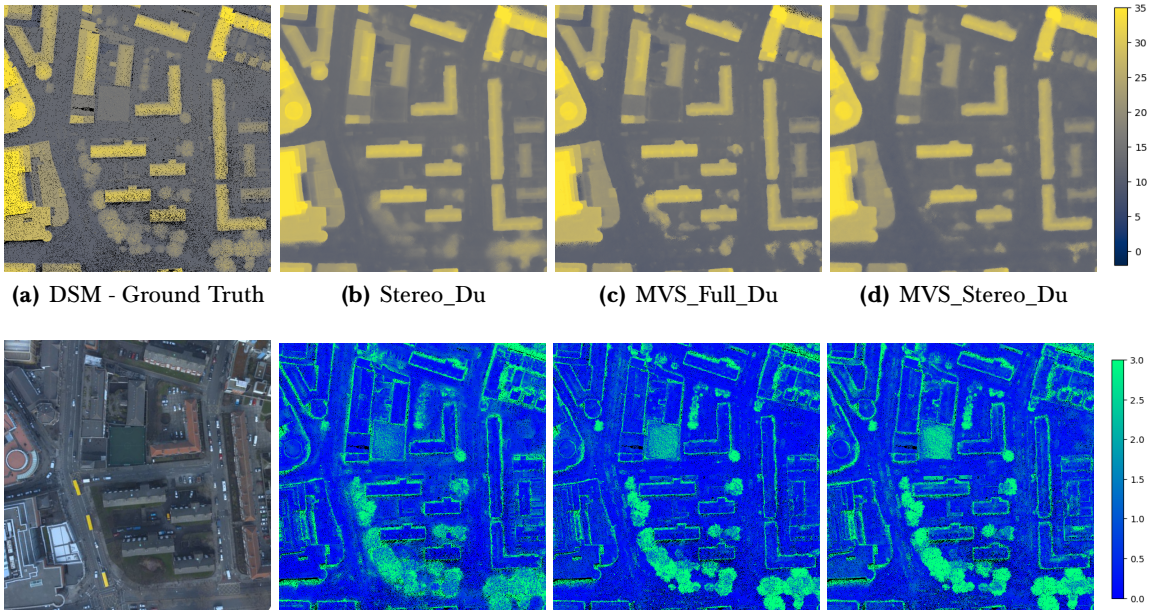
Table 5.2: DSM generation metrics, based on the fusion of stereo and MVS results for the Dublin dataset.

		Metrics		
Network	**Fusion**	**MAD (↓)**	**e3m (↓)**	**e1m (↓)**
Stereo_Du	Mean	2.49	47.06	72.68
	Median	0.56	15.18	36.76
MVS_Full_Du	Mean	0.60	13.97	35.51
	Median	0.55	13.26	33.25
MVS_Stereo_Du	Mean	1.1	21.20	54.27
	Median	0.75	15.52	42.31

For the e3m rate, MVS_Full_Du leads the table with an advantage of 1.92% and 2.26% over Stereo_Du and MVS_Stereo_Du respectively. A similar trend is observed for the stricter e1m rate, with improvements of 3.51% and 9.12%. The difference in the latter metric is high between both MVS solutions, showing MVS_Full_Du is better than MVS_Stereo_Du by a good margin. Although MVS_Full_Du is also better than Stereo_Du, the difference with respect to stereo is not large, especially for MAD. Focusing on MAD for the median of each algorithm, Stereo_Du and MVS_Full_Du have only a change of 0.01%, and 0.2% to MVS_Stereo_Du.

In Fig. 5.11 we show the results for the computed DSMs. The upper row includes the DSMs and the lower one the error maps, in this case with a threshold of 3m as the reconstruction is

less accurate than for the synthetic data. Still, we observe some similarities to the performance described for SyntCities. The quality around the edges is again better using the MVS algorithms as we can see for buildings and trees. Interestingly, for the trees themselves Stereo_Du achieves a better estimation, as for MVS these areas show errors larger than 3m.

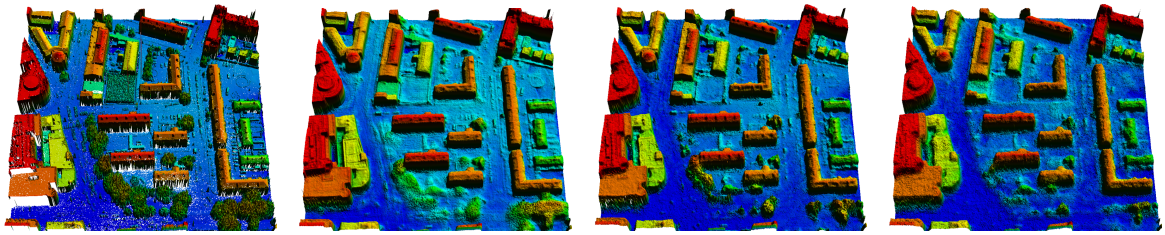


(a) DSM - Ground Truth **(b)** Stereo_Du **(c)** MVS_Full_Du **(d)** MVS_Stereo_Du

(e) Oblique close image **(f)** e3m-Stereo_Du **(g)** e3m-MVS_Full_Du **(h)** e3m-MVS_Stereo_Du

Figure 5.11: DSMs and error maps for a Dublin sample. For ground truth (a), we show the DSMs computed by using the models Stereo_Du (b), MVS_Full_Du (c) and MVS_Stereo_Du (d). The respective 1m-error maps(e1m) for the same models are shown in (f), (g) and (h). Scale bars in meters for the DSMs and error maps are given as a reference. Errors are clipped to a maximum of 3m. Regions in black correspond to undefined pixels by the algorithms. The corresponding orthorectified RGB is not shown, as this was not provided in the original dataset for this region. Instead, we show an oblique image captured close to this region in (e). This image is not aligned with the results.

A 3D visualization of the DSMs is displayed in Fig. 5.12. Rooftops are smoother and include less outliers in the Stereo_Du result. Besides, the vegetation is better represented as most of their surface is above ground level comparing with both MVS results. On the other hand, MVS_Stereo_Du and especially MVS_Full_Du compute a better estimation for pixels on the ground level, but they reduce significantly the expected surface for vegetation.



(a) GT **(b)** Stereo_Du **(c)** MVS_Full_Du **(d)** MVS_Stereo_Du

Figure 5.12: Dublin computed DSMs, 3D view. For the same perspective given for the ground truth (a), we show the results for the models Stereo_Du (b), MVS_Full_Du (c) and MVS_Stereo_Du(d). It covers the same area as the Fig. 5.11.

5.4.4 Results Confidence

In a separate section, we want to discuss the results of using the confidence values for the fusion method presented in 5.3.2. We evaluated the three DSM generation algorithms, namely Stereo_Du, MVS_Full_Du and MVS_Stereo_Du with the same approach, although LAFNet was designed only for stereo data and disparity maps. We studied only the case for the Dublin dataset, as it is more challenging and it has more candidate values for each pixel.

For each of the algorithms we analysed the following cases:

- Optimal: We select the best candidate for each pixel based on the difference with respect to the ground truth. Methods cannot achieve such accuracy, but we use it as a reference of the ideal best performance.
- Mean: We compute the mean of all candidate values to set the height of the pixels.
- Mean$N$: We remove the N% less confident values for each pixel and then we compute the mean. $N \in \{25, 50\}$
- Median: We compute the median of all candidate values to set the height of the pixels.
- Median$N$: We remove the N% less confident values for each pixel and then compute the median. $N \in \{25, 50\}$

Since the mean and the median without removal are the same algorithm as in the previous sections, these values are also found in table 5.2. Despite being the median more robust than the mean, we include both to give insights about the distribution of the candidate values.

With regard to the Stereo_Du case and the mean fusion, we observe that using the confidence values reduces significantly the presence of outliers. We see that for Mean25 and Mean50 the e3m rate drops to 18.33 and 15.33 respectively from the original 47.06. For the stricter e1m rate, the values drop to 43.03 and 38.69 instead of 72.68. This shows that large outliers were assigned a low confidence value. Considering the median values, the error rates decrease as well by approximately 2% in both e3m and e1m. By removing significant outliers from the distribution, the median of the remaining values gets closer to the ground truth. Hence, the confidence based fusion helps to refine the computed DSM for the stereo case.

Nevertheless, the confidence values do not seem to help in a similar manner the results from MVS_Full_Du and MVS_Stereo_Du. If we focus on the MVS_Full_Du case, we observe that the higher the percentage of removed pixels, the higher the error rate as well. Although the difference is very small ($\sim 1\%$), we note that there is no trend towards improvement. Addressing the MVS_Stereo_Du case, we notice for both mean and median a slightly better performance by using $rem_{\%} = 25$ in all metrics. By setting $rem_{\%} = 50$ the error rate is not decreasing. As LAFNet was developed for a distinct input data, many aspects should be taken into account to redesign the network to handle depth maps as well. Some of these aspects include:

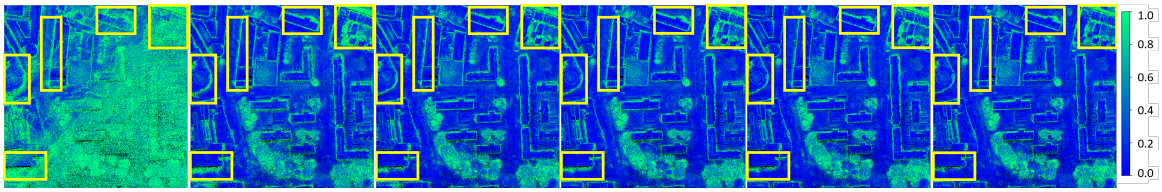
- Disparity maps and images are both in pixels and work in a 2D domain, while depth is meters and represents a 3D space, which is harder to correlate with the input images without the homography matrix information. Besides, depth and disparity ranges are inversely proportional and span different numerical ranges.

- Cost volumes used in UniMVSNet have a downsampling rate of ×4, which means the number of pixels is 1/16 of the original image size, missing details while upsampling the cost volume to be used by LAFNet. Nonetheless, the memory demands of the MVS algorithms limit the size of the cost volume to be computed.
- The learned features for the cost volumes vary from those for stereo matching. Especially for the MVS_Full_Du case, where many views are taken into account, the features for a reference image contain information from many additional views, where not all pixels are always visible. MVS_Stereo_Du seems to suffer less from this effect.
- MVS algorithms already make a fusion from different views based on the learned weights. Hence, the confidence might not be so discriminative to filter bad candidates in the estimated map.

The design of a new confidence network is out of our scope, but after studying the effect on the stereo data, we see potential to use the confidence based fusion as a strategy to create DSMs.

We show visually the results of the stereo case by using different $rem_{\%}$ rates. In Fig. 5.13 the images show the impact of the confidence based fusion. For the mean cases, we see a significant reduction of the error rate, particularly between no confidence guidance and Mean25, it also improves the fusion around edges for the Mean50 result. The median is more robust and as shown in (d) is less influenced by outliers. By using the confidence values, the fusion improves again mostly around building edges. As observed for the results of the stereo method, these areas are challenging for AANet, but with this guided fusion we can improve the accuracy of the computed DSM.

A 3D representation for the same area is shown in Fig. 5.14. Improvements are mostly in the edges of buildings (smoother in the median cases with confidence), less artifacts on the ground level (excluding cars). Regions highlighted in Fig. 5.13 can also be compared for the 3D representation to observe changes.

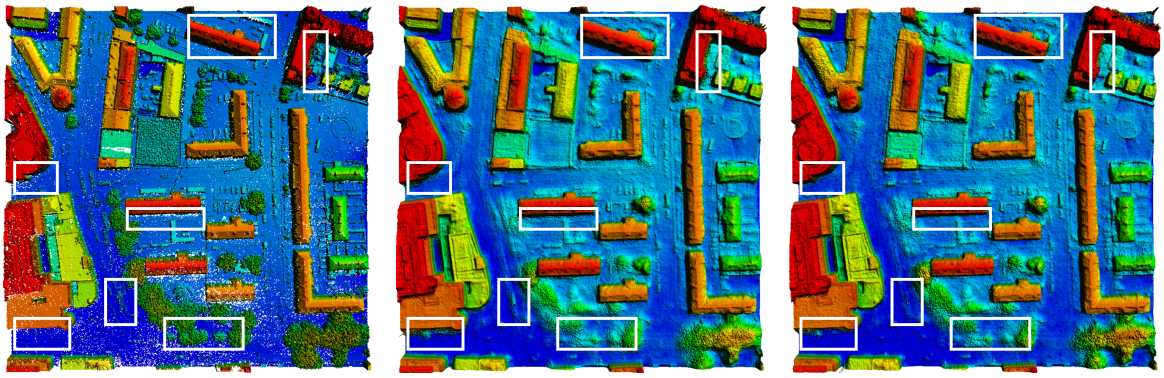


(a) Mean **(b)** Mean25 **(c)** Mean50 **(d)** Median **(e)** Median25 **(f)** Median50

Figure 5.13: Dublin DSMs created with confidence based fusion - Stereo case. We show cases for mean fusion without confidence (a), with $rem_{\%}$ =25 (b) and with $rem_{\%}$ = 50 (c). Similar cases are presented for the median in (d), (e) and (f). Scale bar for the error is given in meters. Yellow rectangles highlight areas with significant differences.

Table 5.3: DSM generation metrics, based on the fusion of stereo and MVS results for the Dublin dataset. In this case, the confidence was used for the fusion process.

		Metrics		
Network	**Fusion**	**MAD (↓)**	**e3m (↓)**	**e1m (↓)**
Stereo_Du	Optimal	0.06	4.00	10.47
	Mean	2.49	47.06	72.68
	Mean25	0.67	18.33	43.03
	Mean50	0.59	15.33	38.69
	Median	0.56	15.18	36.76
	Median25	0.53	14.57	34.88
	Median50	0.53	13.79	34.12
MVS_Full_Du	Optimal	0.14	6.04	14.82
	Mean	0.60	13.97	35.51
	Mean25	0.60	13.98	35.45
	Mean50	0.64	14.43	37.01
	Median	0.55	13.26	33.25
	Median25	0.57	13.40	33.98
	Median50	0.56	14.00	37.00
MVS_Stereo_Du	Optimal	0.09	1.89	6.58
	Mean	1.10	21.20	54.27
	Mean25	0.80	16.12	43.08
	Mean50	0.97	18.93	49.96
	Median	0.75	15.52	42.31
	Median25	0.75	15.48	41.77
	Median50	0.77	15.97	43.10



(a) Ground Truth **(b)** Median **(c)** Median50

Figure 5.14: Generated DSMs for a Dublin region in a 3D representation - Stereo case. Region is the same as for Fig. 5.13. We show three DSMs: ground truth, median fusion (no confidence based) and median fusion $rem_{\%} = 50$. Changes are highlighted in the white rectangles.

5.5 Discussion

We presented in this chapter a comparison between stereo and multi-view stereo (MVS) deep learning algorithms. From the presented results, we show how all solutions (Stereo, MVS_Full and MVS_Stereo) were able to compute a reliable DSM and preserving most of the geometric information. Stereo produces smoother results and is less prone to outliers, facing challenges in

areas adjacent to edges. On the other hand, MVS_Full and MVS_Stereo provide a better height estimation for those areas where the matching is not so challenging, but it also suffer from larger outliers where the matching fails, including textureless areas. We consider MVS_Full to be the most robust solution, also due to the low MAD values. Stereo also shows a good performance and benefits more from context information to compute a similar estimation for regions belonging to the same object, presenting errors mostly on edges instead. MVS_Stereo showed the lowest performance between the three approaches, leading to larger outliers and less accuracy for the strict e1m rate. Between the two basic fusion algorithms, we find median to be superior to the mean in all cases, so we do not recommend the mean fusion, especially for DSMs where the distributions are not normally distributed.

Regarding the confidence based fusion strategy we adopted, the results for the Stereo method showed an improvement, particularly for areas adjacent to the edges where the matching algorithm is prone to errors, compensating this flaw. However, the same method did not lead to more accurate DSMs for the MVS_Full and MVS_Stereo algorithms. We described some factors that could explain this issue, such as the discrepancies between depth and disparity maps, and the cost volumes sizes.

We additionally provide a processed version of the Dublin dataset to be applied in stereo and MVS algorithms, encouraging the community to continue the experiments in this direction or to easily apply the new architectures in the remote sensing field.

To inspire future work, we observed that the confidence based fusion lead to good results in the height maps estimated by the stereo algorithm. We would like to explore adaptations to the network to obtain also a good performance for the MVS cases. Additionally, a more sophisticated algorithm using the confidence values to fuse the DSM should be explored, not only the removal of bad pixels and the median of the remaining values. An architecture using both height and confidence maps as input for fusion could be an appealing research topic.

6

Evaluation of stereo and MVS algorithms for 3D reconstruction with paired data

Contents

This chapter describes an additional contribution from a conference article [154], which is a complementary research to chapter 5. Here, a previous study focusing on the generation of DSMs using only 2 views is considered and in this case only for synthetic data. Nonetheless, conventional and deep learning algorithms are included for this evaluation, providing useful insights of the differences between both. As the background and related work are similar for this and the previous chapter, these sections have been reduced to show only the information that has not been addressed yet.

6.1 Background

The research for 3D reconstruction has been a recurrent topic in the computer vision community. By having two or more images from the same scene, the task is to reconstruct a 3D representation of such scene based on the matching of corresponding points between the images. Most of the algorithms use either the stereo matching or the multi-view stereo (MVS) approach. For stereo matching, pairs of epipolar rectified images are given as input to compute a disparity map. Contrarily, MVS algorithms deal with two or more views and directly work in 3D space. A common strategy for computing the depth is the plane sweep algorithm, where a plane is swung in the 3D space in front of the camera and depth is computed at each location from the different views based on the 2D projections of such plane.

Lately, deep learning algorithms are leading in terms of accuracy and completeness. However, the stereo matching and MVS architectures have been studied separately due to differences in algorithms and input data. In addition, learning models require large amounts of data and ground truth, which is hard to acquire and the ground truth is often incomplete. Hence, using synthetic data is an option to evaluate the performance of the networks, as we can generate data in different formats and retrieve all the required parameter.

In this chapter, we present an evaluation of stereo and MVS deep learning algorithms applied to the same scenes. We train both algorithms in common datasets to set a fair comparison, for which the datasets have been properly adapted. We utilise the SyntCities dataset from our previous work [90], as this resembles remote sensing aerial imagery and provides all necessary input data for the selected algorithms and the SceneFlow [12] datasets, which have been widely used for training. Non-learning algorithms are considered as well for a comparable baseline. As accuracy is crucial in remote sensing applications, such as the generation of Digital Surface Models (DSMs), we evaluate the prediction error with a margin of 3 and 1 m.

Our main contributions are as follow:

- We prepared synthetic data to be compatible with stereo and MVS frameworks, setting similar training conditions.
- We trained different models and evaluated the performance in terms of the accuracy for the predicted depth.
- We study the effect of the baseline and occlusions in the depth predictions.

6.2 Related Work

In this section we describe some of the existing reconstruction algorithms as well as the related datasets, some of which are also used as benchmarks. Detailed differences between stereo and MVS frameworks are also discussed.

6.2.1 Stereo networks

In the conventional stereo matching algorithms, a cost volume is created for the disparity candidates and those disparities with the smallest cost are selected and refined for the final disparity map. A well known algorithm derived from this principle is Semi-Global Matching (SGM) [4] thanks to its trade-off between accuracy and computational cost. SGM computes the cost along different paths and penalizes large disparity changes. Similarly, More Global Matching (MGM) [159] takes into account more than one direction for the cost computation and achieves higher performance than SGM, with slightly more computational resources.

For the deep learning part, MC-CNN [11] was the first architecture used in the stereo matching and conceived only to replace the cost volume generation part, while the refinement was still conducted with no-learning algorithms such as SGM. Some end-to-end networks were designed to encompass the whole stereo pipeline and generate directly the disparity map as output like DispNet [12] and GC-Net. PSMNet [14] additionally introduced a spatial pyramid pooling module to collect information from different scales. GA-Net [17] incorporated layers which are a differentiable form of SGM and AANet [19] replaced 3D convolutions, reducing significantly the computational costs, inference times and with little impact on the accuracy.

For our experiments we selected GA-Net and AANet due to its accuracy and reduced computational cost respectively. They are also a common framework to compare with new architectures and both are based on a cost volume network.

6.2.2 Multi-view networks

Multi-view stereo algorithms take two or more views into account while estimating the depth of the objects in the scene. Normally, the views are sorted according to the camera position and orientation, so views close together are preferred as input for the algorithm. For a reference image, n-additional views are selected to estimate the depth map of such image. A known algorithm for MVS is COLMAP [156], that selects the views according to the geometric and photogrammetric information, and then computes the depth estimation through multi-view geometric consistency and further refinement.

In a similar way to stereo matching, deep learning has also achieved an outstanding performance for MVS. MVSNet [30] proposed to create a depth volume approach based on the plane sweep algorithm and its principles are the base for the development of newer architectures. CasMVSNet [32] improved the efficiency in terms of computational costs by using a coarse to fine scheme. In VisMVSNet [35] an additional uncertainty estimation is computed for the visibility of each pixel, including in that way the information related to the occlusions. Another case is UniMVSNet [36], where a coarse to fine scheme similar to CasMVSNet is enhanced by

a unified representation that deals with the prediction as a regression and a classification task simultaneously. UniMVSNet did not only show a very good performance, but can handle the computational resources efficiently.

Another two important cases are R-MVSNet [31] and PatchMatchnet [160], although these two are not based on a depth-volume strategy as the previous cases. R-MVSNet applies a regularization through a GRU network sequentially, reducing the memory requirements with a higher performance than MVSNet. PatchMatchNet follows an idea based on PatchMatch [29] similar to GIPUMA, leading to both good performance and efficient memory. In our analysis we decided to use UniMVSNet because of its accuracy and memory efficiency. Besides, it is based on a cost volume strategy as GANet and AANet.

6.2.3 Datasets

Deep learning strategies are not only known for their performance, but also for being data demanding. In the autonomous driving field for example, the KITTI 2012 [64] and KITTI [65] datasets are regularly not enough to train a neural network model because of their size and the incomplete ground truth. To help overcome this, synthetic data can be generated with thousands of samples and accurate ground truth. Hence, it is a common strategy to pre-train the model in a extensive synthetic dataset and later apply a fine-tuning stage to compensate for the domain gap. A notable example of synthetic data is the SceneFlow dataset, the main reference to train stereo networks. The dataset comprises more than 35k stereo pairs with corresponding ground truth and a large variety of shapes and textures.

In parallel, datasets have also been developed for the MVS architectures. The DTU dataset [75] made use of a robot arm to take pictures of small objects from different directions. Another remarkable case is the Tanks and Temples (T&T) dataset [76] with images taken from real indoors and outdoors environments, making the 3D reconstruction a challenging task. Both DTU and T&T are a common benchmark to evaluate the performance of MVS architectures. However, the ground truth is not accurate for all the pixels due to the sensor and scene properties. Same as for stereo matching, the synthetic data also represent a solution to train or at least pre-train the models. In this context, BlendedMVS [77] is a computer generated dataset with a large variety of textures, shapes and points of view that is compatible with MVS frameworks, being a common reference for training as SceneFlow is for stereo frameworks.

Still, available large datasets have a format not compatible for the two studied frameworks. Stereo datasets would require additional views from the same scene and the respective camera parameters to be used in a MVS algorithm. Contrariwise, MVS datasets would require epipolar rectification to be applied to a stereo algorithm, which might affect the quality of the ground truth due to the rectification process. Given this situation, we refer to the SyntCities dataset, as the stereo pairs include the camera parameters, enabling both stereo and MVS applications.

6.3 Methodology

In the present section we describe how the datasets have been processed to be compatible with the selected neural networks, as well as the series of experiments and considerations aiming to carry out a fair comparison of the algorithms.

6.3.1 Data preparation

As discussed above, available datasets in their current formats cannot be directly implemented in both stereo and MVS architectures. Therefore, we have selected only two cases, SceneFlow and SyntCities to be processed in a compatible format.

6.3.1.1 SceneFlow preparation

The images included in the SceneFlow dataset are already paired and fulfill the epipolar constraints. To apply them for a MVS algorithm we require to include the camera parameters, which can be derived from the information provided by the authors. Focal length, as well as the principal points and the baseline (defined as 1 in Blender units) are provided for all the images, which helps to create the intrinsic matrices. For the extrinsic matrices, we simplify the parameters to a basic position and rotation. Since the images are taken originally form a video sequence, two pairs of images do not show the exact same scene. Thus, the camera translation between frames is not relevant, as a full reconstruction from the scene is not even possible. For the rotation part, both left and right views can be assumed to come from a camera that has no rotations, as the camera planes are co-planar. Therefore, we can use as extrinsic matrices:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{6.1}$$

for the left and right images, respectively. To generate the depth ground truth, we compute the depth maps from the provided disparities with the formula:

$$z = f * b/d, \tag{6.2}$$

where z = depth, f = focal length, b = baseline and d = disparity. MVS approaches make use of a pre-defined depth range for each image, which is usually given by the sensor and acquisition conditions. For SceneFlow, we take the depth map values of each image and we set the depth range to 2th percentile as minimum and $\mu + \sigma$ as maximum, being μ and σ the mean and standard deviation respectively. This helps to focus on objects closer to the camera.

6.3.1.2 SyntCities preparation

SyntCities is a dataset to train stereo matching networks with patches resembling remote sensing scenes and under controlled simulated conditions. The samples are given for ground

sample distances (GSD) of 10cm, 30cm and 100cm and provided with training and testing subsets. Although not originally designed to work in a MVS framework, the camera parameters are available and samples along the same epipolar line can be used as the additional views. For the current article, we do not use the additional views simultaneously for the reconstruction, but we use the views to create diverse stereo pairs and study the effect of the baseline, which also implies differences in terms of occlusion.

In Figure 6.1 we can observe how the samples are selected for both the stereo and MVS implementation. Within the SyntCities dataset, many samples are rendered with the same conditions but different base height ratios (B/H), which helps us to study the effects of the baseline. By default, SyntCities images are given in pairs, which are represented for simplicity by the legends Baseline 1, Baseline 2 and Baseline 3. From there, we take the left sample from the largest baseline as a reference (R) and use the other images as additional views (V) for stereo pairing. The base height ratio determines the baseline b from the height h as b/h, where h =2000m for all cases. B/H values are 0.1, 0.2, 0.3, 0.4 and 0.5 (with baselines of 200m, 400m, 600m, 800m and 1000m respectively) for the Paris and Venice models. For a B/H = 0.5, the simulated camera resembles an acquisition field of view (FOV) around 28°. Images from the New York samples were not used, as these have a smaller baseline. In Figure 6.2, examples for Paris are given for a reference image with its respective 5 additional views. As expected, bigger changes in the images occur at larger baselines, which also implies larger occluded areas.

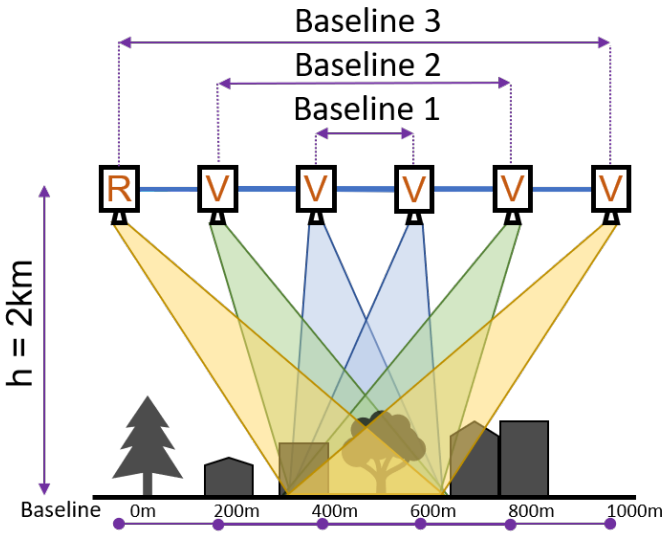
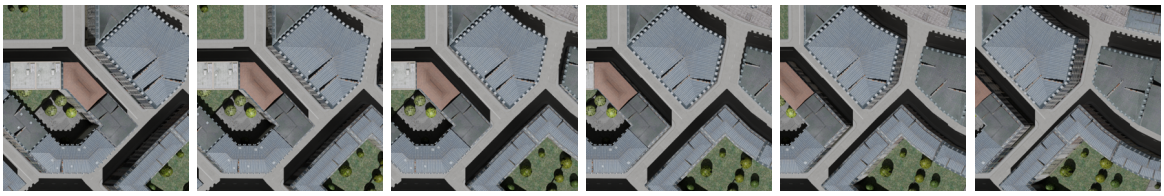


Figure 6.1: Selected geometry for SyntCities samples. All images lie on the same epipolar line with different baselines. For a reference view (R), 5 additional views (V) are available. Baseline distances are given for each view.

6.3.2 Conducted experiments

We utilized few well-known algorithms to test stereo pairs from SyntCities with different baselines. Both learn-based and traditional algorithms were considered. For the traditional part we selected SMG and MGM, as these are a common reference to compare other algorithms.



(a) Ref. Image **(b)** B/H=0.1 **(c)** B/H=0.2 **(d)** B/H=0.3 **(e)** B/H=0.4 **(f)** B/H=0.5

Figure 6.2: Examples of paired images from SyntCities along a common epipolar line. For the reference image (a), images with 5 different base height ratios (b-f) are given.

The used SGM implementation is the one in the CATENA pipeline [149] and for MGM we utilized the one provided by the author [i]. We used $P1 = 400$ and $P2 = 800$ with 16 directions and a Census-cost [5] for SGM. In the case of MGM, we used $P1 = 8$ and $P2 = 32$. Both SGM and MGM were given [-10, 192] as disparity range. Disparity maps are computed before and after applying the left-right consistency (LRC) check. Similar to the neural network results, the case before LRC check produces values for most of the pixels, so we used these results for the comparison. The results after LRC are also relevant, as these show the refinement effect.

We trained all the selected networks (GANet, AANet and UniMVSNet) on the SceneFlow dataset, as this is a common practice for stereo algorithms and it has a large pool of images. Testing, on the other hand, was done on SyntCities images. By avoiding training and testing on the same domain, we do not give additional advantage to the learning algorithms. We trained UniMVSNet with 2 views, so all models are based on the same training dataset with the paired images. GANet was trained for 27 epochs with a disparity range of [0,192] in 4 x GeForce RTX 2080 GPU. AANet was trained with the same conditions but 350 epochs, having a similar training time. UniMVSNet was trained for 16 epochs with 192 depth planes in 1x GeForce RTX 2080 GPU.

An important point to note here is to differentiate between the disparity and depth ranges. From the equation 6.2, we can see that disparity and depth are inversely related. The deep learning MVS frameworks already perform in the 3D space based on the plane sweep algorithm, where the planes hypotheses are uniformly distributed in the space of the camera. Contrarily, the stereo networks search for the disparity candidates in a uniform sampling, which is later non-uniform when the disparities are converted into depth values. This relation also discussed in detail in the CIDER [161] network.

Because of this non-linear relationship, stereo and MVS algorithms are affected by the distribution of the depth values in space. In the figure 6.3, such relationship is displayed for an image of the SceneFlow dataset with $f = 450$ and $b = 1$ for the disparity range [0, 192]. As we can see, the depth values are sparsely sampled for the low disparities and densely sampled for high disparities in stereo algorithms. We have adapted the depth ranges of the images to cover most of the content and alleviate the problem given by the depth - disparity range inconsistencies.

[i] https://github.com/gfacciol/mgm

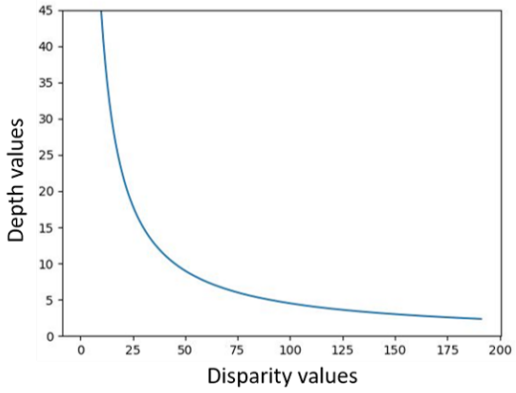


Figure 6.3: Non-linear relationship between disparity and depth values for an image of the SceneFlow dataset. The disparity range was set to [0, 192], which is common for many implementations.

6.4 Evaluation

To asses the results in terms of accuracy, completeness and effect of the baseline, we tested the algorithms:

- SGM: SGM result before LRC check.
- SGM w/LRC: SGM result after LRC check.
- MGM: MGM result before LRC check.
- MGM w/LRC: MGM result after LRC check.
- AANet: result of AANet converted to depth.
- GANet: result of GANet converted to depth.
- UniMVSNet: UniMVSNet result directly as depth map.

The first metric used to analyze the results is the Median Absolute Deviation (MAD), as this is a robust metric for skew distributions [108]. This is computed as:

$$\mathrm{MAD}_{\mathrm{diff}} = \mathrm{median}(|\ X_{\mathrm{diff}} - \widetilde{X}_{\mathrm{diff}}\ |), \tag{6.3}$$

where $\widetilde{X}_{\mathrm{diff}} = \mathrm{median}(X_{\mathrm{diff}})$, and $X_{\mathrm{diff}} = X - \bar{X}$, being X the ground truth, $\bar{X}$ the generated result and X_{diff} the difference between both.

Similarly to disparity maps evaluations, we also compared the error rate of the prediction but in this case oriented to the depth values. We computed the error rate 3 meters (e3m), which is the percentage of pixels where the prediction error is larger than 3 meters. Similarly, we compute the error rate 1 meter (e1m) following the same principle. The latter is critical for remote sensing, where accuracy within 1 meter is expected for applications such as DSM generation. The thresholds are based also on the influence of the disparity - depth relationship. We took an image with 600m baseline, its respective camera parameters and $d = 1, 2$. The corresponding depth values were $z = 1999.01, 1998.01$, having a difference of 1m. In any case, considering that the objects are located at 2000m from the camera, 1m error is a strict margin, so we also evaluate for 3m.

		With occluded pixels				**Non-occluded pixels**			
Method	B(m)	e1m(↓)	e3m(↓)	MAD(↓)	V.pix(↑)	e1m(↓)	e3m(↓)	MAD(↓)	V.pix(↑)
SGM	200	24.86	11.58	0.50	99.53	23.26	10.04	0.48	97.08
	400	22.18	14.33	0.28	98.63	15.79	9.25	0.25	89.31
	600	25.91	19.38	0.24	97.53	14.71	9.85	0.18	82.29
	800	30.77	24.76	**0.23**	96.76	15.19	10.52	0.16	76.39
	1000	35.52	29.62	0.27	95.83	16.14	11.40	**0.14**	70.82
MGM	200	23.67	**9.03**	0.49	99.76	21.86	7.20	0.48	97.05
	400	**21.90**	13.45	0.30	99.71	15.00	7.60	0.26	89.31
	600	28.23	21.53	0.27	99.67	14.97	9.06	0.20	82.31
	800	34.54	28.50	0.29	99.67	16.27	10.64	0.17	76.41
	1000	40.53	34.57	0.39	99.67	18.47	12.45	0.16	70.85
AANet	200	32.43	12.71	0.54	100.00	31.37	11.95	0.52	97.11
	400	30.93	14.88	0.43	100.00	25.57	10.28	0.37	89.36
	600	32.41	17.30	0.43	100.00	23.37	9.25	0.33	82.35
	800	34.27	20.25	0.45	100.00	22.57	10.08	0.31	76.45
	1000	38.18	23.62	0.55	100.00	23.65	10.67	0.31	70.88
GANet	200	36.04	12.32	0.68	100.00	34.80	11.22	0.66	97.11
	400	24.78	13.02	0.41	100.00	18.25	7.42	0.36	89.36
	600	24.95	15.87	0.36	100.00	13.81	**6.37**	0.28	82.35
	800	27.16	18.91	0.36	100.00	13.11	7.06	0.25	76.45
	1000	29.90	21.75	0.36	100.00	**13.07**	7.54	0.23	70.88
Uni-MVSNet	200	26.94	12.00	0.42	100.00	25.50	10.95	0.40	97.11
	400	26.52	14.21	0.35	100.00	20.09	9.14	0.31	89.36
	600	29.83	17.66	0.37	100.00	19.01	8.69	0.28	82.35
	800	34.52	21.87	0.43	100.00	20.36	9.72	0.29	76.45
	1000	39.52	26.80	0.55	100.00	21.87	10.82	0.29	70.88
SGM w/LRC	200	20.81	7.86	0.46	93.35	20.15	7.39	0.46	92.24
	400	12.86	6.51	0.23	85.52	10.81	5.37	0.23	82.45
	600	11.69	7.11	0.17	78.05	8.21	4.57	0.16	73.83
	800	11.64	7.65	0.14	71.16	6.73	3.59	0.13	66.12
	1000	12.31	8.73	0.13	65.00	6.18	3.43	0.11	59.42
MGM w/LRC	200	19.47	5.32	0.45	92.54	18.95	4.95	0.45	91.63
	400	10.82	4.06	0.24	82.81	9.52	3.41	0.23	80.70
	600	9.80	4.76	0.17	74.60	7.71	3.18	0.17	72.13
	800	11.02	6.65	0.15	67.85	7.33	3.40	0.14	64.39
	1000	13.08	8.82	0.13	61.63	7.63	3.72	0.12	57.22

Table 6.1: Experiments results for Paris and Venice images. MAD represents the Median Absolute Deviation, e3m the 3 meters error rate, e1m the 1 meter error rate and V.pix the percentage of pixels with a valid value generated by the algorithm. Underlined bold numbers show the best result (cases w/LRC excluded) for MAD, e3m and e1m. B stands for baseline.

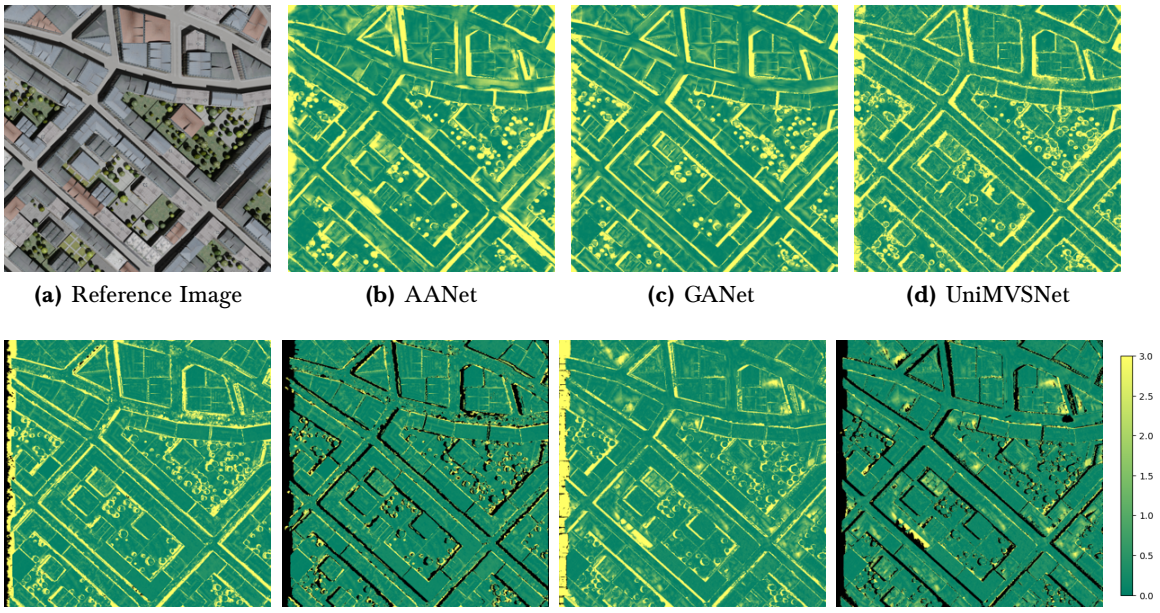
Completeness is also a desired feature for the reconstruction algorithms. Non-learning based approaches like SGM or MGM routinely refine the predicted disparity map with LRC to retrieve only the pixels where the disparities are more reliable and thus shortening the presence of outliers. However, this refinement might reduce significantly the density of the result, creating a lot of no defined regions in the disparity maps. Neural networks on the other hand estimate a value for each pixel in the image, but this allows the outliers to remain in the predicted disparity map. Hence, we report the percentage of pixels that were used for the metrics.

We also study the performance with and without occluded areas. As we have a dense ground truth for disparities, we also created LRC masks from them to identify the occluded areas.

Such masks apply to pixels that are only visible in one of the images. While it is expected that the algorithms cannot estimate the correct depth value in such areas, the prediction can still be satisfactory due to the neighbouring context. For instance, deep learning approaches gather contextual information to smoothly interpolate on the occluded areas. Besides, we want to observe how large is the error in the non-occluded areas, where this is assumed to be low.

We selected 20 images for our study from two virtual cities: 15 from Paris and 5 from Venice. For all the test images, we selected 30cm as GSD and 5 additional views with different baselines As the images of Paris and Venice have the same baselines, these are averaged for the metrics.

6.5 Results



(a) Reference Image **(b)** AANet **(c)** GANet **(d)** UniMVSNet

(e) SGM **(f)** SGM w/LRC **(g)** MGM **(h)** MGM w/LRC

Figure 6.4: Error maps for a Paris sample. For the reference image (a), we show the error maps for the algorithms AANet (b), GANet (c), UniMVSNet (d), SGM (e), SGM w/LRC (f), MGM(g) and MGM w/LRC (h). Scale bar for the errors given as a reference. Errors are clipped to a maximum of 3m. Regions in black correspond to undefined pixels by the algorithms.

From the set of experiments and metrics described above, we present all our results in Table 6.1. We split the SGM and MGM results depending on whether we used the LRC refinement or not, having the former at the end of the table. In all the remaining cases, the generated results covers most of the pixels. In the figure 6.4 we have visual results for the e3m metric in all tested algorithms for one of the tested images, which corresponds to a 200m baseline case.

We analyse first the results considering occluded areas. In terms of density, all cases (excepting those w/LRC refinement) achieve almost a complete depth map having at least 98% coverage. Looking at MAD, SGM and MGM have the best performance, although it is important to remember that some pixels have no defined values. For e3m the traditional methods perform better than the learning ones in the small baselines such as 200m, similar for 400m and worse

for 600m, 800m and 1000m. Hence, non-learning algorithms still outperform neural networks but only for small baselines. Nonetheless, non-learning algorithms have the advantage that no training time is required, just fine-tuning of the parameters to enhance the result. If we compare e1m the trend is similar, where good results for large baselines are given for learning algorithms. Interestingly, GANet is the approach with the best performance in this metric for baselines of 600m and above.

Taking into account the results w/LRC refienement (which applies only to SGM and MGM) we notice much better values in e3m, e1m and MAD. However, this also has a costly price as large sections of the images become undefined. For real applications on the other hand, it is helpful to have an algorithm that delivers only those areas where the estimation offers a good quality, so SGM and MGM are a valuable resource.

For the non-occluded results, in all cases the density is below 100% as expected and for SGM and MGM even lower, as some additional areas are discarded. MAD is better for SGM and MGM, while for neural networks GANet and AANet perform best and worst respectively. Considering e3m, MGM is the best for a small 200m baseline but GANet outperforms in all the other cases. AANet and UniMVSNet behave similarly. For the strict e1m, MGM achieves the best result for the small 200m and 400m baselines and GANet for the rest of the cases. In this part, we notice that UniMVSNet overcomes AANet for larger baselines.

Having analysed the dense results, we focus on the SGM w/LRC and MGM w/LRC cases. Accuracy is very high as can be seen from e3m and e1m values being mostly below 10%. This may be misleading as accuracy has increased while density has decreased. In fact, for the large baselines the depth maps cover less than 60% of the image.

If we look at the LRC check effect more in detail, we can notice that the removed pixels between the before and after results belong mostly to the occluded areas, thus dismissing efficiently the unreliable predictions. For instance, if we compare the case for the 1000m baseline, we notice that e1m for SGM goes from 16.14% to 6.18%, while the percentage of valid pixels goes from 70.82% to 59.42%, close to 10% for both values. A similar trend is observed for MGM, where e1m values for the 1000m baseline are reduced from 18.47% to 7.63% and the percentage of valid pixels from 70.85 to 57.22, having differences of 10.84% and 13.63% respectively. In figure 6.4, we can easily notice how most of the pixels with an error larger than 3m are removed by the LRC check, although these regions become undefined outputs.

Comparing only SGM and MGM we notice a similar performance, being MGM slightly better for small baselines and SGM for the large ones. Cases with or without occlusions, as well as with or without LRC check show a similar behaviour between these two methods. Fine-tuning of the penalty parameters P_1 and P_2 might improve the performance, but these are set empirically.

Between the two learning stereo methods, namely AANet and GANet, we notice how GANet has the best metrics for all cases except for the 200m baseline, where AANet leads for the e1m metric. In general, both show a competitive reconstruction result. In addition, the conversion from disparity to depth, which would represent sparsity in the depth space does not have a strong effect when compared to the UniMVSNet results, being even similar.

With regard to the main objective of this paper, we also study the performance differences between the stereo (GANet and AANet) and MVS (UniMVSNet) frameworks. The obtained results show that:

- In general, all cases have a comparable performance and are suitable for 3D reconstruction, as e3m considering occlusions are between 12% and 27% depending on the baseline
- Overall, for e3m the performance degrades when the baseline increases if occluded areas are also counted.
- If we focus only on the non-occluded areas, algorithms tend to perform best for e3m in intermediate baselines, while for e1m all but the smallest case have a similar performance.
- UniMVSNet is the best for all cases where the baseline is 200m, highlighting its focus on close range views.
- GANet is the best for e3m and e1m in all baselines except 200m, which shows the best accuracy and is particularly good for e1m in the non-occluded regions having a significantly difference with respect to the other algorithms. The matching itself of visible pixels performs the best in this case.
- The prediction for occluded areas in all learning approaches yields better results than the non-learning cases, which shows good capabilities to interpolate from the reliable pixels. Predicted depth maps tend to include smooth regions with sharp boundaries, specially if the baselines are not that large. Such interpolation effect is superior in the stereo networks as the e3m scores are lower.

Last but not least, we note the domain gap effect of training and testing in the different datasets. Due to such gap, the performance of the networks is not as high as it can be when it is fine-tuned in the same domain. It is of interest that the non-learning algorithms have a similar performance to the learning ones when the domain gap is present. Thus, for unseen data both options are a valuable resource.

6.6 Discussion

In the present chapter we conducted a lot of experiments to compare the performance of learning-based stereo and multi-view approaches on a similar setting. We noticed that stereo networks lead to a better reconstruction, especially GANet. Despite a slightly lower performance, MVS networks are also competitive and are even better for small baselines than stereo networks, but the accuracy drops for the large baselines.

We evaluated first considering also occluded areas in the stereo pairs to observe the robustness of the methods in this challenging regions, observing that the interpolation capabilities to predict these values works reasonably well and is slightly better for the stereo networks. In non-occluded areas we noticed a good performance for most of the cases, which shows that the matching itself is not an issue. Besides, we included non-learning algorithms in our study, which yielded good results but reduced the number of valid pixels in the predicted depth maps.

7

Conclusions and Future Work

The reconstruction of 3D urban areas is a challenging task with many aspects to analyse that help to enhance either the input data or the applied algorithms. These enhancements lead to significant accuracy improvements, some of which have been discussed in this dissertation. Creating and modifying datasets, comparing conventional and learning based solutions and evaluating stereo and MVS approaches are part of the discussed aspects.

7.1 Creation of synthetic data for stereo matching and comparison between traditional and learnable algorithms

A pipeline was proposed to simulate remote sensing data and generate thousands of samples required for a robust training. Starting from a 3D city model, it was possible to render optical images as stereo pairs with their respective depth and disparity maps. As all the geometrical definitions are included in the 3D software, the ground truth is very accurate and changes to acquire imagery with a different perspective are easy to implement.

Another benefit of the developed pipeline is the easy manipulation of the stereo array, as baselines can be increased or reduced to simulate different occlusion levels which is difficult to achieve in reality for a common area. Besides, illumination conditions can be modified to provide more diversity and include difficult areas with darker regions. Through our experiments, we observed that the generated SyntCities dataset is a feasible option to train stereo matching networks. Moreover, compared to other synthetic datasets, SyntCities helped to reduce the domain gap as sharper boundaries are estimated after training with this data. Testing in images even without finetuning produced good quality results in aerial and satellite imagery, which is useful in cases where the testing dataset is small or no ground truth for fine tuning is available.

However, we also noticed that the domain gap still played a significant role. Areas representing natural elements such as trees, crop fields, forests or water bodies are insufficient in the synthetic data and struggle to reconstruct the 3D shapes of the true objects. In addition to that, cities around the world offer a large variety of construction and architecture styles. As the SyntCities dataset was based on the distribution and design of European cities, its performance is lower when tested on images acquired in other regions.

Remote sensing images are also not always taken directly with a stereo array, but images from close acquisitions can also be used as input, as long as the overlapping is enough to allow the features matching. An extended case where the synthetic data is captured in a similar setting would be useful as well.

Since the release of the dataset, SyntCities has been used by the research community to conduct some experiments or as a reference to generate new datasets. In [162], the dataset was modified with an image-to-image translation network that helps to reduce the domain gap. SyntCities and US3D were used as input (the former as source and the latter as target domain) and the created samples showed different textures and illumination effects. Models trained on the new samples showed a better performance when tested on unseen US3D samples. The last two publications described in this work used SyntCities for the conducted experiments, too.

SyntCities was applied in this dissertation to study the performance of conventional and learnable algorithms for the stereo matching task. The conventional methods showed a good performance without any prior knowledge of the images and generated an accurate estimation, where the refinement steps effectively removed most of the outliers. This, however, significantly reduced the density of the data and the algorithms are not designed to compensate this by applying interpolation while estimating. Learnable algorithms on the other hand produce a dense result and values for all pixels. Their interpolation capabilities help to fill in the occluded and challenging areas with good performance but outliers of large occluded regions remain in the results. Yet, the metrics for all cases highlight better results of deep learning solutions if the domain gap is not significant and the predicted disparities recover even small and thin objects. Sophisticated designs such as GA-Net are able to get a higher accuracy (specially the deeper version of the architecture) than other networks like MC-CNN, AANet or PSMNet, but they do require more memory for training and longer inference times which hinders its usage for real time applications.

Depending on the final requirements, the balance between the accuracy and computational cost has to be selected to define which method should be applied. For images with rich texture, real time applications in a low memory system or tasks where the accuracy (not completeness) is the main criteria, conventional algorithms are sufficient. For a more complex case such as autonomous driving or matching of complicated areas, learnable algorithms are the best choice. This also applies for cases where the amount and quality of annotated data is sufficient.

7.2 Creation of synthetic data for urban change detection

The creation of the SMARS dataset is a relevant contribution from this thesis, too. It focused on change detection but can be used further for building and semantic segmentation. The applied pipeline starts from the design of the 3D city, where it was viable to manipulate the density of the building distribution and simulate the city growth process with demolitions and new constructions. The rendering framework helped to generate not only the optical imagery but accurate semantic labelling. For the 3D part, a photogrammetric algorithm was used, so that the provided DSM present blurry boundaries and smooth thin structures, as it is usual in the real DSMs. The experiments included in the respective publication showed that models training with SMARS performed well on real datasets like the Potsdam benchmark. Buildings were detected with sharp boundaries and the segmentation networks achieved good performance, with some difficulties to label vegetation or streets, particularly on cross-domain testing.

Nonetheless, the domain gap constraints were noticeable in the experiments. The main issue is the similar height of some categories in the available DSMs, such as buildings and trees, or streets and canals, which are mislabelled if only the DSMs are used as input. The generated 3D shapes, especially for trees, are simplified for the purpose of rendering. But canopies are very different and diverse in practice, so models trained with SMARS struggle with this point. Additionally, using labels that are distinct to other datasets limits the option of pre-training with SMARS and of finetuning with another dataset. Yet, this is a common problem that affects other datasets, as there is no uniform standard for labelling.

After releasing SMARS to be used for the community, some articles have already benefited from the dataset. The authors of [163] used it to train a multimodal co-learning framework that showed good performance on real WorldView-2 data for the building change detection task. A different case was presented in MLCNet [88], where a multi tasking network with an edge, binary and semantic ground truths is trained to produce a more robust change detection mask on three datasets including SMARS.

7.3 Study of stereo and MVS approaches for urban reconstruction

The last two publications mentioned in this thesis concern the 3D reconstruction from two main strategies: stereo matching and multi-view stereo. As these are usually handled separately in the literature, we studied the capability reconstructions from both.

In a first instance, we analysed the behaviour of conventional and deep learning solutions only in synthetic data, where the ground truth is very accurate and allows to observe a direct effect of the used baseline. Interestingly, conventional methods show a competitive result and as mentioned before, they can effectively discard a large percentage of bad estimations. Still, as these results are less dense and suffer from limited interpolation capabilities, the learnable method outperforms them. Considering only the learnable stereo and MVS (with only 2 input views) approaches, the former ones offer a better performance for large baselines and occluded regions, while the latter one is more efficient for close acquisitions.

After that, a more comprehensive study was conducted where multiple stereo pairs and MVS with 6 input views are compared. This study used two data sources: SyntCities for accurate evaluation with synthetic data and an enhanced version of the Dublin dataset [155] (adapted for stereo and MVS networks) as a real application case with challenging scenes. This setting is a study case that reconstructs the digital surface model of Dublin's downtown.

Based on the performance metrics, the result of using a MVS framework with multiple input views is the one that generates the most accurate DSM, followed by the multiple stereo pair cases. Using a direct stereo matching network performs leads to a better result than a MVS network with only 2 views as input, as the latter is prone to generate large outliers in occluded or textureless areas. On the other hand, MVS with many views is able to estimate a good height of ground pixels, define sharp boundaries and reduce the presence of outliers.

The impact of using the confidence to fuse the generated small DSMs into the final Dublin DSM was also analysed. The stereo case, which has already been investigated in this direction, offers some alternatives to measure the confidence. We used LAFNet to create a confidence map for each predicted disparity map and then applied this information to guide the DSM fusion. The DSM created in this way showed to be more accurate and helped to reduce the bad estimations around boundaries, a problem affecting rather the stereo networks than the MVS ones. Hence, the confidence based fusion proved to be a feasible option to refine the DSM. However, this strategy was not suitable for the MVS methods that have a different data nature and this still has to be explored.

Considering all the experiments conducted in this thesis, we observed a better performance for 3D reconstruction when using learnable algorithms. MVS with multiple inputs generates

the most accurate result, with a slight difference to the performance that stereo networks can achieve. Nevertheless, the performance of these algorithms highly depends on the amount and quality of the input data and is subject to domain gaps.

7.4 Future work

Although this thesis has explored some topics related to 3D reconstruction in detail, many aspects can be further studied to improve the quality of current methods. Some ideas to continue with the research in this field are listed below.

Regarding synthetic data:

- Extend the amount of cities in the synthetic datasets, with special focus on cities resembling other continents. This means, include larger variations in building height, rooftop styles, density of buildings and streets, settlements on hilly terrain, green areas, parking lots and industrial estate.
- Consider a more realistic representation of vegetation with a richer variety of textures and geometrical shapes for canopies and trunks.
- Model physical effects that are common during the acquisition, such as surface reflectivity, presence of clouds, scattering by aerosols or camera distortions.
- In the change detection case, adding buildings under construction process where the change is ambiguous, would be a more realistic framework.
- Using Venice as a reference model also sets a biased learning in SMARS, as many canals are present instead of streets. Further models that resemble the majority of the cities have to be included, especially cases with more height profiles.
- So far, SMARS addresses change detection only for the building class. A more general case where changes for streets, parks. etc., are resembled would be more challenging and advantageous to develop new algorithms.

Referring to the reconstruction algorithms:

- The comparison between stereo and MVS networks is still affected by the non-linear relation between disparity and depth. Developing a MVS framework that creates the depth planes with non-regular sampling would help to make a fairer comparison. Moreover, the captured objects might be reconstructed with higher accuracy, as more planes would focus on the depth region where such objects are located.
- The confidence estimation helped to refine the Dublin DSM but it required two networks: one to compute the disparity map and one for the confidence. A multi-task network that can estimate a reliable confidence map on top of the disparity map is a more robust solution, as the information from the whole matching cost volume would be available.

- The confidence guided fusion did not contribute to improve the DSM obtained by the MVS networks, although the applied confidence network was adapted to handle depth ranges. Still, it seems that a network has to be specifically designed for this data type, as not only the depth range is distinct when dealing with MVS networks but the matching features as well.
- Another way to benefit from the confidence values is to design a network that can learn to properly fuse disparity and confidence maps instead of the sorting process considered in this work. Learning the way to assign weights to the input data might create a more robust DSM.
- So far, this study targeted urban areas where dense man-made objects are present. Future cases should also contain complex natural elements such as mountains, water bodies, cliffs or dense forests should be analysed as well, because specific methods (conventional and learnable ones) may handle this type of remote sensing data more efficiently.
- New strategies are dominated by Neural Radiance Fields (NeRF), a technique that achieves a good quality 3D reconstruction and that can also create new views from the reconstructed objects. NeRF networks were not analysed in this work but studies to compare them with MVS and stereo is a pending research topic.

List of Abbreviations

B/A, B/H	Base to altitude/height ratio
BCE	Binary Cross Entropy
CGA	Computer-generated Architecture
CNN	Convolutional Neural Network
CRF	Conditional Random field
DEM	Digital Elevation Model
DL	Deep Learning
DSLR	Digital Single-Lens Reflex Camera
DSM	Digital Surface Model
DTM	Digital Terrain Model
FCN	Fully Connected Network
GRU	Gated Recurrent Units
GSD	Ground Sample Distance
GT	Ground Truth
IDW	Inverse Distance Weighting
LGA	Local Guided Aggregation
LiDAR	Light Detection and Ranging
LRCC	Left-Right Consistency Check
LSTM	Long short-term memory
MAD	Median Absolute Deviation
MVS	Multi-View Stereo
NeRF	Neural Radiance Fields
OSM	Open Street Map
S2P	Satellite Stereo Pipeline
SAR	Synthetic Aperture Radar

SC	SyntCities dataset
SF	SceneFlow dataset
SfM	Structure from Motion
SGA	Semiglobal Guided Aggregation
SGM	Semi-Global Matching
SMARS	Simulated Multimodal Aerial Remote Sensing Dataset
UAV	Unmanned Aerial Vehicle
UTM	Universal Transverse Mercator

List of Figures

List of Tables

Bibliography

[1] Carlo De Franchis, Enric Meinhardt-Llopis, Julien Michel, Jean-Michel Morel, and Gabriele Facciolo. An automatic and modular stereo pipeline for pushbroom images. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2014.

[2] R. Gupta and R.I. Hartley. Linear pushbroom cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):963–975, 1997.

[3] 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.

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

[5] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *European Conference on Computer Vision*, pages 151–158. Springer, 1994.

[6] Xing Mei, Xun Sun, Mingcai Zhou, Shaohui Jiao, Haitao Wang, and Xiaopeng Zhang. On building an accurate stereo matching system on graphics hardware. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 467–474. IEEE, 2011.

[7] Pablo d'Angelo. Improving semi-global matching: cost aggregation and confidence measure. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences-ISPRS Archives*, 41:299–304, 2016.

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

[9] Steven D. Cochran and Gérard G. Medioni. 3-D surface description from binocular stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(10):981–994, 1992.

[10] Xiao Xiang Zhu, Devis Tuia, Lichao Mou, Gui-Song Xia, Liangpei Zhang, Feng Xu, and Friedrich Fraundorfer. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):8–36, 2017.

[11] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17:1–32, 2016.

[12] Nikolaus Mayer, Eddy Ilg, Philip Hausser, 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016.

[13] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 66–75, 2017.

[14] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018.

[15] Xiao Song, Xu Zhao, Liangji Fang, Hanwen Hu, and Yizhou Yu. Edgestereo: An effective multi-task learning network for stereo matching and edge detection. *International Journal of Computer Vision*, 128(4):910–930, 2020.

[16] Xianzhi Du, Mostafa El-Khamy, and Jungwon Lee. AMNet: Deep atrous multiscale stereo disparity estimation networks. *arXiv preprint arXiv:1904.09099*, 2019.

[17] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. GA-Net: Guided aggregation net for end-to-end stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 185–194, 2019.

[18] Yuanxin Xia, Pablo d'Angelo, Friedrich Fraundorfer, Jiaojiao Tian, Mario Fuentes Reyes, and Peter Reinartz. GA-Net-Pyramid: An efficient end-to-end network for dense matching. *Remote Sensing*, 14(8), 2022.

[19] Haofei Xu and Juyong Zhang. AANet: Adaptive aggregation network for efficient stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1959–1968, 2020.

[20] Feihu Zhang, Xiaojuan Qi, Ruigang Yang, Victor Prisacariu, Benjamin Wah, and Philip Torr. Domain-invariant stereo matching networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 420–439. Springer, 2020.

[21] Lahav Lipson, Zachary Teed, and Jia Deng. RAFT-Stereo: Multilevel recurrent field transforms for stereo matching. In *2021 International Conference on 3D Vision*, pages 218–227, 2021.

[22] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[24] Zhaoshuo Li, Xingtong Liu, Nathan Drenkow, Andy Ding, Francis X. Creighton, Russell H. Taylor, and Mathias Unberath. Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6177–6186, 2021.

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

[26] R.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, 1996.

[27] David Gallup, Jan-Michael Frahm, Philippos Mordohai, Qingxiong Yang, and Marc Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[28] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 873–881, 2015.

[29] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), August 2009.

[30] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. MVSNet: Depth inference for unstructured Multi-view Stereo. *European Conference on Computer Vision (ECCV)*, 2018.

[31] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent MVSNet for high-resolution Multi-view Stereo depth inference. *Computer Vision and Pattern Recognition (CVPR)*, 2019.

[32] 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.

[33] Zehao Yu and Shenghua Gao. Fast-MVSnet: Sparse-to-dense multi-view stereo with learned propagation and Gauss-Newton refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1949–1958, 2020.

[34] Zizhuang Wei, Qingtian Zhu, Chen Min, Yisong Chen, and Guoping Wang. AA-RMVSNet: Adaptive aggregation recurrent Multi-view Stereo network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6187–6196, 2021.

[35] Jingyang Zhang, Yao Yao, Shiwei Li, Zixin Luo, and Tian Fang. Visibility-aware Multi-view Stereo network. *British Machine Vision Conference (BMVC)*, 2020.

[36] Rui Peng, Rongjie Wang, Zhenyu Wang, Yawen Lai, and Ronggang Wang. Rethinking depth estimation for Multi-View Stereo: A unified representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8645–8654, June 2022.

[37] Yikang Ding, Wentao Yuan, Qingtian Zhu, Haotian Zhang, Xiangyue Liu, Yuanjiang Wang, and Xiao Liu. TransMVSNet: Global context-aware Multi-view Stereo network with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8585–8594, 2022.

[38] Zhe Zhang, Rui Peng, Yuxi Hu, and Ronggang Wang. GeoMVSNet: Learning Multi-view Stereo with geometry perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21508–21518, 2023.

[39] Gangwei Xu, Xianqi Wang, Xiaohuan Ding, and Xin Yang. Iterative geometry encoding volume for stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21919–21928, June 2023.

[40] Saffet Erdogan. A comparision of interpolation methods for producing digital elevation models at the field scale. *Earth Surface Processes and Landforms*, 34(3):366–376, 2009.

[41] Pablo d'Angelo and Georg Kuschk. Dense Multi-view Stereo from satellite imagery. In *2012 IEEE International Geoscience and Remote Sensing Symposium*, pages 6944–6947, 2012.

[42] 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.

[43] J.P. Leitão and L.M. de Sousa. Towards the optimal fusion of high-resolution Digital Elevation Models for detailed urban flood assessment. *Journal of Hydrology*, 561:651–661, 2018.

[44] Danielle Hoja and Pablo d'Angelo. Analysis of DEM combination methods using high resolution optical stereo imagery and interferometric SAR data. In Christian Heipke, Karsten Jacobsen, Sönke Müller, and Uwe Sörgel, editors, *ISPRS Hannover Workshop 2009 High-Resolution Earth Imaging for Geospatial Information*, volume XXXVII of *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. ISPRS, 2009.

[45] Haris Papasaika, Effrosyni Kokiopoulou, Emmanuel Baltsavias, Konrad Schindler, and Daniel Kressner. Fusion of digital elevation models using sparse representations. In *Photogrammetric Image Analysis: ISPRS Conference, PIA 2011, Munich, Germany, October 5-7, 2011. Proceedings*, pages 171–184. Springer, 2011.

[46] Konrad Schindler, Haris Papasaika-Hanusch, Stefan Schütz, and Emmanuel Baltsavias. Improving wide-area DEMs through data fusion-chances and limits. In *Proceedings of the Photogrammetric Week*, volume 11, pages 159–170, 2011.

[47] Georg Kuschk, Pablo d'Angelo, David Gaudrie, Peter Reinartz, and Daniel Cremers. Spatially regularized fusion of multiresolution digital surface models. *IEEE Transactions on Geoscience and Remote Sensing*, 55(3):1477–1488, 2016.

[48] Howard Schultz, Edward M Riseman, Frank R Stolle, and Dong-Min Woo. Error detection and DEM fusion using self-consistency. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1174–1181. IEEE, 1999.

[49] H. Bagheri, M. Schmitt, and X. X. Zhu. Fusion of TanDEM-X and Cartosat-1 DEMS using TV-norm regularization and ANN-predicted weights. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3369–3372, 2017.

[50] Colleen E Fuss, Aaron A Berg, and John B Lindsay. DEM fusion using a modified k-means clustering algorithm. *International journal of digital earth*, 9(12):1242–1255, 2016.

[51] Rongjun Qin. Automated 3D recovery from very high resolution multi-view satellite images. *arXiv preprint arXiv:1905.07475*, 2019.

[52] Xiaoyan Hu and Philippos Mordohai. A quantitative evaluation of confidence measures for stereo vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2121–2133, 2012.

[53] Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Quantitative evaluation of confidence measures in a machine learning world. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[54] Matteo Poggi and Stefano Mattoccia. Learning from scratch a confidence measure. In *British Machine Vision Conference (BMVC)*, 2016.

[55] Matteo Poggi and Stefano Mattoccia. Learning to predict stereo reliability enforcing local consistency of confidence maps. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4541–4550, 2017.

[56] Sunok Kim, Dongbo Min, Seungryong Kim, and Kwanghoon Sohn. Unified confidence estimation networks for robust stereo matching. *IEEE Transactions on Image Processing*, 28(3):1299–1313, 2019.

[57] Sunok Kim, Seungryong Kim, Dongbo Min, and Kwanghoon Sohn. LAF-Net: Locally adaptive fusion networks for stereo confidence estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[58] Liyan Chen, Weihan Wang, and Philippos Mordohai. Learning the distribution of errors in stereo matching for joint disparity and uncertainty estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17235–17244, 2023.

[59] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3273–3282, 2019.

[60] 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: 36th German Conference, GCPR 2014, Münster, Germany, September 2-5, 2014, Proceedings 36*, pages 31–42. Springer, 2014.

[61] Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps using structured light. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 1, pages I–I. IEEE, 2003.

[62] Daniel Scharstein and Chris Pal. Learning conditional random fields for stereo. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[63] Heiko Hirschmuller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[64] 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*, pages 3354–3361, 2012.

[65] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *IEEE Conference on Computer Vision and Pattern Recognition 2015*, pages 3061–3070, 2015.

[66] Nikolaus Mayer, Eddy Ilg, Philipp Fischer, Caner Hazirbas, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. What makes good synthetic training data for learning disparity and optical flow estimation? *International Journal of Computer Vision*, 126:942–960, 2018.

[67] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, pages 611–625, 2012.

[68] Thomas Schops, Johannes L Schonberger, 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3260–3269, 2017.

[69] Peter Reinartz, Pablo d'Angelo, Thomas Krauß, Daniela Poli, Karsten Jacobsen, and Gurcan Buyuksalih. Benchmarking and quality analysis of DEM generated from high and very high resolution optical stereo satellite data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences; XXXVIII-Part 1*, 38(Part 1), 2010.

[70] Norbert Haala. The landscape of dense image matching algorithms. In *Photogrammetric week*, volume 13, pages 271–284, 2013.

[71] T. Wu, B. Vallet, M. Pierrot-Deseilligny, and E. Rupnik. A new stereo dense matching benchmark dataset for deep learning. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B2-2021:405–412, 2021.

[72] 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*, pages 1524–1532, 2019.

[73] Bertrand Le Saux, Naoto Yokoya, Ronny Hansch, Myron Brown, and Greg Hager. 2019 data fusion contest [technical committees]. *IEEE Geoscience and Remote Sensing Magazine*, 7(1):103–105, 2019.

[74] Shenhong Li, Sheng He, San Jiang, Wanshou Jiang, and Lin Zhang. WHU-Stereo: A challenging benchmark for stereo matching of high-resolution satellite images. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–14, 2023.

[75] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120:153–168, 2016.

[76] 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.

[77] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. BlendedMVS: A large-scale dataset for generalized Multi-view Stereo networks. *Computer Vision and Pattern Recognition (CVPR)*, 2020.

[78] Jin Liu and Shunping Ji. A novel recurrent encoder-decoder structure for large-scale Multi-View Stereo reconstruction from an open aerial dataset. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6049–6058, 2020.

[79] Hao Chen and Zhenwei Shi. A spatial-temporal attention-based method and a new dataset for remote sensing image change detection. *Remote Sensing*, 12(10):1662, 2020.

[80] Li Shen, Yao Lu, Hao Chen, Hao Wei, Donghai Xie, Jiabao Yue, Rui Chen, Shouye Lv, and Bitao Jiang. S2Looking: A satellite side-looking dataset for building change detection. *Remote Sensing*, 13(24):5094, 2021.

[81] Chenxiao Zhang, Peng Yue, Deodato Tapete, Liangcun Jiang, Boyi Shangguan, Li Huang, and Guangchao Liu. A deeply supervised image fusion network for change detection in high resolution bi-temporal remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166:183–200, 2020.

[82] Xiaokang Zhang, Weikang Yu, Man-On Pun, and Wenzhong Shi. Cross-domain landslide mapping from large-scale remote sensing images using prototype-guided domain-aware progressive representation learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 197:1–17, 2023.

[83] Fanjie Kong, Bohao Huang, Kyle Bradbury, and Jordan Malof. The Synthinel-1 dataset: a collection of high resolution synthetic overhead imagery for building segmentation. In *2020 Winter Conference on Applications of Computer Vision*, 2020.

[84] Xuan Li, Kunfeng Wang, Yonglin Tian, Lan Yan, Fang Deng, and Fei-Yue Wang. The ParallelEye dataset: A large collection of virtual images for traffic vision research. *IEEE Transactions on Intelligent Transportation Systems*, 20(6):2072–2084, 2019.

[85] Kai Tang and Jin Chen. ChangeAnywhere: Sample generation for remote sensing change detection via semantic latent diffusion model. *arXiv preprint arXiv:2404.08892*, 2024.

[86] Junshi Xia, Naoto Yokoya, Bruno Adriano, and Clifford Broni-Bediako. OpenEarthMap: A benchmark dataset for global high-resolution land cover mapping. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6254–6264, 2023.

[87] Michał Affek and Julian Szymański. A survey on the datasets and algorithms for satellite data applications. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024.

[88] Taoyuan Liu, Jiepan Li, Fangxiao Lu, Minghao Tang, and Guangyi Yang. MLCNet: Multi-task level-specific constraint network for building change detection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024.

[89] Mario Fuentes Reyes, Yuxing Xie, Xiangtian Yuan, Pablo d'Angelo, Franz Kurz, Daniele Cerra, and Jiaojiao Tian. A 2D/3D multimodal data simulation approach with applications on urban semantic segmentation, building extraction and change detection. *ISPRS Journal of Photogrammetry and Remote Sensing*, 205:74–97, 2023.

[90] Mario Fuentes Reyes, Pablo D'Angelo, and Friedrich Fraundorfer. SyntCities: A large synthetic remote sensing dataset for disparity estimation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:10087–10098, 2022.

[91] Sven Wanner and Bastian Goldluecke. Variational light field analysis for disparity estimation and super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):606–619, 2013.

[92] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3234–3243, 2016.

[93] Bertrand Le Saux, Naoto Yokoya, Ronny Haensch, and Myron Brown. 2019 IEEE GRSS data fusion contest: Large-scale semantic 3D reconstruction [technical committees]. *IEEE Geoscience and Remote Sensing Magazine*, 7(4):33–36, 2019.

[94] Saket Kunwar, Hongyu Chen, Manhui Lin, Hongyan Zhang, Pablo D'Angelo, Daniele Cerra, Seyed Majid Azimi, Myron Brown, Gregory Hager, Naoto Yokoya, Ronny Hänsch, and Bertrand Le Saux. Large-scale semantic 3-D reconstruction: Outcome of the 2019 IEEE GRSS data fusion contest—Part A. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:922–935, 2021.

[95] Yanchao Lian, Tuo Feng, Jinliu Zhou, Meixia Jia, Aijin Li, Zhaoyang Wu, Licheng Jiao, Myron Brown, Gregory Hager, Naoto Yokoya, Ronny Hänsch, and Bertrand Le Saux. Large-scale semantic 3-D reconstruction: Outcome of the 2019 IEEE GRSS data fusion contest—Part B. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:1158–1170, 2021.

[96] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. SegStereo: Exploiting semantic information for disparity estimation. In *Proceedings of the European Conference on Computer Vision*, pages 636–651, 2018.

[97] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1841–1850, 2019.

[98] Junming Zhang, Katherine A Skinner, Ram Vasudevan, and Matthew Johnson-Roberson. DispSegNet: Leveraging semantics for end-to-end learning of disparity estimation from stereo imagery. *IEEE Robotics and Automation Letters*, 4(2):1162–1169, 2019.

[99] Pier Luigi Dovesi, Matteo Poggi, Lorenzo Andraghetti, Miquel Martí, Hedvig Kjellström, Alessandro Pieropan, and Stefano Mattoccia. Real-time semantic stereo matching. In *2020 IEEE International Conference on Robotics and Automation*, pages 10780–10787, 2020.

[100] Zhenyao Wu, Xinyi Wu, Xiaoping Zhang, Song Wang, and Lili Ju. Semantic stereo matching with pyramid cost volumes. In *2019 IEEE/CVF International Conference on Computer Vision*, pages 7483–7492, 2019.

[101] Qin Wang, Dengxin Dai, Lukas Hoyer, Luc Van Gool, and Olga Fink. Domain adaptive semantic segmentation with self-supervised depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[102] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Dmitry Olefir, Tomas Hodan, Youssef Zidan, Mohamad Elbadrawy, Markus Knauer, Harinandan Katam, and Ahsan Lodhi. Blenderproc: Reducing the reality gap with photorealistic rendering. In *International Conference on Robotics: Sciene and Systems, RSS 2020*, 2020.

[103] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.

[104] Franz Kurz, D. Rosenbaum, Oliver Meynberg, G. Mattyus, and Peter Reinartz. Performance of a real-time sensor and processing system on a helicopter. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-1:189–193, 11 2014.

[105] Chen Wang, Xiao Bai, Xiang Wang, Xianglong Liu, Jun Zhou, Xinyu Wu, Hongdong Li, and Dacheng Tao. Self-supervised multiscale adversarial regression network for stereo disparity estimation. *IEEE Transactions on Cybernetics*, 51(10):4770–4783, 2020.

[106] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 764–773, 2017.

[107] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable ConvNets V2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019.

[108] Joachim Höhle and Michael Höhle. Accuracy assessment of digital elevation models by means of robust statistical methods. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(4):398–406, 2009.

[109] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[110] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[111] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016.

[112] Xiao Xiang Zhu, Devis Tuia, Lichao Mou, Gui-Song Xia, Liangpei Zhang, Feng Xu, and Friedrich Fraundorfer. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):8–36, 2017.

[113] Yuxing Xie, Jiaojiao Tian, and Xiao Xiang Zhu. Linking points with labels in 3D: A review of point cloud semantic segmentation. *IEEE Geoscience and Remote Sensing Magazine*, 8(4):38–59, 2020.

[114] Hamidreza Hosseinpour, Farhad Samadzadegan, and Farzaneh Dadrass Javan. CMGFNet: A deep cross-modal gated fusion network for building extraction from very high-resolution remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 184:96–115, 2022.

[115] Pedram Ghamisi, Bernhard Höfle, and Xiao Xiang Zhu. Hyperspectral and LiDAR data fusion using extinction profiles and deep convolutional neural network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(6):3011–3024, 2016.

[116] Jiaojiao Tian, Shiyong Cui, and Peter Reinartz. Building change detection based on satellite stereo imagery and digital surface models. *IEEE Transactions on Geoscience and Remote Sensing*, 52(1):406–417, 2013.

[117] Rongjun Qin, Jiaojiao Tian, and Peter Reinartz. 3D change detection–approaches and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 122:41–56, 2016.

[118] Richard J Chen, Ming Y Lu, Tiffany Y Chen, Drew FK Williamson, and Faisal Mahmood. Synthetic data in machine learning for medicine and healthcare. *Nature Biomedical Engineering*, 5(6):493–497, 2021.

[119] Daniel Stoecklein, Kin Gwn Lore, Michael Davies, Soumik Sarkar, and Baskar Ganapathysubramanian. Deep learning for flow sculpting: Insights into efficient learning using scientific simulation data. *Scientific reports*, 7(1):1–11, 2017.

[120] Han Li, Zhe Wang, and Tianzhen Hong. A synthetic building operation dataset. *Scientific data*, 8(1):1–13, 2021.

[121] Yuxing Xie, Jiaojiao Tian, and Xiao Xiang Zhu. A co-learning method to utilize optical images and photogrammetric point clouds for building extraction. *International Journal of Applied Earth Observation and Geoinformation*, 116:103165, 2023.

[122] R. Caye Daudt, B. Le Saux, A. Boulch, and Y. Gousseau. Urban change detection for multispectral Earth observation using convolutional neural networks. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 2115–2118, July 2018.

[123] Ritwik Gupta, Bryce Goodman, Nirav Patel, Ricky Hosfelt, Sandra Sajeev, Eric Heim, Jigar Doshi, Keane Lucas, Howie Choset, and Matthew Gaston. Creating xBD: A dataset for assessing building damage from satellite imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition workshops*, pages 10–17, 2019.

[124] Rodrigo Caye Daudt, Bertrand Le Saux, Alexandre Boulch, and Yann Gousseau. Multitask learning for large-scale semantic change detection. *Computer Vision and Image Understanding*, 187:102783, 2019.

[125] Ruizhe Shao, Chun Du, Hao Chen, and Jun Li. SUNet: Change detection for heterogeneous remote sensing images from satellite and UAV using a dual-channel fully convolution network. *Remote Sensing*, 13(18):3750, 2021.

[126] V Coletta, V Marsocci, and R Ravanelli. 3DCD: a new dataset for 2D and 3D change detection using deep learning techniques. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B3-2022:1349–1354, 2022.

[127] Valerio Marsocci, Virginia Coletta, Roberta Ravanelli, Simone Scardapane, and Mattia Crespi. Inferring 3D change detection from bitemporal optical images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 196:325–339, 2023.

[128] Anko Börner, Lorenz Wiest, Peter Keller, Ralf Reulke, Rolf Richter, Michael Schaepman, and Daniel Schläpfer. SENSOR: a tool for the simulation of hyperspectral remote sensing systems. *ISPRS Journal of Photogrammetry and Remote Sensing*, 55(5-6):299–312, 2001.

[129] Junyi Tao, Stefan Auer, Gintautas Palubinskas, Peter Reinartz, and Richard Bamler. Automatic SAR simulation technique for object identification in complex urban scenarios. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(3):994–1003, 2013.

[130] Xiaowen Li and Alan H. Strahler. Geometric-optical modeling of a conifer forest canopy. *IEEE Transactions on Geoscience and Remote Sensing*, GE-23(5):705–721, 1985.

[131] Wenzhong Shi, Min Zhang, Rui Zhang, Shanxiong Chen, and Zhao Zhan. Change detection based on artificial intelligence: State-of-the-art and challenges. *Remote Sensing*, 12(10):1688, 2020.

[132] John RG Townshend, Christopher O Justice, Charlotte Gurney, and James McManus. The impact of misregistration on change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 30(5):1054–1060, 1992.

[133] Abdullah Almutairi and Timothy A Warner. Change detection accuracy and image properties: a study using simulated data. *Remote Sensing*, 2(6):1508–1529, 2010.

[134] Iris de Gélis, Sébastien Lefèvre, and Thomas Corpetti. Change detection in urban point clouds: An experimental comparison with simulated 3D datasets. *Remote Sensing*, 13(13):2629, 2021.

[135] Thorsten Hoeser and Claudia Kuenzer. SyntEO: Synthetic dataset generation for earth observation and deep learning–demonstrated for offshore wind farm detection. *ISPRS Journal of Photogrammetry and Remote Sensing*, 189:163–184, 2022.

[136] Jianbo Qi, Donghui Xie, Tiangang Yin, Guangjian Yan, Jean-Philippe Gastellu-Etchegorry, Linyuan Li, Wuming Zhang, Xihan Mu, and Leslie K. Norford. LESS: LargE-Scale remote sensing data and image simulation framework over heterogeneous 3D scenes. *Remote Sensing of Environment*, 221:695–706, 2019.

[137] M Disney, P Lewis, and P Saich. 3D modelling of forest canopy structure for remote sensing simulations in the optical and microwave domains. *Remote Sensing of Environment*, 100(1):114–132, 2006.

[138] Jean-Philippe Gastellu-Etchegorry, Tiangang Yin, Nicolas Lauret, Thomas Cajgfinger, Tristan Gregoire, Eloi Grau, Jean-Baptiste Feret, Maïlys Lopes, Jordan Guilleux, Gérard Dedieu, et al. Discrete anisotropic radiative transfer (DART 5) for modeling airborne and satellite spectroradiometer and LIDAR acquisitions of natural and urban landscapes. *Remote Sensing*, 7(2):1667–1701, 2015.

[139] Růžena Janoutová, Lucie Homolová, Zbyněk Malenovskỳ, Jan Hanuš, Nicolas Lauret, and Jean-Philippe Gastellu-Etchegorry. Influence of 3D spruce tree representation on accuracy of airborne and satellite forest reflectance simulated in DART. *Forests*, 10(3):292, 2019.

[140] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 102–118. Springer, 2016.

[141] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. AirSim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics: Results of the 11th International Conference*, pages 621–635. Springer, 2018.

[142] Matteo Fabbri, Guillem Brasó, Gianluca Maugeri, Orcun Cetintas, Riccardo Gasparini, Aljoša Ošep, Simone Calderara, Laura Leal-Taixé, and Rita Cucchiara. MOTSynth: How can synthetic data help pedestrian detection and tracking? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10849–10859, 2021.

[143] Jong Won Ma, Thomas Czerniawski, and Fernanda Leite. Semantic segmentation of point clouds of building interiors with deep learning: Augmenting training datasets with synthetic BIM-based point clouds. *Automation in Construction*, 113:103144, 2020.

[144] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10912–10922, 2021.

[145] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16. PMLR, 2017.

[146] Lyujie Chen, Feng Liu, Yan Zhao, Wufan Wang, Xiaming Yuan, and Jihong Zhu. VALID: A comprehensive virtual aerial image dataset. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2009–2016, 2020.

[147] Meida Chen, Qingyong Hu, Zifan Yu, Hugues Thomas, Andrew Feng, Yu Hou, Kyle McCullough, Fengbo Ren, and Lucio Soibelman. STPLS3D: A large-scale synthetic and real aerial photogrammetry 3D point cloud dataset. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022.

[148] Aoran Xiao, Jiaxing Huang, Dayan Guan, Fangneng Zhan, and Shijian Lu. Transfer learning from synthetic to real LiDAR point cloud for semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2795–2803, 2022.

[149] Thomas Krauß. Six years operational processing of satellite data using CATENA at DLR: Experiences and recommendations. *KN-Journal of Cartography and Geographic Information*, 64(2):74–80, 2014.

[150] P. d'Angelo and P. Reinartz. Semiglobal matching results on the ISPRS stereo matching benchmark. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVIII-4/W19:79–84, 2011.

[151] Patrick M Bartier and C Peter Keller. Multivariate interpolation to incorporate thematic surface data using inverse distance weighting (IDW). *Computers & Geosciences*, 22(7):795–799, 1996.

[152] Franz Kurz, Sebastian Türmer, Oliver Meynberg, Dominik Rosenbaum, Hartmut Runge, Peter Reinartz, and Jens Leitloff. Low-cost optical camera systems for real-time mapping applications. *Photogrammetrie-Fernerkundung-Geoinformation*, pages 159–176, 2012.

[153] GDAL/OGR contributors. *GDAL/OGR Geospatial Data Abstraction software Library*. Open Source Geospatial Foundation, 2022.

[154] M. Fuentes Reyes, P. d'Angelo, and F. Fraundorfer. An evaluation of stereo and multiview algorithms for 3D reconstruction with synthetic data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-1/W2-2023:1021–1028, 2023.

[155] Debra F. Laefer, Saleh Abuwarda, Anh-Vu Vo, Linh Truong-Hong, and Hamid Gharibi. 2015 aerial laser and photogrammetry survey of Dublin City collection record, 06 2017.

[156] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured Multi-View Stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[157] Akihito Seki and Marc Pollefeys. Patch based confidence prediction for dense disparity map. In *British Machine Vision Conference (BMVC)*, 2016.

[158] Andrea Fusiello, Emanuele Trucco, and Alessandro Verri. A compact algorithm for rectification of stereo pairs. *Machine vision and applications*, 12:16–22, 2000.

[159] Gabriele Facciolo, Carlo de Franchis, and Enric Meinhardt. MGM: A significantly more global matching for stereovision. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 90.1–90.12, September 2015.

[160] F. Wang, S. Galliani, C. Vogel, P. Speciale, and M. Pollefeys. PatchmatchNet: Learned Multi-View patchmatch stereo. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14189–14198, jun 2021.

[161] Qingshan Xu and Wenbing Tao. Learning inverse depth regression for multi-view stereo with correlation cost volume. In *AAAI Conference on Artificial Intelligence*, volume 34, pages 12508–12515, 2020.

[162] Vasudha Venkatesan, Daniel Panangian, Mario Fuentes Reyes, and Ksenia Bittner. Syntstereo2real: Edge-aware gan for remote sensing image-to-image translation while maintaining stereo constraint. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 512–521, June 2024.

[163] Yuxing Xie, Xiangtian Yuan, Xiao Xiang Zhu, and Jiaojiao Tian. Multimodal co-learning for building change detection: A domain adaptation framework using vhr images and digital surface models. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–20, 2024.

A

Appendix

The publications associated with this dissertation involve the release of three datasets. Basic information on how to download and use these datasets is briefly described here.

A.1 SyntCities dataset

The SyntCities dataset [90] is stored in Zenodo and can be downloaded from the site: `https://zenodo.org/records/6967325`. The files are stored according to the following structure:

```
City
└─ City_pC_bhr_0_W_h2km_rXcm_Ye_Za/
   ├─ cameras_view/
   │  └─ N_cam.txt
   ├─ depth_view/
   │  └─ training/ or test/
   │     └─ N_depth_view.tif
   ├─ disparity_view/
   │  └─ training/ or test/
   │     └─ N_disparity_view.tif
   ├─ view_RGB/
   │  └─ training/ or test/
   │     └─ N_view_RGB.png
   ├─ LRC_mask/
   │  └─ training/ or test/
   │     └─ N_lrc_mask.png
   ├─ segmap_view/
   │  └─ training/ or test/
   │     └─ N_segmap_view.png
   └─ pair.txt
```

Notes:

- `City` can be the simulated New-York, Paris or Venice model
- `pC` refers to pivot Central, which is the way the stereo rig has been set within Blender
- `bhr` refers to baseline-to-height ratio. Parameters are related as baseline = bhr · height, where bhr= W/100, being W the value in the folder name

- `h2kn` describes a height of 2km in the simulation models for the camera location
- `X` defines the ground sample distance for points located at 2km
- `Y` and `Z` are the simulated sun elevation and azimuth
- camera files are not in the training/test subfolders, but one per sample and per view is provided
- `N` is used to represent a sample between 0-24
- `view` can be left or right

A.1.1 File formats

- Camera parameters are stored in .txt files with extrinsic and intrinsic matrices, depth information is also included.
- Depth images are stored as TIF files with Int32 accuracy to reduce memory. Values are stored in m. To obtain real values, apply: `depth = depth_stored / 100`
- Disparity images are also stored in TIF files with Int16 accuracy to reduce memory. To obtain real values, apply: `disparity = disparity_stored / 32`
- Left_RGB, right_RGB, segmap_left and segmap_right are all PNG files with three channels
- LRC masks are stored as PNG files with one channel and only binary values.

A.1.2 Camera parameters

The extrinsic matrix includes both the rotation matrix and the translation vector:

$$R = \begin{bmatrix} r_1 & r_2 & r_3 & 0 \\ r_4 & r_5 & r_6 & 0 \\ r_7 & r_8 & r_9 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T = \begin{bmatrix} t_x \\ t_y \\ t_z \\ 1 \end{bmatrix}$$

as $E = [R|t]$ to convert from the 3D model coordinate system to the camera coordinates. The values for the translation are given in m and rotations in radians. For a point $P = [X, Y, Z]$ in the 3D model coordinate system, its rotation and translation w.r.t. to the camera position coordinates $[X_c, Y_c, Z_c]$ is computed as:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

To convert then the 3D point in the camera reference system to the 2D image plane, we use the intrinsics matrix which includes the focal length (f_x, f_y) and the principal points (p_x, p_y) described in pixels. The conversion from $[X_c, Y_c, Z_c]$ to $[x, y]$ is given as:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

$$x = x'/z', \quad y = y'/z'$$

NOTES: The image coordinate system is the same as COLMAP, where X points to the right, Y to the bottom and Z to the front of the image. This is a right-hand system.

Parameters in the last row of the camera file describe the depth in terms of m as:
`[MIN_DEPTH INTERVAL_DEPTH NUMBER_INTERVALS MAX_DEPTH]`

And these can be related as:
`MAX_DEPTH = MIN_DEPTH + (INTERVAL_DEPTH * NUMBER_INTERVALS)`

A.1.3 Categories

Class	RGB value	Class	RGB value
Background	41, 120, 142	Roof (flat)	253, 231, 36
Trees	68, 1, 84	Facades	32, 144, 40
Street	72, 35, 116	Garden	64, 67, 135
Roof (mansard)	189, 222, 38	Piazza	30, 157, 136
Roof (gambrel)	68, 190, 112	Underwalls	53, 183, 120
Roof (gable)	34, 167, 132	Cars	73, 223, 120
Roof (hip)	121, 209, 81		

A.1.4 Overlapping

In the cases where the models were large enough, the overlapping was avoided to provide different scenes. However, there are some cases where this did not happen. The following table (left) shows the existing overlapping for adjacent samples in terms of pixels. The same value applies in both vertical and horizontal directions. Samples within each subset follow the spatial distribution to identify adjacent samples (based on the index) as shown in the table (right).

City/GSD	1m	30cm	10cm
New York	896	512	0
Paris	768	0	0
Venice	512	0	0

Training				Test
4	9	14	19	24
3	8	13	18	23
2	7	12	17	22
1	6	11	16	20
0	5	10	15	20

Note: Samples included in the 'training' subsets do not overlap with those in the 'test' subsets.

A.1.5 Usage for MVS

In each folder there is a pair.txt file which follows the same structure as that used by MVS networks (like MVSNet). Due to the overlapping differences, we suggest to use only those subsets where overlapping exists. The pairs are specified within the same view (left or right) but can be also complemented with the other stereo views.

Additionally, keeping the same values for GSD allows to use the samples with a different `bhr` to add even more views. These samples actually share a lot of content. It is recommended to use the same illumination conditions between samples. A case where patches share more similarities can be created manually by giving other images with the same resolution and sample number but a different `bhr` value as input.

A.2 SMARS dataset

The SMARS dataset [89] is stored in the ISPRS server and can be downloaded from the site: `https://www2.isprs.org/commissions/comm1/wg8/benchmark_smars/`. The files are stored according to the following structure:

```
City
├── GSD/
│   ├── change_map/
│   │   ├── Subset/
│   │   │   ├── City_GSD_Subset_change_map_2classes.tif
│   │   │   └── City_GSD_Subset_change_map_3classes.tif
│   │   ├── City_GSD_change_map_2classes_building_gt.tif
│   │   └── City_GSD_change_map_3classes_gt.tif
│   └── Event/
│       ├── original/
│       │   ├── City_GSD_Event.tif
│       │   ├── City_GSD_Event_building_gt.tif
│       │   ├── City_GSD_Event_dsm.tif
│       │   └── City_GSD_Event_gt.tif
│       └── splitting
│           ├── 5_class_gt/
│           │   └── City_GSD_Event_Subset_gt.tif
│           ├── building_mask_gt/
│           │   └── City_GSD_Event_Subset_building_gt.tif
│           ├── train/
│           │   ├── City_GSD_Event_train.tif
│           │   └── City_GSD_Event_train_dsm.tif
│           ├── val/
│           │   ├── City_GSD_Event_val.tif
│           │   └── City_GSD_Event_val_dsm.tif
│           └── test/
│               ├── City_GSD_Event_test.tif
│               └── City_GSD_Event_test_dsm.tif
├── coor
└── City-splitting-range.xlsx
```

Notes for naming:

- `City` can be the `SParis` or `SVenice` model
- `GSD` means ground sample distance, which is either 30cm or 50cm
- If the map is either in a ‘‘`Subset`’’ folder or has a ‘‘`Subset`’’ term in its name, this means it has been cropped for train, test or validation (val)
- `Event` can be the pre- and post- event case
- Maps with two categories are designed for building detection (building, no-building)
- Three classes maps are applied for change detection (No change, demolition, construction)
- ‘‘`5 class gt`’’ are the ground truth with the 5 available semantic classes
- Files without ‘‘`dsm`’’, ‘‘`gt`’’ or ‘‘`map`’’ in the naming are the optical images

A.2.1 File formats

- All files are provided as GeoTIFF rasters
- Class maps are files with discrete values
- DSM files are stored with float precision
- Change maps represent the transition between pre- and post- events and therefore are located in a different folder
- `coor` files include the coordinates corresponding to the corners of the available Subsets
- `City-splitting-range.xlsx` is the same as `coor` but in `.xlsx` format

A.3 Dublin dataset

The processed Dublin dataset is stored in Zenodo and can be downloaded from the site: `https://zenodo.org/records/12772927`.

The Dublin dataset processed for Stereo and MVS is organized as follows:

- **DSM**: DSM raster used as GT, it has a 10cm GSD. It is obtained from merging all point clouds into one raster

- **Images**: RGB images organized according to the acquisition tracks with camera parameters for each image as `txt`. The parameters are the extrinsic and intrinsic matrices and a depth range for MVS estimation. The images were downsampled from the original by a x9 ratio (getting a GSD of 30.6cm) and the camera parameters were adjusted accordingly.

- **Stereo**: Dataset for stereo pairs. Within each subfolder, left and right images are available. Images are paired as `X_Y.png` and `Y_X.png` The disparity maps are stored with a factor of 256 and can be converted with: `gdal_translate -scale 0 65535 0 256`. Disparity maps are obtained based on the camera position, left image perspective and the ground truth DSM.

- **MVS**: Dataset for MVS. Depth maps are included in a scaled version and can be converted with: `gdal_translate -scale 0 65535 220 476`. Within each folder, a `pair.txt` file is included, which can be used for MVS to select the closest views. These files are in the

format used by COLMAP. The depth maps are obtained based on the camera position, the image perspective and the ground truth DSM.