Dipl.-Ing. Christian Mostegel

Confidence Prediction for Scene Adaptive Multi-View Stereo

DISSERTATION

to be awarded the academic degree of

Doctor of Technical Sciences

submitted to

Graz University of Technology

Supervisors

Prof. Dr. Horst Bischof
Institute for Computer Graphics and Vision

Assoc. Prof. Dr. Stefano Mattoccia
University of Bologna

Graz, Austria, March 2018

Doubt grows with knowledge.

Johann Wolfgang von Goethe

Abstract

In this thesis, we propose a novel way to improve the quality and efficiency of the Multi-View Stereo (MVS) reconstruction process without changing the MVS algorithm itself. The key component of our method is the concept of MVS Confidence Prediction, which allows us to estimate the chances of a successful reconstruction before the MVS algorithm is executed. We use this prediction to actively select image constellations which are well-suited for the MVS algorithm with respect to the presented scene. This scheme allows us to maximize important quality parameters, such as coverage, ground resolution and 3D accuracy, while at the same time minimizing the number of images, the computation time and memory consumption.

The proposed machine learning technique does not require any ground truth or manually labeled data for training, but instead adapts ideas from depth map fusion for providing a supervision signal. The key idea is to use different view points for reasoning about contradictions and consistencies between multiple depth maps generated with the same MVS algorithm. This leads to a fully automated training scheme for MVS Confidence Prediction.

We then use the trained confidence predictor for automating and improving two steps in the reconstruction process. The first step is the image acquisition itself, where we propose an iterative image acquisition technique for autonomous drones. The second step prioritizes the MVS depth map generation process. This means that prior to executing the MVS algorithm, we select a good set of matching partners for each view and rank the resulting view clusters (i.e. key views with matching partners) according to their impact on the reconstruction quality. Both steps can be integrated within the traditional photogrammetric reconstruction process and can speed up the reconstruction process by up to a full order of magnitude without losing much information.

All elements of our approach are evaluated on a reoccurring photgrammetric task in a challenging environment; i.e. the preservation of prehistoric art on open rock surfaces surrounded by vegetation in the Valley of Valcamonica. Additionally, we evaluate our

training data generation approach on public two-view stereo datasets with ground truth. Finally, we evaluate the generalization performance of our MVS Prioritization on the task of reconstructing single family houses in an environment which was never seen in training. All experiments show that automated confidence learning can be a substantial benefit for MVS, leading to compact 3D reconstructions with highly complete content at a significantly lower computational cost.

Keywords. Multi-View Stereo, MVS Confidence Prediction, Machine Learning, Confidence Measures, MVS Prioritization, Image Clustering, View Cluster Ranking, View Planning, Active 3D Reconstruction

Kurzfassung

In dieser Arbeit präsentieren wir einen Ansatz um die Qualität und Effizienz des Multi-View Stereo (MVS) Rekonstruktionsprozesses zu steigern ohne den MVS-Algorithmus selbst zu verändern. Die Basis unserer Methode ist das Konzept der MVS-Konfidenzvorhersage, welches uns erlaubt die Chancen einer erfolgreichen Rekonstruktion abzuschätzen, noch bevor der MVS Algorithmus selbst ausgeführt wird. Diese Vorhersage erlaubt uns dann aktiv Bildkonstellationen auszuwählen, welche gut für den verwendeten MVS-Algorithmus, im Bezug auf die gegebene Szene, geeignet sind. Unter Verwendung dieses Ansatzes, maximiert unsere Methode dann wichtige Qualitätsparameter, wie Vollständigkeit, Bodenauflösung und 3D Genauigkeit, während sie gleichzeitig die Anzahl der Bilder, die Rechenzeit und den Speicheraufwand minimiert.

Um diese Vorhersage zu ermöglichen, präsentieren wir eine neue Art des maschinellen Lernens, welche keine Referenzrekonstruktion oder manuelle Interaktion zur Trainingsdatenbeschaffung benötigt, sondern stattdessen Ideen aus dem Bereich der Tiefenkartenfusion als Überwachungssignal verwendet. Die grundlegende Idee unseres Ansatzes ist es mit Hilfe verschiedener Blickwinkel Übereinstimmungen und Widersprüche zwischen mehreren Tiefenkarten, welche mit dem selben MVS-Algorithmus generiert wurden, zu finden und diese für die vollautomatische Trainingsdatengenerierung zu verwenden.

Nach dem Training, verwenden wir den resultierenden Konfidenzprädiktor dann um zwei Schritte im Rekonstruktionsprozess zu verbessern und zu automatisieren. Der erste Schritt ist die Bildaufnahme selbst, für die wir einen neuen iterativen Prozess zur Bildaufnahme mit autonomen Drohnen vorstellen. Der zweite Schritt ist die Priorisierung des MVS-Tiefenkartengenerierungsprozesses. Das heißt, dass wir vor der Ausführung des MVS-Algorithmus für jedes Bild gute Partnerbilder auswählen und die resultierenden Bildgruppen dann nach ihrer Bedeutung für die Rekonstruktionsqualität reihen. Beide Schritte können mit geringem Aufwand in den traditionellen photogrammetrischen Rekonstruktionsprozess integriert werden und haben das Potential diesen Prozess um bis zu einer

Größenordnung zu beschleunigen, ohne essentielle Information zu verlieren.

Alle Elemente unseres Ansatzes werden mit einer wiederkehrenden photogrammetrischen Aufgabe in einer herausfordernden Umgebung evaluiert. Diese Aufgabe ist die Rekonstruktion von prähistorischer Kunst, welche auf offenen Gesteinsformationen im Valcamonica Tal verewigt wurde und von dichter Vegetation umgeben ist. Zusätzlich werten wir unseren Ansatz für Trainingsdatengenerierung auf öffentlichen Two-View Stereo Datensätzen mit Referenztiefenkarten aus. Schlussendlich evaluieren wir das Generalisierungspotential unserer MVS-Priorisierung bei der Rekonstruktion von Einfamilienhäusern in einer Umgebung, welche nie im Training gesehen wurde. Alle Experimente demonstrieren, dass vollautomatisches Konfidenz-Lernen einen substanziellen Vorteil für MVS bringen kann und für die Generierung von kompakten und hoch-vollständigen 3D Rekonstruktionen bei signifikant reduziertem Rechenaufwand verwendet werden kann.

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources. The text document uploaded to TUGRAZonline is identical to the present dissertation.

(date)	(signature)

Acknowledgments

The successful completion of this thesis would not have been possible without the constructive contribution and support of many people.

First of all, I would like to thank my supervisor, Prof. Horst Bischof, for granting me the opportunity to pursue a PhD in such an interesting and competitive field of research and allowing me to freely develop my skills and knowledge.

Second, I would like to thank my second supervisor, Prof. Stefano Mattoccia, for taking the time to read my thesis and come the long way to Graz for attending my defense.

Third, I would like to thank all my colleagues from the institute, the reading group and all members of 3D Pitoti project for the valuable discussions, their help and the bibulous comradeship. A very special thanks in this regard goes to all my fellow group members, who were not only always extremely supportive, but also put up with my peculiarities, including flying drones through the office (maybe sometimes a little bit too close over their heads), swinging staffs next to them or simply distracting them from work with trivialities to satisfy my daily need for personal interaction.

Finally, I am greatly indebted to my family and friends for their support, their time and their love. I owe my deepest gratitude to my loving wife, Theodora, who selflessly assisted me on every step of the way and in every possible way. She did not only grant me the opportunity to concurrently pursue a PhD and found a steadily growing family, but she was and is my safe Island encircled by a powerful reef that can calm any wave threatening to sink our ship. Thank you!

This thesis received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 730294, the EC FP7 project 3D-PITOTI (ICT-2011-600545) and from the Austrian Research Promotion Agency

(FFG) together with OMICRON electronics GmbH as Bridge1 project 843450.

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Contribution . . . 4

2 Related Work **9**
- 2.1 3D Reconstruction from 2D Images - Basics and Notation . . . 10
- 2.2 Multi-View Stereo . . . 19
- 2.3 Multi-View Stereo Confidence Prediction . . . 24
- 2.4 View Planning . . . 28
- 2.5 Matching Partner Selection . . . 31
- 2.6 Summary . . . 33

3 Confidence Learning without Ground Truth **35**
- 3.1 Overview . . . 36
- 3.2 Setup . . . 38
- 3.3 Motion-Supervised Training Data Generation . . . 39
  - 3.3.1 Stage 1: Support Assessment . . . 39
  - 3.3.2 Stage 2: Consistency Voting . . . 42
  - 3.3.3 Stage 3: Detection of Outliers and Missing Measurements . . . 44
- 3.4 Learning and Predicting MVS Confidence . . . 46

4 Automated Image Acquisition for Multi-View Stereo **49**
- 4.1 Image Acquisition System . . . 50
- 4.2 View Planning . . . 52

5 Multi-View Stereo Prioritization **61**
- 5.1 Preprocessing . . . 62
- 5.2 Matching Partner Selection . . . 63
- 5.3 Probabilistic MVS Confidence Extension . . . 66
- 5.4 Next-Best View Ranking . . . 69

6 Experiments **73**
6.1 Training Data Generation for Two-View Stereo Confidence Measures 74
6.1.1 General Setup . . . 74
6.1.2 KITTI Dataset . . . 75
6.1.3 Middlebury Dataset . . . 79
6.1.4 Strecha Dataset . . . 81
6.1.5 Discussion . . . 83
6.2 MVS Confidence Prediction . . . 85
6.2.1 KITTI2012 Dataset . . . 85
6.2.2 Valcamonica Dataset . . . 87
6.2.3 Discussion . . . 93
6.3 Automated Image Acquisition for Multi-View Stereo . . . 95
6.3.1 Experimental Procedure . . . 95
6.3.2 Results . . . 97
6.3.3 Discussion . . . 99
6.4 Multi-View Stereo Prioritization . . . 103
6.4.1 Evaluation Details . . . 103
6.4.2 Selecting the k-best Matching Partners . . . 106
6.4.3 Ranking Only . . . 110
6.4.4 Full MVS Prioritization . . . 114
6.4.5 Discussion . . . 120

7 Conclusion and Outlook **123**

A Acronyms **127**

B List of Publications **129**
B.1 2014 . . . 129
B.1.1 Active Monocular Localization: Towards Autonomous Monocular Exploration for Multirotor MAVs . . . 129
B.2 2015 . . . 130
B.2.1 Graz Griffins Solution to the European Robotics Challenges 2014 . . . 130
B.2.2 Performance Evaluation of Vision-Based Algorithms for MAVs 130
B.3 2016 . . . 131
B.3.1 Using Self-Contradiction to Learn Confidence Measures in Stereo Vision . . . 131
B.3.2 UAV-based Autonomous Image Acquisition with Multi-View Stereo Quality Assurance by Confidence Prediction . . . 132

B.3.3 The 3D-PITOTI Project with a Focus on Multi-Scale 3D Reconstruction using Semi-Autonomous UAVs 133
B.4 2017 134
B.4.1 Evaluations on multi-scale camera networks for precise and geo-accurate reconstructions from aerial and terrestrial images with user guidance 134
B.4.2 Der Einsatz unbemannter Flugsysteme zur Charakterisierung von gesprengtem Haufwerk 135
B.4.3 Scalable Surface Reconstruction from Point Clouds with Extreme Scale and Density Diversity 135
B.5 2018 136
B.5.1 Prioritized Multi-View Stereo Depth Map Generation using Confidence Prediction. 136

Bibliography **138**

Chapter 1

Introduction

1.1 Motivation

The process of extracting 3D information from 2D images (aka photogrammetry) has a long standing history, which goes back to the 19^{th} century [74, 83]. The key idea is to find correspondences between images and then use the knowledge of the camera orientation and position at the time of the image acquisition to triangulate the position of the correspondences in 3D. Nowadays, state-of-the-art photogrammetric pipelines (e.g. [30, 70, 89, 110, 115]) can generate very accurate and complete 3D reconstructions from a given set of images – provided that this set of images fulfills a range of prerequisites.

Some of these prerequisites are innate to the task. I.e. in order to measure 3D information at a pixel location, the object at this location must be observed from at least two different view points. Further, the object surface (which we aim to measure) must be visible to the sensor; i.e. the illumination has to be suitable for the camera settings (to avoid under or overexposure) and the object should not be completely transparent or reflective. The last prerequisite is that the object surface at a pixel location should be sufficiently visually distinct from other surface parts, so that it is possible to find the corresponding pixel in another image.

Note that for a natural image taken with a physical camera, it is extremely unlikely that all these basic assumptions (especially the visual distinctiveness) are fulfilled for all pixels. Therefore, all common photogrammetric pipelines first focus only on very salient image regions. This first step is called Structure-from-Motion (SfM) and its main purpose is to determine the relative camera poses (i.e. the

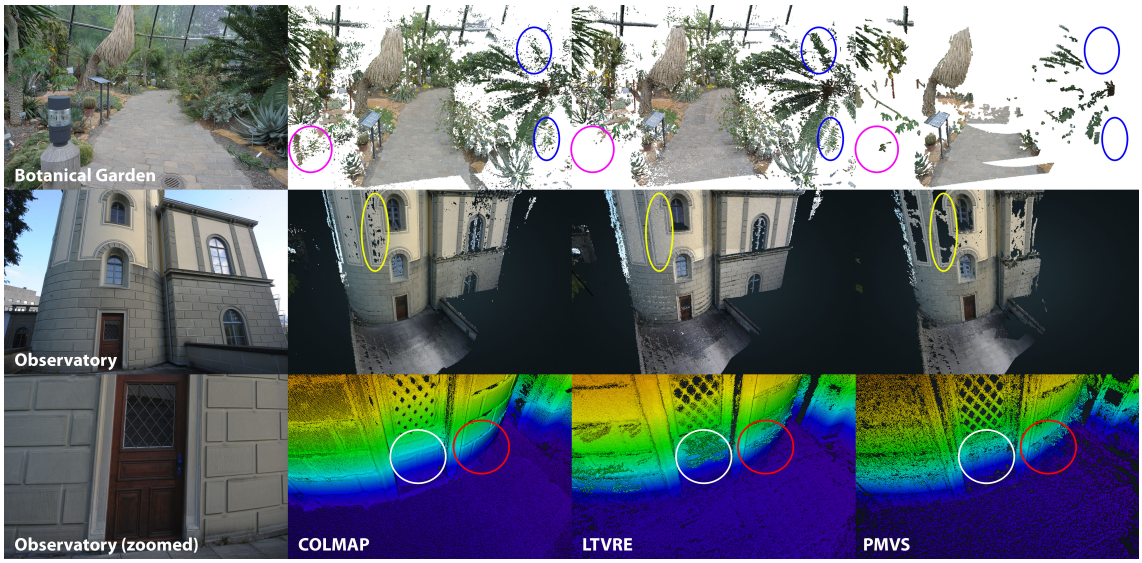
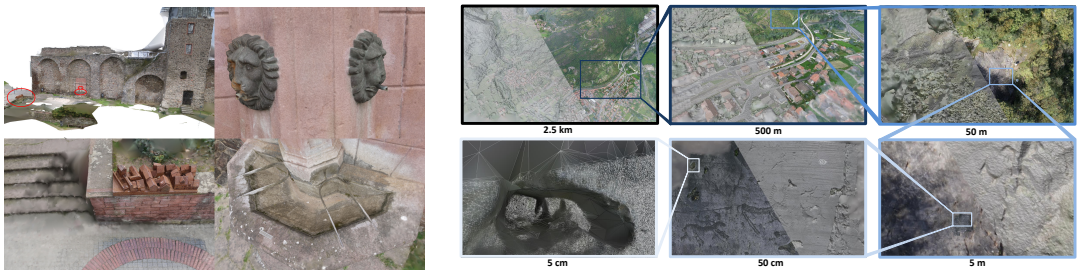


Figure 1.1: Multi-View Stereo (MVS) challenges. We show results for 3 different MVS approaches (COLMAP [115], LTVRE [70] and PMVS [32]) on two scenes (Botanical Garden and Observatory) of the very recent ETH3D dataset [116]. Note that the Observatory scene is shown two times, once with a point cloud colored with the input images and once a zoomed detail of the same point cloud colored with the object height. While all approaches operate on the same set of images, the results strongly vary as the underlying assumptions of the MVS approaches differ quite significantly. E.g. in contrast to the other two approaches, PMVS has a very strong assumption of local planarity and visual saliency. Consequently, it has extreme problems with high-frequency structures such as vegetation (top row) and regions with low texture (yellow ellipse middle row), where it simply fails to produce any output. If we compare COLMAP and LTVRE, we see a trend that LTVRE is slightly more complete (blue and yellow ellipses), however it also contains significantly more outliers (ellipses in bottom row). Aside from this trend, it is often very hard to tell why one approach delivers results in one part of the scene and not in others (compare cyan and blue ellipses in the top row).

relative position and orientation of the camera at the time when the images were taken). Commonly, this is achieved by first detecting key points (typically corners and/or blobs), then computing a descriptor for each key point and finally matching the descriptors between images. The found correspondences are then used to simultaneously optimize the 3D position of the key points and the relative camera poses.

After this step, a process called Multi-View Stereo (MVS) uses the (now fixed) camera poses to recover as much 3D information as possible. Ideally, an MVS algorithm would measure a 3D value for each pixel in each image. However, as



(a) Citywall Dataset.

(b) Valley Dataset.

Figure 1.2: Examples of surface reconstruction as a subsequent step to MVS. Both examples were processed with a scalable state-of-the-art surface reconstruction approach [89]. (a) shows the resulting surface mesh for the Citywall dataset [30], which consists of 564 images of a historical city wall. This results in a point cloud of 300M points with MVE [30] as MVS algorithm. The top left image shows an overview, where the red circles indicate the location of the zoomed out details of the other images. In the bottom left image, we can observe reconstruction problems at the stair case, which are the result of large regions without any measurements from the MVS algorithm. However, if the MVS algorithm delivers sufficient supporting points, then the surface reconstruction is able to extract a very detailed 3D model (images on the right). Note that the image acquisition for this kind of data only takes around one hour, while the surface reconstruction alone takes a full day. (b) shows the Valley dataset [89], which consists of a MVS point cloud containing 2 billion points obtained with SURE [108] as MVS algorithm. Turning this point cloud into a consistently connected mesh takes 9 days using 120GB of RAM on a server with 40 CPU cores. Note that the computational complexity of all surface extraction approaches is at least linear in the number of input points (i.e. the points generated by the MVS algorithms). For scalable approaches like [89], this means that a reduction in the point cloud size directly relates to approximately the same reduction in run-time.

mentioned before, it is highly unlikely that all basic requirements are fulfilled for each pixel. Consequently, MVS algorithms have to make additional assumptions to fulfill this challenging task. These assumptions can include a static environment, local planarity, Labertian ("diffuse") reflectance properties and/or sufficient visual saliency. The set of assumptions made the MVS algorithm has direct influence on its performance. If all assumptions are fulfilled, current MVS approaches are able to produce highly accurate and complete 3D reconstructions, however, if some of the assumptions are violated, then the resulting reconstructions very often contain outliers and/or undesirable holes. Sometimes the assumptions of an MVS algorithm can lead to a complete reconstruction failure for certain types of objects (see Figure 1.1).

Another problem of MVS is its extreme demand for resources (i.e. computation time and memory). If we look at the current state-of-the-art in MVS, we see that most approaches (e.g. [35, 45, 108, 115, 142]) aim to produce one depth map for each image. While this approach has the advantage of being completely scalable, this also means that an immense amount of data is generated. More specifically, this means that 3D points in the order of 10^7 are created per image of a modern camera. With a few hundred images, this leads to billions of points that have to be stored, visualized and/or handled by subsequent processing steps such as depth map fusion or surface reconstruction. In Figure 1.2, we visualize the impact of the immense amount of generated data for the run-time of the subsequent step of surface reconstruction. For this subsequent step, the run-time complexity is linear in the number of input points. This means that if the number of required images is lower, then we do not only save a lot of computation time and memory in the MVS processing step itself, but also directly save computation time in the surface reconstruction step. However, this goal of improved efficiency stands in direct opposition to common photogrammetric practice, which tries to capture as many images from as many angles as possible (see e.g. [137]). The main reason for this intentional redundancy is that more images can only lead to higher completeness, however, very often this leads to unnecessarily redundant image sets, which only makes the 3D reconstruction process more costly in each and every step. This motivated us to investigate one big unsolved question in photogrammetry: What is (un)necessary redundancy for dense 3D reconstruction?

1.2 Contribution

This thesis contains three main contributions, where the last two can be seen as applications of the first.

Our first contribution deals with the problem of predicting if a MVS reconstruction will succeed or not. As mentioned before, the reconstruction success depends on many factors, including the image constellation, the assumptions of the MVS algorithm, the light conditions, the image exposure and the scene structures themselves. Thus, our first contribution is a learning-based confidence prediction framework (published in [91]), which aims to estimate the chances of a successful MVS reconstruction *prior* to executing the MVS algorithm itself.

The most notable property of our learning approach (published in [92]) is that it

does not require any labeled training data or ground truth. Instead we use ideas of depth map fusion to detect and learn unmodeled errors in the 3D output. The main motivation behind this approach is that the systematic assumption violations are not the same from each view point. For example, if there is no visual information in one part of the image due to overexposure, images from significantly different view points often do not contain the same problem. Motivated by this observation, we evaluate different reconstruction subsets against each other and thus generate training data for a machine learning algorithm. All that our approach requires is a large set of images, which observe the same scene from many different view points.

The output of our machine learning approach is a pixel-wise confidence predictor, where the confidence represents the likelihood of a successful 3D reconstruction. With "successful" we mean that the produced 3D/depth value at a pixel location abides with our uncertainty model. We show that the nature of the confidence function, which we parameterize with the triangulation angle[1], depends on the used algorithm as well as the nature of the 3D structure itself. High-frequency structures, such as vegetation or fences, result in a very different confidence function than smooth structures, which are much easier to reconstruct.

The other two contributions of this thesis are two inclusive approaches that aim to increase the efficiency of the MVS reconstruction process, while at the same time fulfilling a set of desired quality requirements such as coverage, ground resolution and 3D accuracy.

The first approach is the automated image acquisition for MVS reconstruction with high-resolution images at close range (published in [91]). The aim is to acquire images in such a way that they optimize three competing goals at the same time. The first goal is that the images are suited for processing with a specific MVS algorithm. The second goal is that the images maximize the quality requirements (i.e. coverage, ground resolution and 3D accuracy). And the third goal is that the acquisition time is as low as possible. For achieving these goals, we loop between planning and autonomous execution to iteratively explore the scene and improve the reconstruction quality. Within this procedure, the confidence prediction allows us to integrate the requirements of an MVS approach with respect to the scene structure, which leads to an improved reconstruction success without the need of running the

[1]The triangulation angle is the angle spanned by two cameras and a 3D measurement (see Figure 2.4 for a visual example).

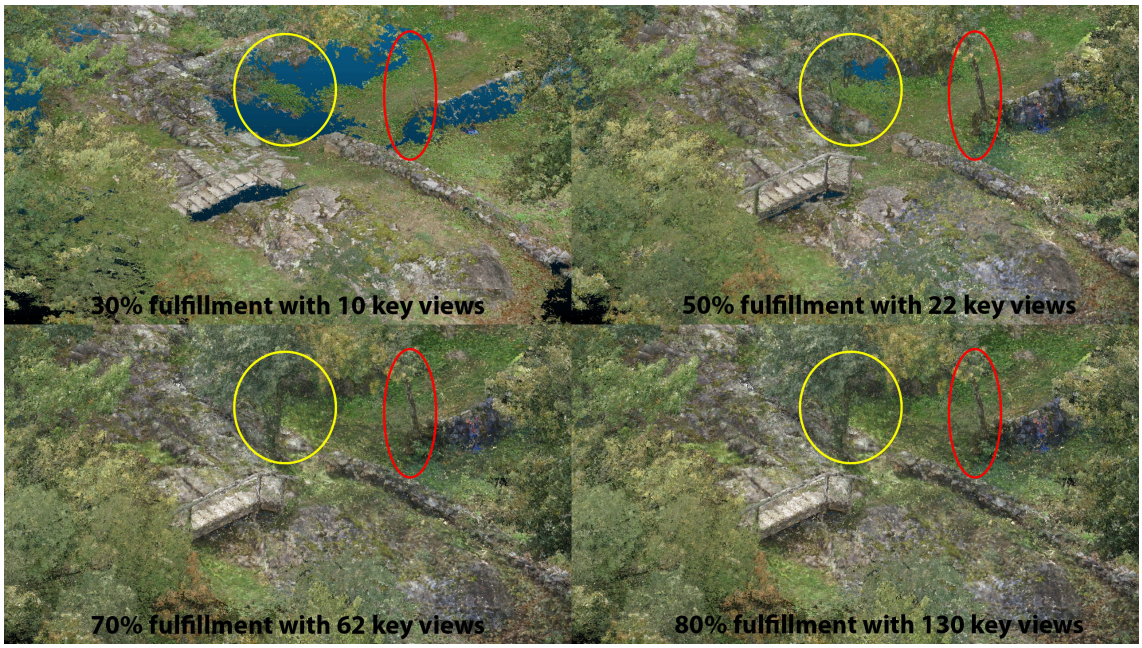


Figure 1.3: MVS Prioritization as one application of MVS Confidence Prediction. Our approach allows us to prioritize/rank view clusters (i.e. key views with matching partners) such that a highly complete and accurate point cloud can be obtained with a very small fraction of the available images as key views. Here, we show the point clouds from the raw depth maps of the view clusters (with 11 matching partners) ranked with our approach after reaching 30%,50%,70% and 80% of the maximal achievable quality fulfillment (i.e. completeness with respect to a desired ground sampling distance and accuracy of 1cm). The color of the points is extracted from the images and the blue gradient background illustrates holes in the reconstruction. Already with 50% fulfillment and only 22 view clusters / key views (i.e. 1.8% of the potential key views), most parts of this complex scene are already contained in the reconstruction (red ellipse) and only a small part is missing (yellow ellipse). With 70% fulfillment, even strongly occluded parts such as tree trunks (see ellipses) are contained in the point cloud, although this point cloud is computed from only 62 key views (i.e. 5% of the potential key views). For going from 70% to 80% fulfillment, the number of necessary key views already has to be more than doubled, however, the visual difference between those two point clouds is nearly imperceptible (see Video [87]).

costly MVS approach during acquisition. This approach can be used stand-alone to reduce the number of required images or in conjunction with traditional acquisition methods such as grid planning to increase the scene coverage at low cost.

The second approach (published in [88]) assumes that the image set is fixed (i.e. the acquisition is completed and no further images can be acquired) and aims to reduce the run-time of MVS as well as the size of the resulting point cloud by

exploiting the fact that, in most cases, the set of images is highly redundant.

Our approach, which we call MVS Prioritization, works in two steps. First, we find a good set of matching partners (i.e. images used for matching/depth inference) for each image in the image set, such that a specific MVS algorithm has a high chance of generating a complete and accurate depth map for these images. In this first step, all images are considered as the central images for depth map generation (further called *key views*), which results in a set of view clusters, where each image acts as the key view for exactly one view cluster, while it can also be part of other view clusters as matching partner.

In the second step, the resulting view clusters are then prioritized/ranked depending on their impact on the quality requirements. As we formulate the quality requirements as a monotone submodular function, we are able to obtain strong optimality guarantees for a solution found with a greedy algorithm [94]. The confidence prediction supports both steps and allows us to rank all view clusters with an estimate of the expected quality fulfillment (i.e. completeness with respect to a desired resolution and accuracy) for each entry – without having to execute the actual MVS algorithm within the ranking procedure.

This formulation has many advantages. First of all, the computed quality fulfillment function provides the opportunity to decide how many view clusters are necessary to obtain a certain quality fulfillment level with respect to a maximal achievable quality (i.e. the quality obtained with all available view clusters). Thus, the operator can either choose to reconstruct the n best view clusters and has an estimation of the expected level of quality fulfillment or can simply query how large n should be to reach a certain level. The second advantage is that the inherent parallelism of MVS based on depth maps is maintained as our ranking procedure happens before executing the MVS reconstruction step. Third, the overall efficiency of the MVS reconstruction step can be significantly improved without changing the MVS algorithm itself. In our experiments, we were thus able to obtain a quality fulfillment of 70% with only 5% of the available key views. This leads to a speed up factor of approximately 10 and a complexity/memory reduction factor of approximately 20 for the resulting point cloud without losing much information (see Figure 1.3).

Chapter 2

Related Work

In this thesis, we aim to improve dense 3D reconstruction from 2D images. Thus, we start our related work chapter with reviewing the basics of this field in Section 2.1. If you are already familiar with the pinhole camera model, epipolar geometry and structure-from-motion, then you can directly start with the next section (i.e. Section 2.2), which reviews the current state-of-the-art in Multi-View Stereo (MVS) and works out the most common assumptions made by current MVS approaches and their implications.

These implications then directly lead us to Section 2.3, which reviews works related to our first contribution, i.e. MVS Confidence Prediction. The goal of MVS Confidence Prediction is to predict the chances of a successful MVS reconstruction before the actual MVS algorithm is executed. To make this prediction possible, we have to answer two questions. First, how can we model the prediction (which is related to the field of confidence measures), and second, what supervision signal can we use to learn this prediction in an automated and scalable fashion (which is related to the field of alternative machine learning supervision).

In Section 2.4, we then come to a field of research called view planning. This field is related to both our other contributions. For our second contribution (i.e. automated image acquisition), view planning plays an essential role as we have to actively decide, where, in which constellation and order further images should be acquired. For our third contribution (i.e. MVS Prioritization), where we aim to improve the efficiency of MVS by ranking view clusters (i.e. images with matching partners), the underlying problem still remains closely related to view planning, with the difference that the image set is fixed and no further images can be acquired.

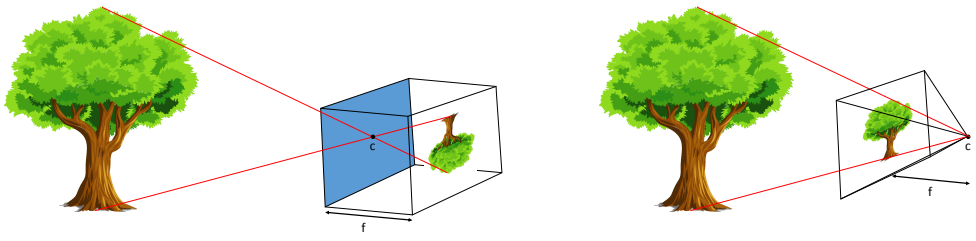
The second field, which plays an important role for MVS Prioritization, is the field of matching partner selection, which we review in Section 2.5. Finally, we close this chapter by summarizing the highlights of the reviewed related work in Section 2.6.

2.1 3D Reconstruction from 2D Images - Basics and Notation

3D reconstruction from 2D images starts, as the name suggests, with 2D images captured with a "normal" digital camera. The basic projective nature of such a camera is typically modeled with an idealized pinhole camera (see Figure 2.1). In a pinhole camera, all rays of light are completely straight and go from all directions within the field of view of the camera through the "pinhole" (i.e. a point opening of infinitesimal size) and project onto an ideal sensor plane, which can record the intensity and wave length of all light that projects onto the sensor within an infinitesimal small time frame.

Pinhole Model versus Real Digital Camera. At this point, we would like to make the reader aware that the pinhole camera model can be a useful instrument of abstraction, but that a real digital camera is a much more complex system with many physical constraints and imperfections. First of all, a real digital camera has a physical lens system, which does not let all incoming light through on a straight path, but bends the light in a non-linear way, which slightly differs for various wave lengths (this effect is called chromatic aberration). Even worse, the lens system partially absorbs the incoming light and this absorption typically increases with the distance from the optical axis[1] (this effect is known as vignetting). Further, the light rays pass through an aperture (i.e. the physical counterpart to an ideal pinhole) with a non-negligible size. This means that, while an ideal pinhole camera sees everything sharp, a real camera only sees sharp in a certain range (aka depth of field) around the plane of focus. After the light passed the lens system, it falls onto a digital sensor, which discretizes the image into pixels. This means that all light rays that fall onto the area of the pixel (which has a physical size) are grouped

[1]The optical axis is the one virtual light path through the center of the lens system which stays completely straight



(a) Idea of Pinhole Camera.

(b) Visualization of Pinhole Camera.

Figure 2.1: Pinhole Camera Model. In (a), we visualize the basic idea of the pinhole camera, which is a simple box, where light can only enter through an infinitesimal small hole (i.e. the pinhole/optical center/camera center, denoted as c). At the back of the box, there is the image plane (physically represented by a film or sensor), which records the light passing through the pinhole. Due to the pinhole, all light can be represented by light rays (red lines), which all follow a straight path from the scene through the pinhole to the image plane. The focal length f (i.e. the distance of the image plane to the camera center) defines how large the projection is on the sensor/image plane. In (b), we show a second type of visualizing a pinhole camera, which places the image plane in front of the camera center. While not physically sound (as it represents an outward projection and not an inward projection), this type of representation has the advantage that the image is not rotated, which makes it easier to interpret what is happening. Consequently, nearly all works (this work included) use this representation for visualizing the pose of a pinhole camera.

into a single value, which is then quantized into a certain number of bits (typically 8 to 16). Further, the light in one pixel is integrated over a non-negligible period of time (aka exposure time or shutter speed), which might introduce blur effects (aka motion blur) if the camera moves during this period of time. Aside from this, there are many, many more physical limitations and imperfections related to a real sensor (e.g. sensitivity, linearity or noise), of which the most important are summarized and modeled in the EMVA Standard 1288 [24].

Using the Pinhole Model. Now that we have established that a real digital camera is far from an ideal pinhole camera, it is worth mentioning that nearly all photogrammetric pipelines have something very close to a pinhole camera in their core due to its simplicity and linearity. The main reason why this works is that many of the real world effects of the camera system can be removed or at least mitigated

through calibration.

Intrinsic Camera Calibration. For 3D reconstruction, the most important type of the calibration is the intrinsic (geometric) calibration of the camera. This type of calibration aims to determine the direction of each pixel ray. Note that a pixel ray is an idealized ray going through the virtual optical center of the camera and the center of the pixel. In the real world, a pixel ray would correspond to something close to a cone, where the base shape of this cone would be close to a quadrilateral, but this base shape would vary slightly for each pixel depending on the lens distortion. However, for the purpose of calibration the notion of a pixel ray is sufficient as most calibration methods will try to fit some kind of global distortion model for the whole image. There exist many of such distortion models, of which many can achieve sub-pixel accuracy with a suitable calibration method (see [128] for a recent analysis on distortion models).

After the geometric calibration, the orientation of each pixel ray is known (with some uncertainty) with respect to a camera coordinate system, where the origin is in the virtual optical center (also called camera center). In theory, the orientation of the axes of this coordinate system can be placed arbitrarily, however, the x- and y-axis are typically aligned close to parallel with the sensor axes and the z-axis with the optical axis. The camera distortion model, now allows us to warp the original image into the camera coordinate system of a virtual pinhole camera (this process is called "undistortion").

After the undistortion, the undistorted image can be treated as if it were captured with the virtual pinhole camera. The intrinsic calibration of this virtual pinhole camera can be described by a matrix $\mathbf{K}$:

$$\mathbf{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.1}$$

where f is the focal length (which represents the distance of the image plane to the camera center in pixel), and $P_c = (c_x, c_y)$ is the principle point (i.e. the point where the z-axis (or optical axis) intersects the image plane in pixel coordinates). Note that there are more complex representations of $\mathbf{K}$ (see e.g. [50] Section 6.1), but for

most applications this representation of the virtual pinhole camera is sufficient, also because the undistortion process can be used to warp the image such that it fits this simple model.

Extrinsic Calibration. If we have a camera setup with multiple rigidly connected cameras, then it is possible to calibrate their relative pose (i.e. relative position and orientation) to each other. An important point in this extrinsic calibration process is that some scene structure (e.g. the size of a checkerboard) has to be known metrically, as otherwise the relative pose between the cameras can only be determined up to an unknown scale factor. A camera pose has 6 degrees of freedom. The rotation is most of the time represented as a 3×3 rotation matrix $\mathbf{R}$ with 3 degrees of freedom and the position of the optical center as 3×1 vector $\mathbf{c}$ with 3 degrees of freedom. Rotation and position make up the "extrinsic" calibration of a camera to a common coordinate frame (which can be chosen arbitrarily). Together with the intrinsic calibration, this lets us define the 3×4 projection matrix $\mathbf{P}$ of a camera:

$$\mathbf{P} = \mathbf{K} \cdot [\mathbf{R}|\mathbf{t}], \tag{2.2}$$

where $\mathbf{t}$ is often referred to as the translation vector and is defined as $\mathbf{t} = -\mathbf{R} \cdot \mathbf{c}$. This projection matrix now allows us to project a 3D point $\mathbf{x_{3D}}$ into the pinhole camera. The corresponding projected point $\mathbf{x_{2D}}$ can then be found by using homogeneous coordinates and matrix multiplication as:

$$\mathbf{X_{2D}} = \mathbf{P} \cdot \mathbf{X_{3D}}, \tag{2.3}$$

where $\mathbf{X_{3D}} = [\mathbf{x_{3D}}^T|1]^T$ is the homogeneous 3D point and $\mathbf{X_{2D}}$ is the homogeneous projected point in image coordinates of the undistorted image with:

$$\mathbf{x_{2D}} = \mathbf{X_{2D}}(1:2)/\mathbf{X_{2D}}(3). \tag{2.4}$$

For monocular 3D reconstruction, where only one camera is used to take images from multiple view points, the extrinsic calibration cannot be precomputed in advance and instead is part of an optimization process called Structure-from-Motion, however, the basic camera model and notation stays the same.

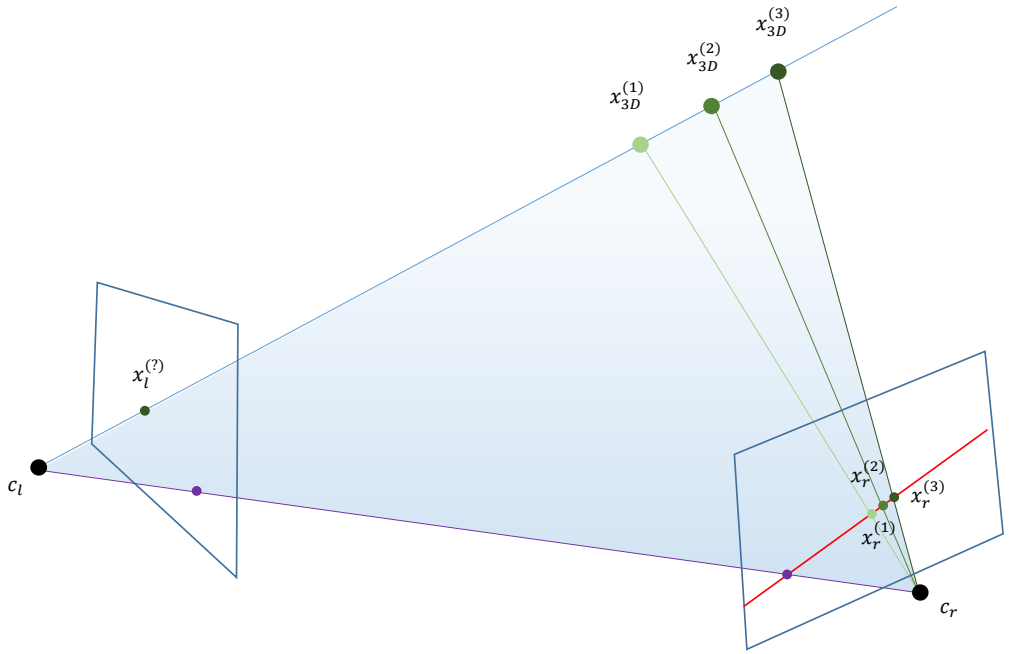


Figure 2.2: Epipolar Geometry. We show two pinhole cameras with associated camera centers c_l and c_r. Here, we aim to find the correspondence for the key point x_l in the right image. For this point, the epipolar geometry (i.e. the relative pose of the two pinhole cameras) defines an epipolar line in the right image (red line), on which the corresponding point x_r has to lie. This means that, purely geometrically, all points along the epipolar line could be correct correspondences (see $x_r^{(i)}$ with $i = 1, 2, 3$)). This ambiguity has to be resolved through visual correspondences (in the case of keypoints, this is achieved by descriptor matching) to find the unknown 3D position x_{3D} corresponding to the object surface point projecting to the image location x_l.

Structure-from-Motion (SfM). The main purpose of SfM is to determine the relative camera poses. With the term "camera pose", we refer to the pose of the virtual pinhole camera at the time that an image was taken in a common coordinate system. I.e. each (undistorted) image is associated with a virtual pinhole camera (further simply referred to as "camera") with its intrinsic calibration and pose.

To estimate the camera poses, all common SfM pipelines start with detecting salient points (aka key points). A key point is typically a structure, which can be well-localized in the image (e.g. a corner or a blob). Then the region around a key point is described in a discriminative and dimensionality reducing way, such that key points originating from the same 3D structure (but from different images) should have very similar descriptors, while key points from different structures should have

more dissimilar descriptors. As a key point with descriptor is also referred to as "feature", the corresponding process is often referred to as "feature extraction". The most popular feature extraction approaches are SIFT [76] and SURF [8], both with their own detector and descriptor.

After the feature extraction, the features are matched between images by comparing the descriptors, which results in a set of potential correspondences. As this set will contain many wrong matches (i.e. outliers), they are robustly verified with the correspondence condition of epipolar geometry [50] within a RANSAC [28] procedure. The correspondence condition is essential for 3D reconstruction algorithms and is defined as follows:

$$\mathbf{X_1}^T \cdot \mathbf{F} \cdot \mathbf{X_2} = 0, \tag{2.5}$$

where $\mathbf{F}$ is the fundamental matrix and $\mathbf{X_1}$ and $\mathbf{X_2}$ are arbitrary corresponding homogeneous 2D points in two images. The fundamental matrix is a 3×3 matrix with 7 degrees of freedom, which is defined through the relative pose of two calibrated cameras (see Section 9.2 in [50] for more details). The meaning of this correspondence condition is that if $\mathbf{X_1}$ is a valid correspondence of $\mathbf{X_2}$, then $\mathbf{X_1}$ has to lie on the epipolar line l_1 defined through $\mathbf{X_2}$ and the fundamental matrix as $l_1 = \mathbf{F} \cdot \mathbf{X_2}$ (and vice versa). Note that this also means that once the poses of the calibrated cameras are known, the search for a corresponding point reduces to a search along the epipolar line, which we visualize in Figure 2.2.

After the fundamental matrix is found for each pair of cameras, the 3D reconstruction and pose estimation process can start. This can be achieved incrementally, camera by camera, (see e.g. [30, 110, 114]) or globally for all cameras jointly (see e.g. [18, 127]). The main idea of the more common incremental SfM is to fix one camera and estimate the relative pose to a second camera with a suitable algorithm (e.g. 5-point algorithm [95]). Then the correspondences are triangulated in 3D, leading to a sparse point cloud. After this initialization, new cameras are incrementally added to the reconstruction by verifying the previously found correspondences with the existing sparse reconstruction (e.g. with the perspective-3-point algorithm [38] in a RANSAC [28] loop). When a new camera is added, all features with correspondences to already integrated cameras are either added to the feature track of existing 3D points or are used for triangulating new 3D points. This whole process is supported by a non-linear optimization called bundle-adjustment [50]. This optimization tries

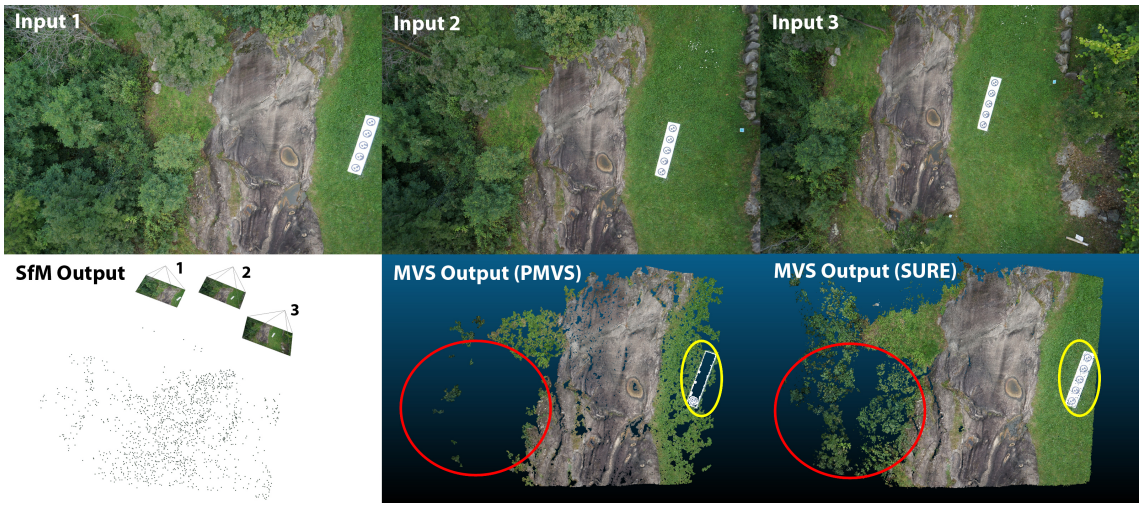


Figure 2.3: 3D reconstruction example. The top row shows the three input images, which are used for a fully automated 3D reconstruction. In a first step, a process called Structure-from-Motion (SfM) estimates the camera poses and a sparse reconstruction of salient scene points (bottom left). Then in a second step, a process called Multi-View Stereo (MVS) uses the SfM camera poses to extract 3D information for as many pixels as possible. Whether or not an MVS algorithm succeeds in this task depends on many parameters, including the scene structure, the image constellation and the assumptions made by the MVS algorithm. In the bottom, we visualize the output of two different MVS algorithms. Note that for this example, SURE [108] produces a much more complete reconstruction than PMVS [32], although both MVS algorithms use the exact same images. While both approaches deliver good results for planar surfaces (rock surface and short grass), PMVS has problems in regions with low texture (yellow ellipse) and high frequency structures such as trees (red ellipse), which leads to many missing parts in the reconstruction.

to minimize the reprojection error (i.e. the distance between feature measurements and the projection of the corresponding 3D point) by jointly optimizing the camera poses and the 3D points. Bundle-adjustment is a very important part of all SfM pipelines and can also be used to refine and correct the camera distortion (see e.g. [110]), however, like all non-linear optimization techniques, it strongly depends on a good initialization.

The main output of SfM for our purposes are the camera poses. In this sense, SfM can be seen as method for the extrinsic calibration of all (virtual) cameras. Like for other extrinsic calibrations, the relative camera poses can only be determined up to an unknown scale factor. If a metric or geo-referenced reconstruction is desired, then additional external metric information has to be provided (e.g. through ground control points [110]). Additional to the camera poses, SfM also delivers a sparse

point cloud, however, for most purposes this point cloud is not dense enough and even for a human observer it is often very hard to interpret these point clouds (see Figure 2.3). This problem is solved by a subsequent processing step, called Multi-View Stereo (MVS). However, before we review the current state-of-the-art in MVS, we have to discuss one last highly relevant topic, i.e. the modeling of 3D uncertainty.

Modeling 3D Uncertainty. As photogrammetry aims to measure 3D structures, it is important to have an estimate of expected error of a 3D measurement (aka 3D uncertainty or 3D accuracy). Such an estimation is typically obtained by modeling the uncertainty distribution.

In this work, we use the uncertainty model proposed in [49]. This model has three main assumptions. The first assumption is that the noise of the camera parameters is small compared to the noise of the 3D points. The second assumption is that we have a good estimate for the expected 2D uncertainty. The third assumption is that the uncertainty distribution of a 3D point can be approximated with a Gaussian 3D distribution.

Under these assumptions, we can use first order covariance propagation to propagate the assumed 2D uncertainty to 3D and thus estimate the 3D uncertainty distribution (represented by a covariance matrix) for each 3D point [50]. As point of linearization, we use the estimated 3D point $\mathbf{x_{3D}}$ and obtain the corresponding covariance matrix as:

$$Cov_{3D} = \left(\sum_{C_i \in C^k} J_i^T \cdot Cov_{2D}^{-1} \cdot J_i \right)^{-1}, \tag{2.6}$$

where J_i is the 2×3 Jacobian of the projection function (Equation 2.4) of camera C_i in the 3D point $\mathbf{x_{3D}}$, C^k is the set of k cameras that observe $\mathbf{x_{3D}}$ and Cov_{2D} is the assumed 2D covariance (i.e. the image noise).

Note that there are many other approaches available for modeling the 3D uncertainty, where some also represent the pose estimation uncertainty (e.g. [84]) and others use a different parameterization (e.g. [16, 22, 36]), however, for our task the model presented above works sufficiently well.

One of the properties that all these models (including the model used in this work) share is that the 3D uncertainty strongly depends on the triangulation angle.

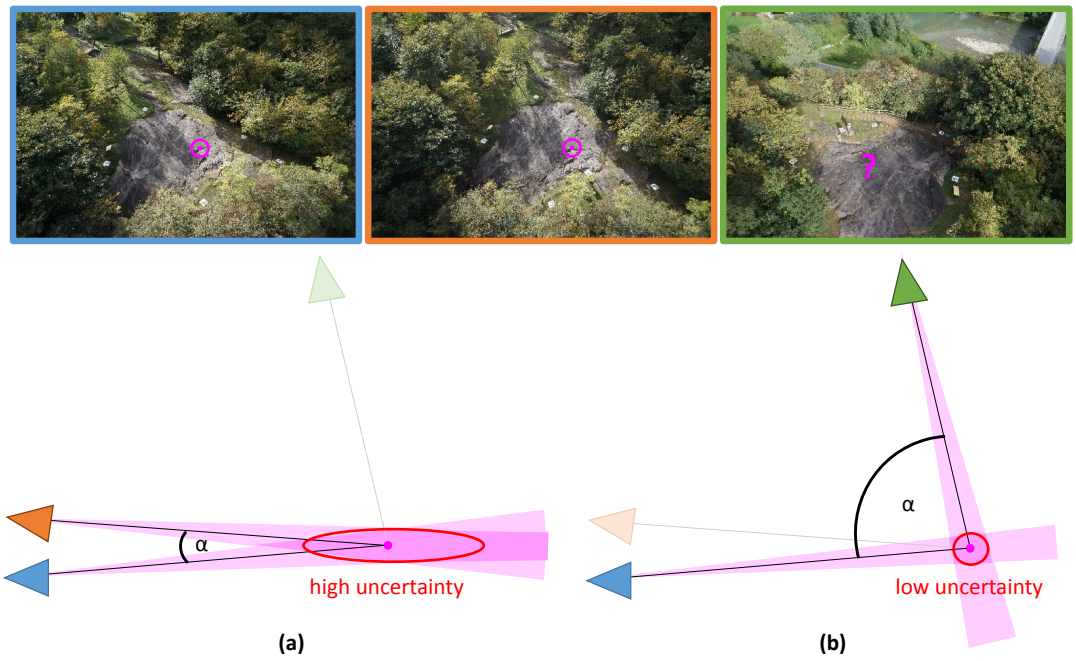


Figure 2.4: Triangulation Angle Trade-off. This example illustrates an intrinsic problem of 3D reconstruction from 2D images in relation to the triangulation angle α. The goal in this example is to estimate the 3D position of the point marked in cyan. If we assume that a correspondence can be located in image space with a known uncertainty, this induces an uncertainty cone in 3D (shown in light cyan). If we now assume that a correspondence between 2 images can be correctly found, then a large triangulation angle (b) leads to a lower 3D uncertainty, than a small triangulation angle (a). However in practice, it is more likely to find a correct correspondence for images with a similar view point (blue and orange image) than for images with dissimilar view points (blue and green image).

This basic property stems from the projective nature of a camera and the epipolar geometry. In order to infer 3D information, the camera has to observe the same scene from different view points. If we assume that a correspondence/2D measurement can be localized with a specific uncertainty distribution in image space (e.g. a 2D Gaussian), then each 2D measurement induces cone-like distribution in 3D. If we now intersect two such distributions in 3D, then the overlapping volume (which is related to the 3D uncertainty) becomes smaller if the triangulation angle becomes larger. We visualize this property in the bottom row of Figure 2.4. This property can be seen as a direct opponent to feature matching. While the 3D uncertainty declines with increasing triangulation angle, the resulting view point change also makes the feature matching (i.e. finding correspondences) increasingly more difficult. For this

work, the knowledge of this triangulation trade-off is very important, as we want to concurrently minimize the 3D uncertainty and maximize the number of correct correspondences. However, let us first define the main task of this work, i.e. Multi-View Stereo.

2.2 Multi-View Stereo

Given a set of images with known intrinsic and extrinsic calibration, the goal of Multi-View Stereo (MVS) is to densely extract accurate 3D information for as many pixels as possible. To achieve this goal, all MVS algorithms use some kind of similarity measure (e.g. Census Transform [141] or Normalized Cross Correlation) with some optimization strategy. However, there are several issues that make the MVS optimization very challenging.

First of all, the search space is very large. For each pixel in an image the corresponding pixel in another image can lie along the epipolar line. This means that the number of times, which a brute force approach would have to evaluate the similarity measure, is in the order of $\mathcal{O}(s^3 \cdot n^2)$, where s is the average image side length and n is the number of images in the dataset (this assumes that the average epipolar line length is directly related to image dimensions and that the similarity measure is evaluated with a constant step size in pixels along the epipolar line). For medium sized MVS datasets (such as the Citywall dataset [30], this would mean in the order of 10^{16} similarity score evaluations (with $4000^3 \cdot 500^2 = 1.6 \cdot 10^{16}$), which would require many petabytes of data just to store the resulting scores.

The next problem is that, on the one hand, it is desirable to use all available visual information in the image, on the other hand, MVS algorithms have to be robust to various degrees of image noise and illumination changes. As a consequence, there will be many local optima along the epipolar line and the global optimum along the epipolar line does not necessarily represent a correct correspondence. This whole process is further complicated by the fact that there is a significant number of pixels, where no correspondence to another image can be found, as the corresponding scene part is not visible in the other image due to an occlusion. However, prior to the MVS reconstruction, it is unknown which scene parts are occluded in each image.

To solve this task despite these challenges, many approaches have been proposed, each with its own set of assumptions, where each assumption has a direct influence

on the reconstruction performance in certain environments and conditions. In the following, we review the most relevant MVS approaches and work out the main underlying assumptions.

One approach to MVS is to directly densify the sparse SfM point cloud. This approach is used by PMVS [32] and MVE [30]. The idea is to use the sparse point cloud as seed points for an MVS region growing process. I.e. Starting with the depth value of a sparse 3D point (i.e. a feature point projected into a specific image) a local optimization tries to fit a planar patch to this point, yielding an improved depth estimate and a surface normal. Depth and surface normal are then propagated to the neighboring pixels, where they are optimized once more. Within the optimization, great care has to be taken that views, where the current scene part is not visible due to occlusions, do not have a negative influence on the optimization.

The underlying assumptions of this approach are the following. First, it assumes that all scene parts can be reached along a smooth surface from at least one sparse 3D point. Second, it assumes that the scene surface is sufficiently salient to allow a stable local optimization. Third, it assumes the region within the optimization window (MVE uses a 5×5 and PMVS a 7×7 pixel window) can be approximated with a plane. Fourth, it assumes the occlusions can be identified by bad similarity scores (as a similarity measure, MVE uses the Sum of Squared Difference (SSD) with an estimated scaling factor and PMVS the Normalized Cross Correlation (NCC)). Note that this directly assumes that all surfaces have Lambertian surface properties and that specular highlights are rejected as outliers.

Another popular MVS approach is PatchMatch (e.g. [35, 115, 142]), which also uses the idea of local patch probagation, but does not use the sparse points as initialization. PatchMatch was first presented for structural image editing [7] and then for two-view stereo [11]. The basic idea of PatchMatch is to initialize each pixel with a random surface patch (i.e. a random depth value with random orientation). Then the algorithm sweeps across the image (e.g. [11] sweeps from the top left corner in row-major order to the bottom right corner and then in the reverse direction), and checks if propagating the current patch to the neighbor improves the similarity score. After the probagation the patch is locally optimized by iteratively adding random noise to the patch parameters and checking if the score improves.

The currently best performing PatchMatch approach is COLMAP [115]. This approach is highly engineered and contains a large range of heuristics and carefully

chosen parameters to make the optimization process more robust (each one with its own assumption). Thus, we only focus on the most prominent assumptions.

First, the use of local planar patches and the local probagation induces a local planarity assumption and typically requires quite large patches (e.g. [142] uses 15×15 pixels, COLMAP 11×11) for a stable optimization. This leads to a very strong planarity assumption, which COLMAP mitigates by using a bilaterally weighted NCC as a similarity score. This means that pixels farther from the patch center and pixels with a significantly different color from the center pixel do not have so much influence on the similarity score. The second assumption is that the scene surface is sufficiently salient to allow a stable local optimization. Third, there is an assumption about the expected scene depth range to constrain the random initialization. This range is typically set with the help of the sparse point cloud. Fourth, COLMAP contains a weak assumption that occlusions can be detected by a bad similarity score. In the optimization, COLMAP does not explicitly detect occlusions and rejects them, however, it guides the view selection process for a given pixel such that dissimilar patches are less likely to be contained in the optimization process. Note that COLMAP produces very accurate and complete results, however, for reaching this performance it requires a lot of memory and computation time. In the paper, they always load as many images as fit into their 48GB GPU memory (i.e. $4\times$ Nvidia Titan X) and the generation of a single depth map still takes 70s with this powerful setup.

Another approach to MVS is Space-Sweep [17] (aka Plane-Sweep). Plane-Sweep discretizes the search space by sampling planes in 3D, which allows this approach to sweep through the 3D space and compute the similarity measure for each plane and image pair. The standard version uses only fronto-parallel planes, however, also slanted planes are possible [37]. The biggest problem of this approach is the plane sampling. If many planes are sampled, the problem soon becomes computationally intractable (especially if multiple plane rotations are considered), however, if too few planes are sampled, most scene parts are not well represented. Thus, Gallup et al. [37] try to recover good sweeping directions by finding dominant perpendicular structures in the sparse point cloud. This works well in human-made environments, where walls are typically perpendicular to the ground and perpendicular to each other, but can lead to problems in other environments. The final depth is typically extracted with a winner-takes-all strategy [37, 48, 51, 53, 63] over the sweeping

space.

Plane-sweep has many assumptions. First, it has a strong planarity assumption. Second, it assumes that the sampled planes have a similar orientation to the object surface. The impact of these two assumptions on the reconstruction performance depends on the used patch size and is thus stronger for some approaches (e.g. [51] uses 32×32 patches) than for others (e.g. [63] uses 5×5 patches). The drawback of very small patches (especially in conjunction with a winner-takes-all strategy) is a very high outlier rate or, if they are removed, a low completeness. This is also related to a strong local saliency assumption made by the winner-takes-all strategy. Thus, Plane-Sweep is typically used in conjunction with a very outlier robust, global surface extraction method [53, 63] as post-processing step to remove the remaining outliers and fill in gaps in the reconstruction. Despite this expensive post-processing step, this kind of approach (represented through CMPMVS [62]) does not reach state-of-the-art performance on challenging datasets such as the ETH3D dataset [116].

The only MVS approaches, which are currently rivaling the performance of COLMAP, are based on Semi-Global Matching (SGM) [54] (further referred to as SGM-MVS). Note that SGM is actually a two-view stereo method. In the field of two-view stereo, it is one of the most dominant approaches, because it delivers very accurate and complete results in a very efficient way.

The key idea of SGM is to accumulate matching costs along scan lines in different directions. The matching cost depends on the similarity score and depth changes along the scan line (i.e. staying on the same depth does not incur additional costs, while changing the depth does and strong changes incur more costs than small changes). Two very prominent MVS approaches are based on SGM, i.e. SURE [108], which is a state-of-the-art commercial photogrammetric software, and LTVRE [70], which is a closed source implementation that was already evaluated on the ETH3D benchmark [116], where it slightly outperforms COLMAP.

LTVRE [70] uses the original SGM (with hierarchical mutual information [54] as similarity score) to generate depth maps and then fuses the output in 3D using an octree structure. SURE [108], on the other hand, uses a slightly different version of SGM (with Census Transform [141] as similarity measure) on multiple scale levels for increased efficiency. Further, SURE fuses the SGM depth maps in the camera coordinate system of each undistorted image. Note that the individual SGM depth maps are not computed in the coordinate system of the undistorted image, but in

the coordinate system of the rectified stereo images, which is different for each pair of images (as the cameras are rotated such that the epipolar lines correspond to the x-axis of the images). However, the fusion in the coordinate system of the undistorted image allows SURE to remove outliers and improve the accuracy of the geometry without having a full 3D representation at this stage (which is more efficient than the octree fusion of LTVRE).

The assumptions made by SGM-MVS are the following. First, SGM-MVS has a local planarity assumption, as it uses patches for computing the similarity scores. Second, SGM-MVS has a strong bias towards fronto-parallel planes in the coordinate system of the rectified stereo images. This bias is contained in the similarity score computation and the SGM regularization (where staying on the same depth level does not incur additional costs). Third, the SGM regularization leads to a strongly planarity assumption for weakly textured regions. I.e. if a region does not contain a lot of visual information SGM tends to fit a plane in between points with more visual information. On the one hand, this means that sometimes a plane is fit inappropriately, on the other hand, this means that the local saliency assumption is not as strong as in other approaches, which leads to a higher completeness in many scenarios.

Summary of MVS Assumptions. In the following, we summarize the most important assumptions shared by all previously mentioned MVS approaches. First, all of them share a local planarity assumption. This assumption reflects in the use of image patches for the similarity score computation as well as in the schemes for depth value probagation and regularization. Second, all of them have an assumption of sufficient local saliency in their optimization schemes. Third, all of them have an assumption that most surfaces have Lambertian reflection properties and no approach has an explicit model for reflective or partially transparent surfaces. Fourth, all of them contain the implicit assumption that all relevant parts of the scene are completely static. This assumption is deeply rooted in MVS through the use of one single camera pose for each image. For reconstructing dynamic objects, multiple camera poses (one for each moving object) would be required to use epipolar geometry for structure inference (see e.g. the work of Ranftl et al. [105] for monocular video sequences).

Implications. While these assumptions work well for most scene structures, all of these assumptions can (and are) violated in real world environments, which leads to outliers or holes in the 3D reconstruction (see Figure 1.1). The exact point when such a failure occurs is not simply described and depends on many factors.

Let us explain this difficulty using the example of the local saliency assumption. While local saliency is very important, the moment when the saliency of region is too low is not clearly defined. On the one hand, you can have an object without any image gradient, which would correspond to no local visual information on the whole object surface. However, if this object occludes a scene structure to which it strongly contrasts, the location of the occluding object can be very well-localized. On the other hand, there can be image regions with a very strong image gradient, which would typically indicate a lot of visual information. However, this strong gradient might correspond to image noise (e.g. a lens flare) or could be caused by very thin structures. Thin structures are typically problematic for MVS algorithms as the local planarity assumption is violated. However, if the background behind the thin object stays similar, the similarity score will still spike at the correct correspondence. If the MVS algorithm can then actually interpret this spike correctly strongly depends on the inner workings of the MVS algorithm with all its assumptions and parameters. E.g. a simple change in the regularization strength might make one object visible in one part of the scene, but cause a lot of wrong matches in other parts.

This motivated us to develop a framework, which allows us to learn and predict this kind of failures for specific MVS algorithms (with fixed parameters) with respect to all present scene structures in specific environments. We call this prediction process *Multi-View Stereo Confidence Prediction*.

2.3 Multi-View Stereo Confidence Prediction

In this work, we introduce a new concept, which we call Multi-View Stereo (MVS) confidence prediction and we define its goal as:

Definition 2.3.1. Given a coarse scene reconstruction and a 2D image, the goal of MVS Confidence Prediction is to predict the chances of a successful depth estimation with a specific MVS algorithm and a specific camera constellation for each pixel of the image before the MVS algorithm is executed.

To the best of our knowledge, we are the first to attempt such a confidence prediction task for MVS. However, there exists a great deal of work in the field of two-view stereo vision, which aims to separate correct depth measurements from incorrect ones for an already computed depth map. These works fall under the concept of two-view stereo confidence measures, which we review in the next paragraphs.

Confidence Measures. Confidence measures have been around in the field of two-view stereo for more than a decade. For a given depth hypothesis at a certain pixel location, a confidence measure estimates the likelihood of the depth hypothesis being correct. A confidence measure is typically computed using image intensities, disparity values and/or matching costs. Early works in this field are purely hand-crafted and surveys about these traditional confidence measures can be found in [20, 21, 60]. In the simplest way a confidence measure can be used to remove very likely wrong measurements from the depth map. This process is called sparsification. The most common way for sparsification without training is the left-right consistency check [60], which can be seen as a binary confidence measure. While this check already detects many outliers, it cannot detect errors caused by a systematic problem of an approach (e.g. foreground fattening).

With the increasing success of machine learning, also confidence measures started to benefit from this trend. Haeusler et al. [46] showed that ensemble learning of many different hand-crafted confidence measures with random decision forests (RF) can significantly improve the sparsification performance. Note that confidence measures are also learned in similar fashion in the domain of optical flow, e.g. [41, 77]. Spyropoulos et al. [125] used a RF-based confidence measure as a soft-constraint in a Markov random field to improve the stereo output. In the work of Park and Yoon [97], a RF-based confidence measure is used to modulate the matching cost of a semi-global matcher (SGM) [54] and thus increase its performance. Poggi and Mattoccia [101] integrated a RF-based confidence measure in the SGM cost aggregation to reduce streaking artifacts and the memory footprint.

More recent works then shifted from random forests to more powerful convolutional neural networks (CNNs). Thus, Poggi and Mattoccia [102] propose a purely CNN-based confidence measure with the raw disparity map as input, whereas Seki and Pollefeys [120] propose a CNN-based confidence measure with two channel input (i.e. disparity patches of left and right image). In [103], Poggi and Mattoccia

show that the performance of confidence measures (traditional and machine learning based) can be further boosted by training a CNN on top of the confidence output of another confidence measure to enforce local consistency. That local consistency plays an important role for confidence measures was also observed by Kim et al. [68], who boost the performance of RF-based confidence measures by using super-pixels. In the very recent work [104], Poggi et al. present a quantitative evaluation of 52 state-of-the-art confidence measures. This evaluation shows that learning based approaches consistently outperform hand-crafted confidence measures and that CNN-based approaches lead to a better performance than RF-based approaches – provided that they have sufficient training data. And here lies one of the big problems of learning based confidence measures, as the necessary training data is very hard and costly to obtain.

Training Data Generation for Confidence Measures. Previous approaches used three main sources of training data. The first source is manual labeling. While this is the traditional approach in the fields of classification and segmentation (e.g. [25, 123, 131]), it requires hundreds of man-hours even in 2D. Because the task becomes even more taxing in 3D, only very few manually labeled datasets exist in this domain (e.g. [72]). The second source is synthetic data generation [13, 15, 79, 99]. While this kind of data shows great potential for initializing the weights of a CNN, fine tuning these weights with real data leads to a significant improvement in performance [79, 130]. The third source is to record ground truth data with active depth sensors, which is currently the most popular source [42, 81, 112, 117, 126]. If a projector based setup is used [112], the ground truth can achieve a very high accuracy, but the data acquisition takes a lot of time and is restricted to indoor scenes. For outdoor scenes the method of choice is typically the use of a laser scanner [42, 81, 117, 126]. Aside from requiring a non-trivial registration between the laser reconstruction and the recorded images, this method is also subject to a range of assumptions itself. This fact makes a manual removal of obviously incorrect ground truth data necessary for outdoor datasets [42, 81, 117]. Some approaches, like [81], combine these three sources. They combine active sensing with synthetic car models and manual annotation to increase the quality of ground truth data.

None of these methods shows good scaling properties in the sense of required man-hours per training data. Thus, we present a novel way [92] to generate training

data for confidence measures by means of self-supervision.

While ours was the first work on self-supervision for confidence measures, recently Tosi et al. [132] presented an alternative approach. The main idea of their work is to use a set of carefully selected hand-crafted confidence measures for training data generation. If at a certain pixel location all traditional measures agree that a measurement is correct or incorrect, then this pixel is used for training (either as positive or a negative sample). One drawback of this approach is that it strongly biases the samples towards the hand-crafted confidence measures. The main advantage of this approach over ours is that it only uses a single stereo pair and can thus handle static and dynamic scene parts equally. While this is a great advantage for dynamic scene parts, it also prevents the approach from using multiple view points for obtaining additional information about static scene parts to uncover systematic problems of a stereo algorithm. E.g. if a stereo algorithm consistently fattens foreground objects, then a single view point is insufficient to uncover this problem, while multi-view consistency has the potential to detect this problem if the camera motion is favorable. Another drawback of this method, with respect to MVS, is that it requires access to the cost volume. However, a huge bulk of MVS methods do not have an explicit cost volume (e.g. [32, 35, 45, 115, 142]), which precludes this approach from being used for MVS Confidence Prediction.

Supervision for Sparse Feature Matching. Aside from stereo vision, there exist some works that deal with learning the matchability of features. Some of these works [12, 47, 134] use ground truth data collected by [12]. To generate the ground truth data they use the multi-view stereo reconstruction algorithm provided by Goesele et al. [45] and trust this approach to be accurate enough. The problem with applying this approach to dense stereo is that a learning algorithm will try to tune its output to reproduce any systematic error made by [45]. Philbin et al. [100] use SIFT [76] nearest-neighbors together with a RANSAC verification to generate negative and positive training data, whereas Simonyan et al. [124] first compute a homography between images using SIFT and RANSAC and then establish region correspondences using the homography. Hartmann et al. [52] learn the matchability of SIFT features by collecting features that survive the matching stage and those which are rejected as positive and negative training data. All of these approaches focus on a specific type of sparse feature and do not generalize well to dense stereo.

Image-based Supervision for other Tasks. In the last two years, the data hunger of CNNs has spawned a lot of research on alternative ways of supervision for many different tasks. For training a CNN for single view depth estimation, Garg et al. [39] use pairs of sequential images. In one of the images they estimate the depth and in the second image (which has a known orientation to the first image) they then evaluate the photometric error. Godard at al. [44] use a similar idea but with calibrated stereo images, where they also enforce left-right consistency between the images. Yu et al. [140] also use a photometric error for training a CNN for optical flow. Based on an (oriented) image sequence, they combine a photometric loss with a loss that enforces spatial smoothness in the flow field. Pathak et al. [98] use videos of moving objects to learn object based features that can then be used for object detection. Long et al. [75] use image sequences to first train a CNN on the task of image interpolation (i.e. for 3 given consecutive frames, they train the network to reconstruct the middle frame from the outer two). The resulting CNN is then used for optical flow estimation by inverting the network and tracing how it would interpolate the two images. While these approaches work well for their individual tasks, they cannot be easily generalized to MVS Confidence Prediction.

2.4 View Planning

Next-best view (NBV) planning, view planning, sensor placement, path planning for reconstruction and/or coverage, visual inspection and exploration are all closely related topics. They all have to answer the questions "what parts of the scene are already sufficiently covered?" and "how can I best improve this coverage?". In this sense they are all related to the art gallery problem [96] or the coverage problem [129], which both have been shown to be NP-hard.

Thus, researchers from the communities of photogrammetry, robotics and computer vision have developed and are still developing methods which are honed to very specific tasks exploiting all available domain specific knowledge to conquer this challenging task with one specific actuator/sensor setup. Some works (e.g. [2, 3, 26, 133]) thus focus on the reconstruction of small scale scenes in laboratories. For this kind of approaches we refer the reader to a very recent review article of [67]. Other works (e.g. [29, 93, 107, 111]) include the feature richness into the path planning to ensure good visual localization. With the increasing avail-

ability of robotic platforms, also the research in the field of structure inspection drastically increases (e.g. [4, 9, 10, 23, 33, 55]). A review of current approaches on this topic can be found in [5]. This field is also closely related to coverage search (e.g. [19, 34]). For photogrammetry, there also exist several works on view planning for UAVs (e.g [58, 64, 65, 78, 91, 106, 113]). Most works in the field follow the same paradigm. Under the assumption that they have a good 3D approximation of the scene, they first sample an over-complete set of views. Then they try to select a small subset of these views such that some quality or coverage function is sufficiently fulfilled. Finally, they have some path optimization algorithm to solve the remaining path planning problem.

As input for the planning procedure Martin et al. [78] use a rough terrain model (with 10m resolution), Schmid et al. [113] a digital surface model (DSM), Jing et al. [65] use a 2D map and manually extrude the building height, while Hoppe et al. [58] assume that they have a surface mesh of the scene of interest. All these approaches assume that a sufficiently good 3D model is given prior to the actual 3D reconstruction, which is typically not the case. In this regard, the very recent work of Roberts et al. [106] is more general. Roberts et al. [106] first let the drone fly an ellipse around the scene of interest, then land the drone and run a full multi-view stereo pipeline on the acquired set of images. The resulting 3D reconstruction is then used for planning. The problem with this two step approach is that if the geometry is insufficiently covered in the first flight, the 3D model will be wrong and consequently the computed view plan will be suboptimal with respect to the real geometry.

For sampling potential views each approach follows its own strategy. Martin et al. [78] sample randomly over a region of interest, while Hoppe et al. [58] sample one view for each triangle in the mesh at a fixed distance. Schmid et al. [113] sample potential views in a regular pattern on the hull around the scene of interest looking towards the geometry. Roberts et al. [106] sample on a regular grid with regular orientations and then use a greedy algorithm to fix the view rotations. Jing et al. [64] first randomly sample positions with a buffer zone around the geometry and then set the orientation based on a geometry force field such that the views look towards the closest geometry.

After sampling, the next task is to reduce the number of views and then connect them to obtain the final view plan. Thus, Hoppe et al. [58] greedily select the ones

that maximizes their objective function. In their objective function they consider the 3D uncertainty and the camera overlap. They terminate if 95% of the scene is covered and optimize the path length with a greedy algorithm. Schmid et al. [113] select a subset of the sampled views based on a heuristic such that the resulting views are not redundant (considering what parts of the DSM they observe and under which angle). Then they minimize the path length with a Farthest-Insertion-Heuristic. Martin et al. [78] optimize the scene coverage with a genetic algorithm and then find the global optimal solution to the remaining traveling salesman problem with a binary integer linear program. As this problem is NP-hard, they only use a very small number of views in their experiments (i.e. 30 views). Jing et al. [65] formulate the view planning problem as a set covering problem, which they optimize with a greedy algorithm. Roberts et al. [106] set themselves apart from other works by proposing an additive approximation for the scene coverage function. With this approximation they are able to formulate the view planning problem as an orienteering problem using a submodular objective function. An orienteering problem tries to solve a knapsack problem and a traveling salesman problem at the same time. In the case of view planning, this means that the algorithm tries to maximize the coverage considering the available budget in terms of flight path length. All works mentioned above focus on rather simple structures (i.e. mostly houses or well-textured flat terrain), which are quite easy to reconstruct with MVS. However, in our work [91] we show that for more challenging scene structures the image constellation together with the used MVS algorithm (with all its underline assumptions) play an important role for the MVS reconstruction success. Thus, we use this knowledge (which is encoded by a MVS Confidence Predictor) to improve the image acquisition process. In particular, this also allows us to avoid constellations that do not lead to the desired outcome. One good example in this regard is a grass surface. From far away, grass is close to random texture and can be easily reconstructed, however, if the distance to the camera becomes too small then some approaches won't be able to reconstruct this part of the scene at all. Our approach can detect such circumstances during the image acquisition and thus enables a reliable reconstruction even in challenging environments.

Aside from image acquisition, view planning is also related to the second application presented in this thesis, i.e. the MVS Prioritization. Given a fixed set of images, the aim of our MVS Prioritization is to establish an ordering of images

(with a suitable set of other images for matching) such that any subset of the ranked entries from entry 1 to n retains a maximum of information about the scene for any n. While this task shares many aspects with view planning, the problem tackled in this task also shows one important difference: The set of images is fixed. On the one hand, this is an advantage as subtasks such as path planning do not need to be handled anymore. On the other hand, if the image constellations are suboptimal then there is no possibility to improve the situation.

From all the works mentioned above, we see the works of [59] and [80] as most related to the task of MVS Prioritization. Both approaches try to obtain a maximally complete reconstruction from a given image set by iteratively updating the geometry estimate. For updating the geometry, [59] use the complete PMVS algorithm [31] after each added view, while [80] execute the stereo algorithm of [136] after selecting the next-best stereo pair. We contrast from these works in several points. First of all, our aim is not an iterative update scheme with the MVS algorithm in the loop, but instead we aim to rank all view clusters (i.e. key views with matching partners) according to their importance for the final reconstruction. On the one hand, we are thus able to obtain a complete reconstruction with a fraction of available view clusters. On the other hand, we can pre-compute the whole key view ranking, which allows us to maintain the natural parallelization capabilities of MVS based on depth maps in the execution phase. Instead of updating the geometry estimate in each iteration, we use all measurements of the sparse reconstruction to obtain a high quality surface mesh at a low computational cost once at the start of our algorithm. Through this formulation, we combine the advantages of NBV planning and MVS based on depth maps, i.e. data reduction and parallelism. This leads to a light-weight approach, which can be easily integrated in most photogrammetric pipelines to increase the overall efficiency.

2.5 Matching Partner Selection

Most MVS approaches based on depth maps formulate some kind of heuristic to select the k best matching partners for each key view to increase the efficiency of MVS. While this significantly decreases the run-time, the size of the resulting point cloud stays approximately the same, as each image is considered as a key view. The heuristics for matching partner selection strongly depend on how the images

are acquired (structured versus unstructured) and the requirements of the MVS algorithm. If the images are acquired in a regular grid, the k closest images are a natural choice to maximize the completeness. For more unstructured settings, this technique can be problematic as the images can look into different directions. In an unstructured setting, the connectivity in the sparse reconstruction (i.e. how many sparse 3D points are shared between two cameras) is typically a more reliable cue to determine if the MVS matching step will work or not. However, in an unstructured setting it can also happen that many images are acquired from nearly the same position (this also happens if panoramic image acquisition techniques [137] are used). In such cases, taking the most connected images as matching partners will very likely lead to insufficient parallax and a very high depth uncertainty. Thus, Goesele et al. [45] combine the connectivity with geometric constraints in a greedy fashion. Their formulation down-weights connections (shared features) with a triangulation angle below 10° and dissimilar scale. Additionally to these two terms, Bailer et al. [6] also add a coverage term, which favors connections that have not been covered by other selected images. Shen et al. [121] use a formulation without connectivity only based on the geometric constraints on the triangulation angle and the distance between images. For very small datasets, where all images nearly see the same part of the scene (as in the DTU dataset [1]), also random selection of matching partners can lead to good results [35]. Note there also exist works that select suitable matching partners on a per pixel basis during the reconstruction also based on hand-crafted constrains [115, 138, 142, 143]. Of all formulations mentioned above, the formulation of Bailer et al. [6] seems to be the closest to our approach. However, their approach has two main drawbacks. First, they use a hand-crafted model for the triangulation angle, which is specifically tuned for their own PatchMatch-based approach [6] for the reconstruction of flat, human-made environments. Second, they use the sparse point cloud for evaluating the coverage. The problem with this approach is that it strongly biases the algorithm to well-textured parts of the scene, which typically generate a lot more sparse points than weakly textured scene parts. This can be problematic as many important scene parts (such as building facades) are often completely unrepresented in the sparse point cloud, although they very often contain sufficient micro-texture for a reliable MVS reconstruction.

2.6 Summary

In this chapter, we first tried to make the reader aware of the differences between a real physical camera and the pinhole camera model, which is used in nearly all photogrammetric pipelines (Section 2.1). The main point of this comparison was to show that real physical cameras have many limitations, of which the most important ones (such as the lens distortion) are modeled, while others (such as vignetting, chromatic aberration or depth of field) are typically left unmodeled and are assumed to be small enough to be handled by robust algorithms.

Then we gave a broad review over current MVS approaches and discussed their underlying assumptions (Section 2.2). The main point of this discussion was to show that all approaches share a similar set of general assumptions (including a static environment, Lambertian reflectance, local planarity and local saliency), but also differ a lot in how they use this assumptions and how strongly they depend on them. As the impact of these assumptions does not only depend on the algorithm, but also the complete set of parameters (including e.g. the regularization weights), it is exceedingly hard to tell why sometimes one algorithm is able to obtain a complete and accurate reconstruction of one object, while another fails (see e.g. the ellipses in Figures1.1).

This motivated us to learn to predict such failures such that we can detect and avoid unfavorable image constellations. However, the lack of suitable supervision (in form of reliable reference data) made it necessary to think about alternative ways of supervising the learning procedure. On this aspect, we gave an overview (Section 2.3) over many fields in computer vision (including confidence measures for two-view stereo, sparse feature matching and single image depth estimation), which all thirst for cheap and scalable supervision in form of automated processes. In this regard, we did not find a single approach which would suit our purpose of predicting MVS failures.

For our two applications (i.e. Automated Image Acquisition and MVS Prioritization), we then reviewed the field of view planning (Section 2.4) and matching partner selection (Section 2.5). The main conclusion of this review is that in both fields nearly all aspects are hand-crafted and tuned towards one specific application, in one environment, with one specific MVS algorithm due to the difficulty of the underlying tasks. No approach (that we are aware of) in these fields has yet attempted

to learn the requirements of MVS and use this knowledge to actively control the set of images with learned likelihood functions rather than hand-crafted models.

Chapter 3

Confidence Learning without Ground Truth

In this chapter, we present our fully automated approach for confidence learning, which can be used for two-view stereo [92] as well as multi-view stereo (MVS) [91]. The main idea of our approach is to use many independent 3D reconstructions to separate *consistent* 3D hypotheses from *contradicting* 3D hypotheses. With "independent 3D reconstructions", we mean 3D reconstructions produced with the same algorithm from independent image sets, which view the scene from significantly different view points. This approach is to some extend related to depth map fusion (see e.g. [82]), but instead of outputting an improved depth map, we generate pixel-wise training data for machine learning. Another significant difference is that we can also refrain from using pixels as training data, in case we are not able to assess if the measurement is correct or not. Care has to be taken in this assessment to keep the number of false positives (consistent but incorrect) and false negatives (inconsistent but correct) as low as possible. This means that systematic errors, which are consistent to each other, should not be taken as positive training samples and also not lead to correct measurements being classified as negative samples. We mitigate this problem in ensuring that only reconstructions from significantly different view points (i.e. with large relative motion) are respected for positive support assessment. This restriction has shown to be useful, as the systematic errors (e.g. edge fattening) of independent reconstructions with a large relative motion tend to appear at different locations. In the following, we give an overview of our automated

learning procedure and then outline all important steps in more detail.

3.1 Overview

As input our approach only requires an unordered set of images, viewing one or multiple scenes from many view points. With this input, we then estimate the camera poses and extract a sparse point cloud with Structure-from-Motion (SfM) (see Figure 3.1). Now instead of reconstructing the whole scene with all available information, we split the cameras into small clusters and run the query MVS algorithm on each small cluster.

Then we use the resulting MVS reconstructions to benchmark the quality of each other. The goal is to find consistent measurements, which have a high probability of being correct, and inconsistent measurements, which have a high probability of being incorrect. To achieve this, we propose an approach with three stages.

The first stage has the purpose to identify parts of the reconstruction, which are consistent over multiple, significantly different view points. This assessment is important, because otherwise systematic errors made by the approach tend to accumulate if many images were taken from similar view points.

In the next stage, the resulting highly consistent measurements of the reconstructions vote for the consistency of all other measurements. This results in many labeled training samples (negative and positive), however, a lot of "easy" negative samples are also left out, because they are not on the line of sight of the highly consistent measurements.

In the last stage, we uncover these outliers in a second voting scheme, which is less restrictive and lets all measurements vote. Note that this less restrictive voting scheme can only result in negative training samples. The last step of our approach also has the purpose to identify "missing measurements"; i.e. measurements that would have been geometrically possible, but for some reason the MVS algorithm failed to generate them. We achieve this by comparing the MVS depth maps to a fused depth map and a surface reconstruction derived from the sparse point cloud. This leads to more negative samples, which can then be used for training.

With all collected training data, we then train a random forest. The advantage of a random forest over CNNs is that it allows us to store meta data with the training samples. For a query sample, the stored meta data allows us to extract

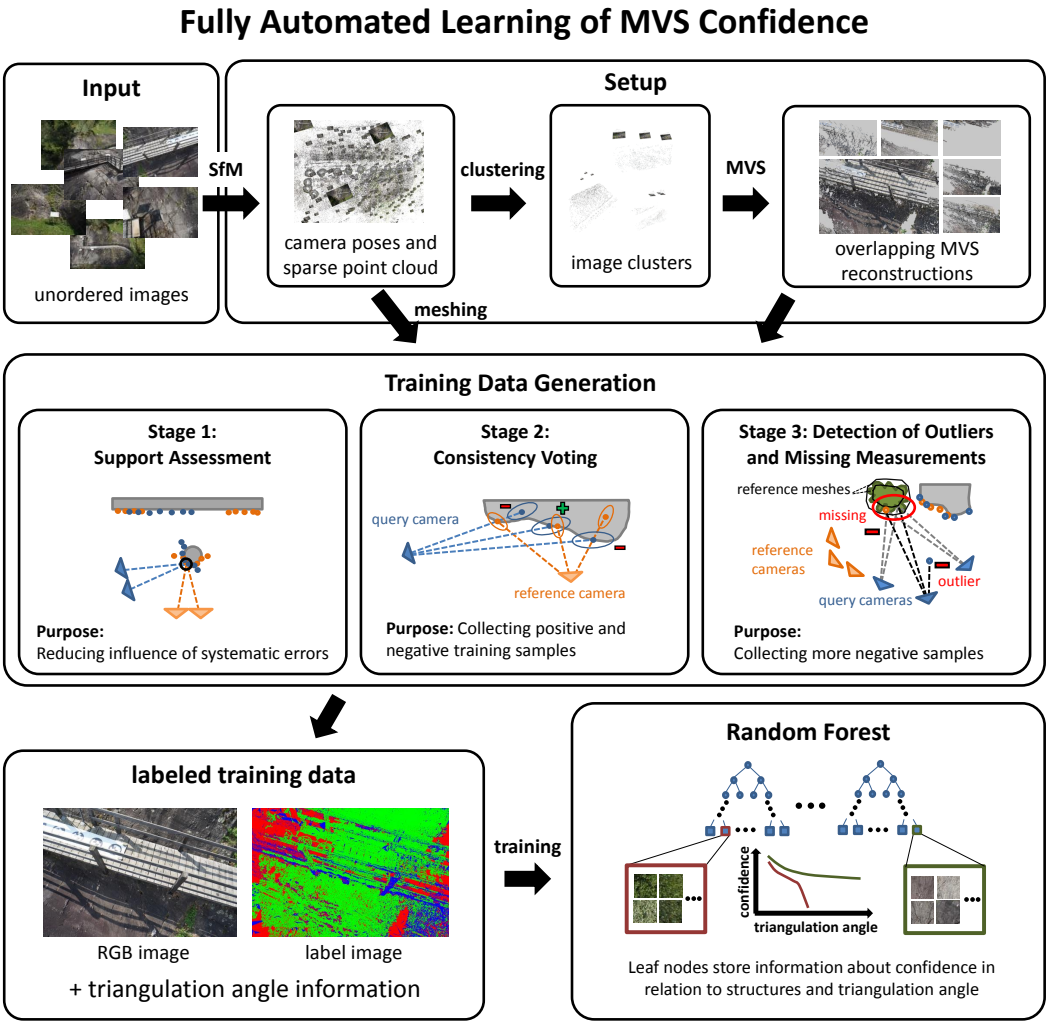


Figure 3.1: Fully Automated Confidence Learning as proposed in [91, 92]. With a set of unordered images as input, we first perform Structure-from-Motion (SfM) to obtain a sparse point cloud and the camera poses. From the point cloud, we reconstruct two surface meshes with different object boundaries. From the camera poses, we extract small image clusters for executing a query Multi-View Stereo (MVS) algorithm. This leads to many overlapping 3D reconstructions of the same scene. With this data, we first assess the mutual support between reconstructions from significantly different view points to reduce the influence of systematic errors. Then we perform an information weighted voting process to collect positive and negative training data. Severe outliers and missing parts are then detected with the help of other reliable reconstructions and the surface meshes. This leads to labeled training data with the necessary meta data (triangulation angle information). With this data, we train a random forest, where the leaves then store the likelihood of a successful reconstruction (aka confidence) in relation to the visual information of the 3D structure and the triangulation angle.

confidence functions (parameterized over the triangulation angle) instead of just a single confidence value. These functions then allow us to separate good image constellations from suboptimal ones, which we then use to modify the set of images before the MVS is executed.

3.2 Setup

Starting with an unordered set of images, we process these images with a fully automated Structure-from-Motion (SfM) pipeline [110]. After this step, we then split the resulting camera network (i.e. camera poses with associated sparse 3D points) into small independent subsets/clusters, such that these clusters can then be used to uncover inconsistencies between 3D reconstructions obtained with the same MVS algorithm, but from different view points. For MVS, we use a cluster size of three, as this is the minimum number for most MVS algorithms. In theory one could use larger clusters, however, with each added camera it becomes harder to judge the individual influence of each camera on the reconstruction output.

As one of our aims is to predict the reconstruction confidence in dependence of the triangulation angle, we have to ensure that a large diversity of triangulation angles is present in the training data. Thus, we randomly sample camera triplets from a fixed number of triangulation angle bins as follows.

First, we randomly choose one camera. Then, we randomly choose one of t triangulation angle bins. The lowest bin starts at a minimum angle β_{min} and ranges to double that value, where the next bin then starts. Now, we randomly select a camera that shares measurements (in the sparse point cloud) with the central camera and check if the following three criteria are fulfilled: (1) The median triangulation angle is within the currently chosen bin. (2) To ensure sufficient overlap between the cameras, the area spanned by the shared measurements has to be at least greater than half the total image area for one of the cameras. (3) To avoid scale issues, the median scale ratio s_{res} between the two cameras shall not exceed s_{max}. The scale ratio of a single shared 3D point $\mathbf{p}_{\text{shared}}$ is defined as $s_{\text{res,shared}} = max\{\text{res}_1/\text{res}_2, \text{res}_2/\text{res}_1\}$ with $\text{res}_\text{x} = f_\text{x}/\|\mathbf{c}_\text{x} - \mathbf{p}_{\text{shared}}\|$, where $\mathbf{c}_\text{x}$ is the camera center and f_x the focal length.

If these criteria are fulfilled, the same check is repeated for the second camera. This generation scheme runs for a large number of iterations i and produces a set of camera triplets with a very diverse range of triangulation angles. For two-view

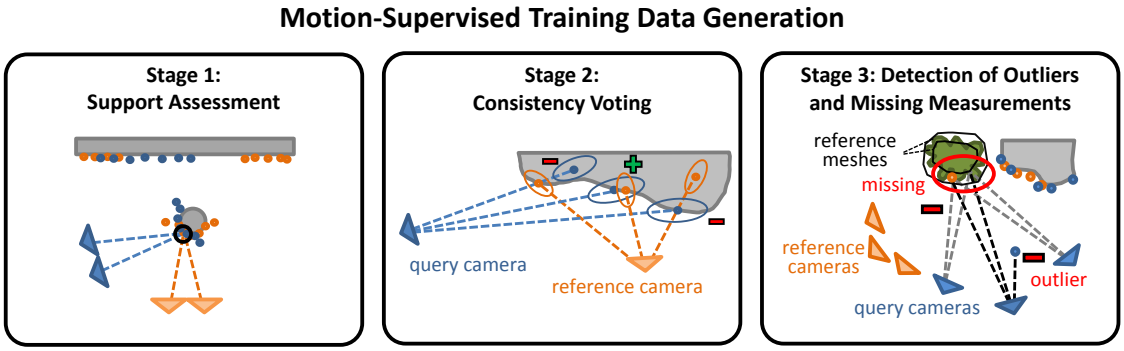


Figure 3.2: Motion-Supervised Training Data Generation. From left to right we show the three main stages of our approach. In Stage 1, we collect the support between reconstructions from far away view points. In Stage 2, we use visibility rays to collect positive and negative training samples. In Stage 3, we collect outliers and missing measurements as negative samples.

stereo, we skip the clustering step and directly treat each stereo pair as its own cluster. For each cluster, we then execute the query algorithm, which results in a set of highly redundant reconstructions of the scene from many different viewpoints.

3.3 Motion-Supervised Training Data Generation

For training data generation, we now want to use the set of highly redundant reconstructions to evaluate each other and in this way collect consistencies and contradictions as positive and negative training samples. To avoid labeling consistent systematic errors as positive training data, our approach works in three main stages (see Figure 3.2).

3.3.1 Stage 1: Support Assessment

The first stage has the purpose of reducing the influence of all consistent but incorrect measurements. In practice, we can observe that the likelihood that two measurements of independent 3D reconstructions[1] are consistent but incorrect at the same time decreases as the relative view point difference increases. Thus, we analyze how well each measurement is supported by reference reconstructions from

[1] 3D reconstructions that were produced with the same MVS algorithm from independent image sets.

a sufficiently different view point. The aim of this stage is to accumulate the support $\mathbf{supp}_{\text{query}}$ for each measurement $\mathbf{p}_{\text{query}}$ of each available depth map.

In this process of support accumulation, we have to take care that images from similar view points do not bias the voting process with the same systematic error. So before we accumulate the support of a measurement $\mathbf{p}_{\text{query}}$, we split all reference measurements that project into the same pixel of the query camera as $\mathbf{p}_{\text{query}}$ into support clusters $\mathbf{S}_{\text{ref}}$. The purpose of these clusters is to group measurements from approximately the same view point (with respect to the query measurement) together and limit the maximum support of such a reference cluster to one.

For MVS, this clustering is first done via mean shift on the logarithm of the resolution (scaled with $log(s_{\text{min}})$) and a unit bandwidth. Here we use a logarithmic space such that all relative relations have the same distance (i.e. $log(\frac{a}{b}) = log(a) - log(b) = -log(\frac{b}{a})$). Then the resulting clusters are further refined by performing mean shift clustering of the viewing direction with a bandwidth α_{min}. For two-view stereo in a street view scenario [92], we use a simpler clustering scheme as the view point variation is very restricted. In this scenario, we found that using fix-spaced angular bins for clustering is sufficient. In both scenarios, the purpose of the clustering stays the same, i.e. reference measurements of a similar view point shall only be able to express their support once.

Using the support clusters, we then look if they contain measurements that are sufficiently different from the view point of the query measurement. We treat a reference measurement as sufficiently different if the view angle difference $\alpha_{\text{diff}} > \alpha_{\text{min}}$ or the scale difference $s_{\text{res,query}} > s_{\text{min}}$ is sufficiently large. We compute these values as $\alpha_{\text{diff}} = \measuredangle(\overrightarrow{\mathbf{p}_{\text{query}}\mathbf{c}_{\text{ref}}}, \overrightarrow{\mathbf{p}_{\text{query}}\mathbf{c}_{\text{query}}})$ and $s_{\text{res,query}} = \text{res}_{\text{ref}}/\text{res}_{\text{query}}$ with $\text{res}_{\text{x}} = f_{\text{x}}/\|\mathbf{c}_{\text{x}} - \mathbf{p}_{\text{query}}\|$, where $\mathbf{c}_{\text{x}}$ is the mean camera center and f_{x} the mean focal length of a camera triplet.

If a reference measurement is sufficiently different, we have to assess if it supports the query measurement. To assess this property, we require an uncertainty model.

In this work, we use the uncertainty model explained in Section 2.1. With this model, it is possible to propagate an assumed 2D uncertainty distribution from the images into 3D to obtain a first order estimate of the 3D uncertainty via a covariance matrix. For this task, we assume that 2D uncertainty distribution is a Gaussian with unit pixel noise. For Equation 2.6, this means that $Cov_{2D} = I_2$ with I_2 as the 2D identity matrix.

Data: Query measurement $\mathbf{p}_{\text{query}}$ and a set of reference measurements $\mathbf{P}_{\text{ref}}$ that project into the same pixel of the query camera as $\mathbf{p}_{\text{query}}$. Each reference measurement $\mathbf{p}_{\text{ref}} \in \mathbf{P}_{\text{ref}}$ is part of a support cluster $\mathbf{S}_{\text{ref}}$, which groups all reference measurements with a similar view point (i.e. multiple $\mathbf{p}_{\text{ref}}$ share the same $\mathbf{S}_{\text{ref}}$).
Result: Accumulated support $\mathbf{supp}_{\text{query}}$ for the query measurement $\mathbf{p}_{\text{query}}$
$\mathbf{supp}_{\text{query}} \leftarrow 0$;
for *each support cluster* $\mathbf{S}_{ref}$ **do**
  $\mathbf{activations}_{\mathbf{S}_{\text{ref}}} \leftarrow 0$
end
for *each* $\mathbf{p}_{ref} \in \mathbf{P}_{ref}$ **do**
  if $\mathbf{activations}_{\mathbf{S}_{ref}} = 0$ **and** *view points of* $\mathbf{p}_{ref}$ *and* $\mathbf{p}_{query}$ *sufficiently different* **and** $d_M(\mathbf{p}_{ref}, \mathbf{p}_{query}) < \sigma_{max}$ **then**
    $\mathbf{activations}_{\mathbf{S}_{\text{ref}}} \leftarrow 1$;
  end
end
for *each support cluster* $\mathbf{S}_{ref}$ **do**
  $\mathbf{supp}_{\text{query}} \leftarrow \mathbf{supp}_{\text{query}} + \mathbf{activations}_{\mathbf{S}_{\text{ref}}}$
end

Algorithm 1: Stage 1: Support Assessment

Using this model, we now assess if a reference measurement $\mathbf{p}_{\text{ref}}$ is within a fixed theoretical tolerance σ_{max} of the query measurement $\mathbf{p}_{\text{query}}$. For this assessment, we use the theoretic Mahalanobis distance $d_M(\mathbf{p}_{\text{ref}}, \mathbf{p}_{\text{query}})$, which we compute as:

$$d_M(\mathbf{p}_{\text{ref}}, \mathbf{p}_{\text{query}}) = \sqrt{(\mathbf{p}_{\text{query}} - \mathbf{p}_{\text{ref}})^T \, Cov_{3D,min}^{-1} (\mathbf{p}_{\text{query}} - \mathbf{p}_{\text{ref}})}, \tag{3.1}$$

where $Cov_{3D,min}$ is the covariance matrix with smaller uncertainty (i.e. largest eigenvalue) of either $\mathbf{p}_{\text{ref}}$ or $\mathbf{p}_{\text{query}}$. Note that the two matrix inverses in Equation 2.6 and Equation 3.1 cancel each other and never have to be computed explicitly. Also note that, as we assume the distribution to be fixed and only the 3D point to be unknown, $Cov_{3D}^{-1} = \mathcal{I}_{3D}$, where $\mathcal{I}_{3D}$ is the Fisher information matrix. Thus also the largest eigenvalue of Cov_{3D} can be found as the smallest eigenvalue of $\mathcal{I}_{3D}$.

Now if the view points of $\mathbf{p}_{\text{ref}}$ and $\mathbf{p}_{\text{query}}$ are sufficiently different and $d_M(\mathbf{p}_{\text{ref}}, \mathbf{p}_{\text{query}}) < \sigma_{max}$, then we set the $\mathbf{activations}_{\mathbf{S}_{\text{ref}}}$ of the corresponding reference cluster $\mathbf{S}_{\text{ref}}$ to one. Finally, we sum up all reference cluster activations to obtain the final support $\mathbf{supp}_{\text{query}}$ for a query measurement $\mathbf{p}_{\text{query}}$. The complete first stage is summarized in Algorithm 1.

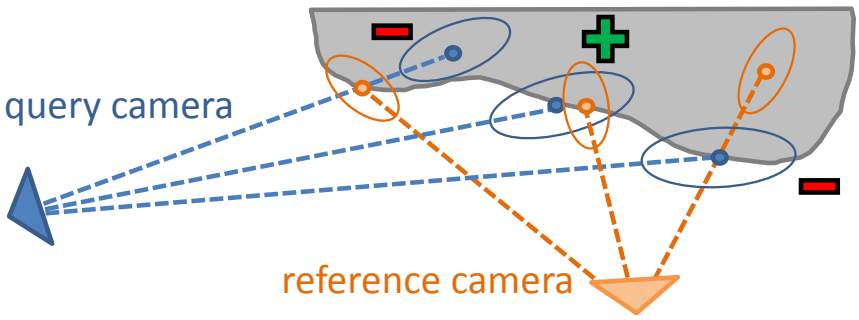


Figure 3.3: Consistency Voting. There are three possibilities for voting. A positive vote (center) is only cast if the reference measurement is within the uncertainty boundary of the query measurement. A negative vote is either cast if a reference measurement would block the line of sight of the query camera (left) or the other way around (right).

3.3.2 Stage 2: Consistency Voting

The basic idea of this stage is to let all depth maps vote for the (in)consistency of a query depth map. Similar to works in depth map fusion (e.g. [82]), negative votes are cast by free space violations and occlusions and positive votes are cast by measurements which are sufficiently close to each other (see Figure 3.3). Compared to fusion approaches, we aim for a completely different output. While works in depth map fusion try to improve/fuse the depth map, we only aim to decide which parts of the depth map cause contradictions and which parts are sufficiently consistent. Furthermore, we have to reduce the influence of systematic errors in the voting scheme, which we achieve with the support of a reference measurement computed in the previous stage. In particular, this means that only parts which have a support from at least one significantly different observation angle are eligible for voting.

For casting a positive vote v_+, a reference measurement has to fulfill two properties. First, it shall be more accurate than the query measurement. We evaluate this property with the largest eigenvalue of the corresponding covariance matrix. Second, the reference measurement has to be within a fixed theoretical tolerance of $\sigma_{\max}$ of the query measurement. For this evaluation we use the Mahalanobis distance based on the covariance matrix of the query 3D point. We define a positive vote as:

$$v_+ = \sqrt{i_{\text{ref}}} \cdot support_{\text{ref}} \tag{3.2}$$

where i_{ref} is the smallest eigenvalue of the Fisher information matrix of the reference

Data: Query measurement $\mathbf{p}_{\text{query}}$ and a set of reference measurements $\mathbf{P}_{\text{ref}}$ that project into the same pixel in either the query view or a reference view and where each reference measurement $\mathbf{p}_{\text{ref}} \in \mathbf{P}_{\text{ref}}$ has a positive support (i.e. $\mathbf{supp}_{\text{ref}} > 0$)
Result: Label for the query measurement $\mathbf{p}_{\text{query}}$ ($\mathbf{label}_{\text{query}}$)
$\mathbf{vote}_{\text{query}} \leftarrow 0$;
$\mathbf{activation}_{\text{query}} \leftarrow 0$;
for *each* $\mathbf{p}_{ref} \in \mathbf{P}_{ref}$ **do**
  if $\mathbf{p}_{ref}$ *more accurate than* $\mathbf{p}_{query}$ **then**
    if $\mathbf{p}_{ref}$ *within a theoretical tolerance of* σ_{max} *of* $\mathbf{p}_{query}$ **then**
      $\mathbf{vote}_{\text{query}} \leftarrow \mathbf{vote}_{\text{query}} + v_{+}$;
      $\mathbf{activation}_{\text{query}} \leftarrow 1$;
    else
      if $\mathbf{p}_{query}$ *causes a free space violation or occlusion with respect to* $\mathbf{p}_{ref}$ **then**
        $\mathbf{vote}_{\text{query}} \leftarrow \mathbf{vote}_{\text{query}} + v_{-}$;
        $\mathbf{activation}_{\text{query}} \leftarrow 1$;
      end
    end
  end
end
$\mathbf{label}_{\text{query}} \leftarrow$ **ignore**;
if $\mathbf{activation}_{query} = 1$ **then**
  if $\mathbf{vote}_{query} > 0$ **then**
    $\mathbf{label}_{\text{query}} \leftarrow$ **positive**
  else
    $\mathbf{label}_{\text{query}} \leftarrow$ **negative**
  end
end

Algorithm 2: Stage 2: Consistency Voting

3D point. This means that measurements with a low theoretic uncertainty get a higher voting strength, as $\sqrt{i_{\text{ref}}} = 1/\sqrt{u_{\text{ref}}}$, where u_{ref} is the largest eigenvalue of the covariance matrix and hence $\sqrt{u_{\text{ref}}}$ can be interpreted as the standard deviation along the axis of the highest uncertainty.

For casting a negative vote v_{-}, a reference measurement has to fulfill three properties. First, it also has to be more accurate than the query measurement. Second, it has to be outside the fixed theoretical tolerance of σ_{max}. Third, it has to cause a free space violation or occlusion as depicted in Figure 3.3. In a free space violation, a reference measurement would block the line of sight of a query measurement (left side in Figure 3.3), whereas the other way around would cause an occlusion (right

side in Figure 3.3). If these properties are met, a negative vote is cast:

$$v_- = -\sqrt{i_{\text{ref}}} \cdot support_{\text{ref}} \tag{3.3}$$

For each pixel in the query depth map the votes are collected. The label of a pixel with more than zero votes is then set depending on the sign of the final sum of votes. The complete voting scheme of this stage is summarized by Algorithm 2.

3.3.3 Stage 3: Detection of Outliers and Missing Measurements

The third stage deviates for two-view stereo [92] and multi-view stereo [91]. In two-view stereo, the confidence also has the purpose of identifying occlusions from already computed depth maps, whereas in the case of MVS, we do not have the final reconstruction during prediction time (only a rough scene representation). Thus in the MVS case, we only desire training samples from regions that are visible in all three images. More precisely, we have to detect if the MVS algorithm failed to produce any output in a region where it should have been geometrically possible and use this case as a negative training sample. In the following paragraphs, we first present a depth map augmentation (which has the purpose of detecting obvious outliers and is used for two-view and multi-view stereo) and then explain our extension for multi-view stereo (which uses two surface meshes for detecting missing measurements).

Depth Map Augmentation. For computing the augmented depth map, we collect all depth values of the other depth maps that would project into a pixel of the query image (Here we respect the pixel radius induced through scale change). Then we sort these depth values and search for the closest depth value which obtains a positive score in a second voting scheme. This voting scheme is very similar to the one proposed in the previous stage, but many more depth values will end up with a positive score although they are incorrect.

There are 4 differences to the other voting scheme: (1) Every depth map can vote (without accuracy restrictions), (2) the border between consistent and contradicting vote is set to $(1/\sqrt{u_{\text{query}}} + 1/\sqrt{u_{\text{ref}}}) \cdot \sigma_{\max}$, (3) $support_{\text{ref}} = 1$ for all measurements and (4) a depth value has to obtain at least three votes to be considered valid. If no such depth value is found, the original depth is kept.

Using the augmented depth map, we now treat a depth value as a negative sample if the following two criteria are met. First, the query depth value has to be smaller than the depth value of the augmented depth map. Second, the difference between those two depth values has to be larger than $\sigma_{\max} \cdot 1/\sqrt{u_{\text{augmented}}}$, where $u_{\text{augmented}}$ stands for the largest eigenvalue of the covariance matrix of the augmented measurement if we pretend that it is only visible from the query camera cluster.

MVS Extension. Additional to the augmented depth map, we use two meshes to detect scene parts which an approach completely fails to reconstruct (e.g. some approaches will simply not work on vegetation). The two meshes are created independent of the query algorithm from the sparse point cloud. We use two meshes with slightly different object boundaries to account for errors in the meshes. To construct these meshes, we first use all available images in the dataset to compute a joint sparse point cloud [109]. From this point cloud we robustly extract a surface mesh [71, 135], and then shrink and expand this mesh for our purpose. The exact implementation details of our meshing approach can be found in [90]. From this base mesh, we obtain a shrunken mesh by performing three iterations of neighbor-based smoothing. In each iteration a vertex moves half the distance to the average position of the vertices that share an edge with this vertex. For the second mesh, we expand the shrunken mesh again. For this purpose, we compute a vector by averaging the motion vectors of a vertex and its neighbors from the shrinking procedure. Each vertex is then moved twice the vector length in the opposite direction of this vector. Both meshes (shrunken and expanded) are very similar to each other, but have slightly different object boundaries. Now we use the two meshes and the depth map augmentations to check which parts of the scene should be visible in all three cameras, but are missing in the query depth map. If the query depth map does not contain any measurement at a pixel location, but the two meshes and the augmented depth map agree that a measurement should have been possible, the corresponding pixel is used as negative training samples.

Final Labels. The final label images are then a combination of the labeled pixels from the voting stage together with the negative samples from this stage. Note that for MVS, we also save the triangulation angle as meta information for each pixel. For positive samples, we use the measurement and the average pair-wise triangulation

angle of all cameras that were used for obtaining the measurement. For negative samples, we use the corresponding 3D point from the depth map augmentation (if available), otherwise we estimate the depth value with the shrunken mesh.

3.4 Learning and Predicting MVS Confidence

For both our applications (Automated Image Acquisition and MVS Prioritization), we want to know which camera constellation will give us a high likelihood of getting a complete and accurate 3D reconstruction. As this likelihood depends on the scene structure and the quality of the images, we want to use the acquired images together with a coarse scene representation. For training, we pose the problem as a pixel-wise classification task. During run-time, we compute the MVS Confidence depending on the triangulation angle and the scene around the pixel of interest.

For this task, we chose Semantic Texton Forests (STFs) [122]. We selected this approach for three main reasons. First, this approach is very fast in the execution phase as it operates directly on the input image (without any feature extractions or filtering). Second, STFs have shown a reasonable performance in semantic image segmentation. Third, it is possible to store meta information in the leaves of the forest. We use this property to store the triangulation angle under which a sample was obtain (or failed to obtain). This does not influence the learning procedure, but allows us to predict the reconstruction confidence in dependence of the triangulation angle at evaluation time.

Training. We implemented the STFs in the random forest framework of [118]. We only use STFs in its basic form without image-level prior [122] as this would require semantic categories. This means that all split decisions are made directly on the image data (Lab color space) within a patch of the size $d \times d$ without any explicit feature extraction (which makes the evaluation extremely fast). This is possible as the set of split decisions from the root node to leaf can be interpreted as something similar to a BRIEF descriptor [14].

In STFs, there are four types of split functions, of which one is randomly chosen in training for each node in the tree, and then compared to a randomly chosen threshold. The first type is a single color value v_{p_1,c_1}, where p_1 are pixel coordinates within the patch and c_1 is one of the three image channels. The second type is

the sum of two values $v_{p_1,c_1} + v_{p_2,c_2}$, the third type is the difference of two values $v_{p_1,c_1} - v_{p_2,c_2}$ and the last type is the absolute difference of two values $|v_{p_1,c_1} - v_{p_2,c_2}|$.

In all experiments, we used the following set of parameters. The parameters were chosen such that we obtain a good prediction performance, within a reasonable time frame as one of our applications demands real-time performance. For the patch size we used $d = 27$. Note that this is larger than in the original paper (which uses $d = 21$), however, in a parameter sweep this resulted in slightly better results. For the tree structure itself, we chose 20 trees with a maximum depth of 20. Note that the original paper only uses 5 trees with a depth 10, however, in our experiments this led to significantly worse results. Our choice is also supported by other works (e.g. [52, 119]), where 25 trees with a depth of 20-25 led to a good trade-off between performance and speed. We also experimented with 25 trees and a depth of 25, but could not observe a significant performance difference compared to 20 trees with 20 depth, while the run-time increased significantly.

For the training strategy, we set the parameters so that the training takes roughly a day on a Intel(R) Xeon(R) E5-2680 CPU. The resulting parameters are a minimum leaf size for further splitting of 50, 5000 node tests, 100 thresholds and 1000 random training samples at each node with approximately 4 million training patches for each class in training. Note that we cannot guarantee that this set is the optimal set of parameters, but in our experiments it worked sufficiently well.

Prediction. For both tasks (i.e. Automated Image Acquisition and MVS Prioritization), we want to keep the computational overhead as small as possible. Although our STF implementation is already quite fast on the CPU (approx. 600k predictions per second), it would still take several seconds for the evaluation of a 16Mpix image.

Thus we provide two ways to reduce the prediction time. First, we restructure the STF leaf nodes to contain a fixed number (b) of angular bins with one confidence value for each bin. Second, we can make use of the property that the confidence is in general a smooth function for a specific type of object (Section 6.2.2) (note that this fact is also exploited by Poggi and Mattoccia [103] and Kim et al. [68]). Thus, we evaluate the MVS Confidence on a regular grid and compute a confidence image with b channels for each input image.

With this approach it is now possible to obtain a quasi-dense prediction for the whole image within a definable fixed time frame. This means that we can

precompute the reconstruction confidence for each image and can simply look up the precomputed predictions in the image space for each potential constellation.

To obtain a MVS Confidence Prediction from the random forest for a specific image constellation, all that is needed is a target triangle (from the coarse mesh) and the oriented images. With this information, the MVS Confidence can simply be looked up by projecting the triangle into the images.

Chapter 4

Automated Image Acquisition for Multi-View Stereo

The main aim of our image acquisition method [91] is to automate the close-range acquisition of images for high-resolution monocular 3D reconstruction such that the acquisition time is minimized, while the completeness of the reconstruction should be as high as possible (also respecting the desired ground sampling distance and 3D accuracy). Compared to traditional aerial image acquisition, which mostly covers large areas from a safe distance, the image acquisition at a close range brings many new challenges. These challenges include: (1) Regular flight patterns, such as grids or domes, are not well suited for close-range image acquisition in complex environments due to a lack of flexibility and large safety distances, which strongly limits the maximal obtainable ground sampling distance and completeness. In order to reconstruct highly complex scenes at close-range, the system should be able to adapt to the environment and adaptively avoid objects. (2) In practice, no sufficiently accurate 3D model can be assumed to be available beforehand. This consequently leads to the requirement of on-site 3D reconstruction and planning strategy. (3) For irregular image acquisition, the camera constellation strongly influences the 3D accuracy and the completeness. This behavior cannot fully be modeled mathematically due to the heuristic nature of the matching step in monocular 3D reconstruction. As this behavior also depends on the applied algorithm and the present type of 3D structure, this motivated us to apply machine learning to automatically learn this relation (via the MVS Confidence described in the previous chapter). (4) As the

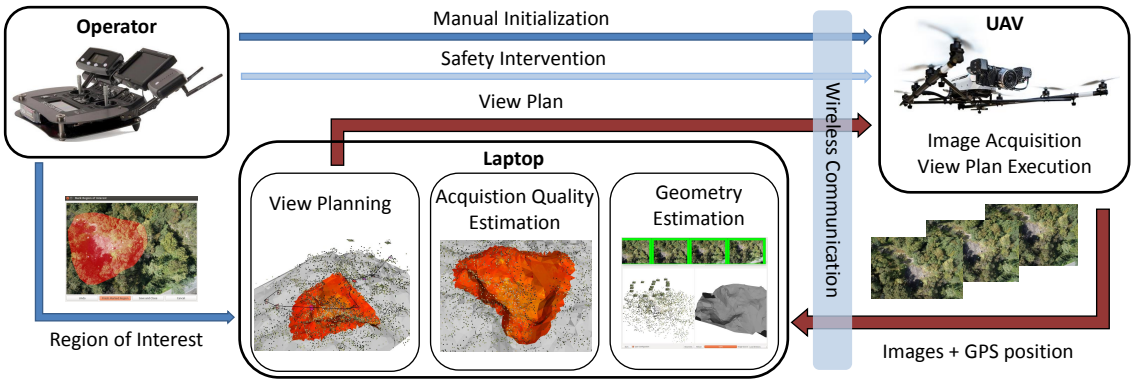


Figure 4.1: Automated Image Acquisition. After a manual initialization, our system loops between view planning and autonomous execution. This process is repeated as long as the reconstruction can be safely improved. Within the view planning procedure, we use the MVS Confidence to predict the best camera constellation for the presented scene and a specific MVS algorithm. This MVS algorithm will use the recorded high resolution images to produce a highly accurate and complete 3D reconstruction off-site in the lab.

view planning has to be done on-site, the planning time should be as low as possible.

In the following, we outline our approach [91] to handle these challenges and introduce the basic execution flow of the proposed system from a high level perspective (Section 4.1) and then explain the key parts of our planning strategy in more detail in Section 4.2.

4.1 Image Acquisition System

Our image acquisition system is requires three major components (see Figure 4.1). For acquiring images, we require an active robotic platform, which accepts 6D poses (aka way points) as input (in our experiments we used an Unmanned Aerial Vehicle (UAV)). For the geometry estimation and the view planning, we require a computational unit with a connection to the robotic platform. In our experiments, we used a laptop on the ground with a wireless connection to the UAV for this purpose. Finally, we require a human operator to define the goal of the acquisition and initialize the system manually. In the following, we describe the image acquisition system in more detail.

Initialization. To initialize our system, the operator first steers the UAV manually and acquires a few images of the scene of interest. These images, together with the

GPS position of the UAV, are directly streamed to a laptop which performs the geometry estimation in real-time. We use the GPS information to scale and georeference the reconstruction. After acquiring some images, the operator tells the system which parts of the scene are of interest by highlighting those parts in one or more of the acquired images. In the further steps, the system will focus on the marked region of interest for the quality estimation and disregard the other parts.

Geometry Estimation. For estimating the scene geometry, our system performs incremental structure-from-motion (SfM) [57] and incremental mesh updates based on a Delaunay triangulation of the SfM point cloud [56]. Both modules run concurrently in real-time for our scenario. To speed up the feature matching, the system uses a vocabulary tree [61] of SIFT features [76]. The matched features are used for camera pose estimation and the triangulation of 3D points. After a fixed number of iterations of local bundle adjustment, the sparse 3D points are incrementally integrated by the meshing procedure, which outputs a closed surface mesh. When the operator invokes the planning procedure, we run one last iteration of bundle adjustment and then make all points available to the meshing procedure.

View Planning. The first step of our view planning strategy is the assessment of the current acquisition quality. In other words this means that, before we start to plan, we need to identify which parts of the scene still need improvement. For estimating the acquisition quality, we use a snapshot of the mesh and the camera poses from the geometry estimation. With this information the system estimates how well the scene of interest is covered by the already captured images. For the quality estimation we use four different factors. The first three factors encode the fulfillment of geometric requirements defined by the operator, i.e. coverage, ground resolution and 3D accuracy. The fourth factor is the MVS Confidence Prediction output (Chapter 3), which encodes the likelihood that the MVS algorithm will work as intended.

After evaluating the acquisition quality, we now plan a fixed number of camera positions (view plan) that improve the acquisition quality. As we perform all our computations on-site during the acquisition, we only have a limited time frame available for the plan creation. Moreover, it is very likely that the geometry estimate is not complete and many parts of the scene are not yet sufficiently represented in

the mesh due to occlusions. Thus, we aim to obtain a good view plan in a limited time frame rather than one close to optimal solution. The final quality of the MVS reconstruction depends on a combination of many factors. These factors include the scene geometry, texture, lighting, the camera constellation and the chosen MVS algorithm. In this work, we respect these factors in predicting the MVS Confidence based on the rough scene estimate and 2D images. The exact algorithm is detailed in Section 4.2.

Closing the Loop. The resulting view plan is translated into GPS positions and camera angles and is sent to the UAV, which then executes the plan autonomously. When an image is taken, it is again streamed down to the laptop, where it is integrated into the reconstruction. After the execution, the system reassesses the acquisition quality and launches a new iteration of view planning and execution if our requirements are not yet fulfilled.

4.2 View Planning

The aim of our view planning approach [91] is to plan a set of useful camera poses in a fixed time frame. As the view planning problem is NP-hard, we have to make several simplifications to constrain the computation time.

One of our most prominent simplifications is that we plan camera triplets instead of single cameras. On the one hand, this lets us directly integrate our MVS Confidence Prediction and, on the other hand, we can treat each camera triplet as an independent measurement unit. In Figure 4.2 we show an overview of our approach, which we use to guide the reader through our algorithm and its submodules. As input our approach requires a snapshot of the estimated geometry (mesh and camera poses), as well as the pre-computed MVS Confidence images. Further, the operator has to label a region of interest in one of the images, and define the desired quality constraints (ground resolution and 3D accuracy). Optionally, the operator can also mark his observation position in one of the images. This position will be used to ensure that there is a direct line of sight between the planned UAV trajectory and the operator, such that he can manually intercept at any stage.

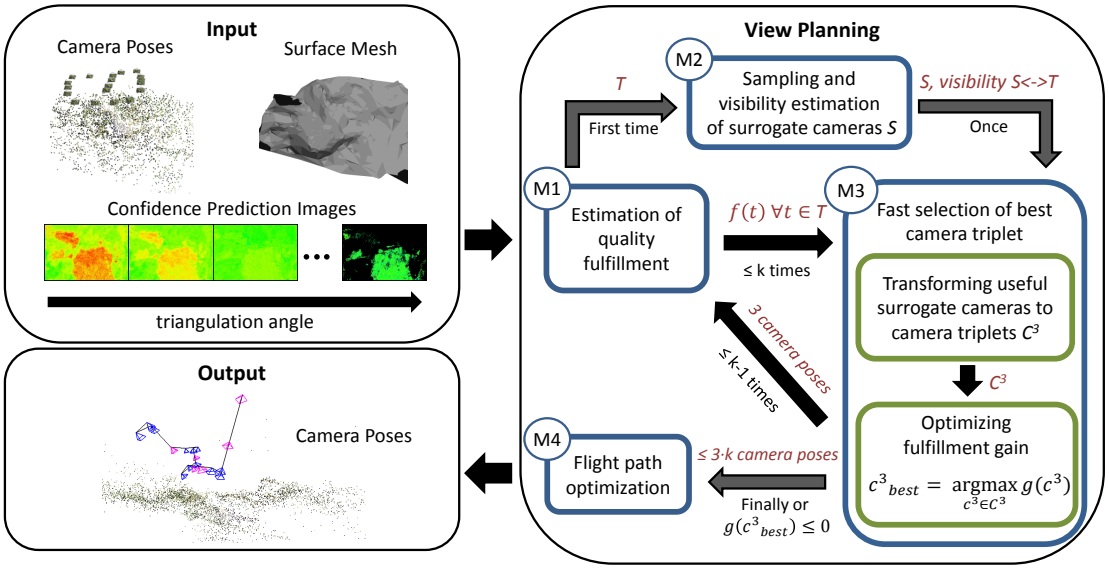


Figure 4.2: View planning. Our algorithm tries to find the k next best camera triplets for improving the acquisition quality. It uses pre-computed MVS Confidence Prediction images to constrain the triangulation angle between the cameras with the learned limitations of the MVS algorithm. Next to the arrows, we show the data communication between our submodules (M1-M4) in red and in black we show how often this data is computed. S is the set of surrogate cameras, T the set of considered unfulfilled triangles and C^3 the set of camera triplets generated from the surrogate cameras.

Estimation of quality fulfillment (M1). The whole view planning algorithm starts with the estimation of the current quality fulfillment, i.e. which parts of the scene have been sufficiently covered and which parts could be improved with further images. Therefore, we assess how the desired quality constraints are currently fulfilled by the already captured images. For this estimation, we need the already acquired images and their camera poses C as well as the surface mesh. Then we split the triangles of the surface mesh within the region of interest to approximately the same size through iteratively splitting them until the maximum edge length equals the average edge length before splitting. Within the region of interest, we then randomly select a fixed number N_t of triangles. Next, we determine the visibility information between these triangles and the camera poses C through rendering the mesh from the camera poses. After the rendering, we also know which cameras see which triangles, i.e. each camera that sees the triangle t is in the set C_t. As C_t can contain many camera combinations that will not work, we can speed up the compu-

tation significantly in pre-grouping the cameras in feasible clusters using mean shift. This clustering is done first via 1D mean shift on the logarithm of the resolution (i.e. $log(r_{c,t})/log(r_{\min})$ with $c \in C_t$) and a unit bandwidth. We use the logarithm here as the relative scale changes (e.g. 0.5 and 2) have the same distance in logarithmic space. The resulting clusters are then further refined by performing mean shift clustering of the viewing direction with a bandwidth $\alpha_{\min}$. This splits the camera set C_t into subsets $C_{t,i}$ (i.e. $C_t = \bigcup_i C_{t,i}$ and $\bigcap_i C_{t,i} = \emptyset$). For each triangle, we now generate all possible combinations of camera triplets from the cameras that observe a triangle t respecting the camera clusters (i.e. $c^3 \in C^3_{t,i} \forall i$). For each camera triplet c^3, we now evaluate the following four fulfillment functions.

(1) The *coverage* is modeled as a Boolean, which is true if sufficient cameras ($\geq x$) observe the triangle, i.e.

$$f_{cov} = \begin{cases} 1, & \text{if } \exists C_{t,i} \in C_t \text{ with } |C_{t,i}| \geq x \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

(2) The *resolution* requirement (px/m^2) is defined as a truncated ratio:

$$f_{res} = \min\left\{\frac{r}{r_d}, 1\right\}, \tag{4.2}$$

where r_d is the desired resolution and r is the estimated resolution. We determine r by projecting the 3D triangle into each camera of c^3 and averaging the pixel area. The desired resolution r_d can be computed from a desired ground sampling distance g_d as $r_d = 1/g_d^2$.

(3) The fulfillment of the *3D uncertainty* requirement for a desired accuracy a_d is defined as:

$$f_{unc} = \min\left\{\frac{a_d}{\sqrt{u}}, 1\right\}, \tag{4.3}$$

Here, u stands for the maximum eigenvalue of the covariance matrix Cov_{3D} related to a triangle's centroid. For computing the covariance matrix, we use the same uncertainty model as for the training data generation (see Equation 2.6).

(4) The last fulfillment function is the output of our MVS Confidence Prediction algorithm f_{conf} (Sec. 3.4). Note that the prediction tries to capture everything that is not covered by the 3D uncertainty model. In other words, f_{conf} represents

the likelihood that we will obtain a 3D measurement which abides our uncertainty model.

With these four functions, we then evaluate the combined fulfillment function as:

$$f(t, c^3) = (\alpha f_{res} + (1 - \alpha) f_{unc}) \cdot f_{cov} \cdot f_{conf} \tag{4.4}$$

This formulation allows the operator to define the relative weight α between desired ground resolution and 3D accuracy, while the coverage and MVS Confidence encode the chances of a successful reconstruction. The overall fulfillment of a triangle t is computed as

$$f(t) = \max_{c^3 \in C_t^3} f(t, c^3). \tag{4.5}$$

Based on the fulfillment information, we now further reduce the number of considered triangles to a triangle set T. We guide this reduction such that we end up with triangles that have a low fulfillment but are well distributed over the scene of interest. Thus, we randomly select a fixed number N_v of triangles from a piece-wise constant distribution, where the chance of selecting a triangle t is weighted with $w(t) = 1 - f(t)/f_{conf}(t)$. We remove f_{conf} from the weight to avoid bias towards structures that might not be reconstructible at all.

The output of this module (M1) is a set of considered triangles T (with $|T| \leq N_v$) and the current fulfillment $f(t)$ for all triangle $t \in T$.

Sampling and Visibility Estimation (M2). As sampling camera poses with orientation and then estimating the visibility of the geometry for each view contains many redundant computations, we approach this topic in an more efficient way by inverse geometry rendering. The key idea of our approach is that we place virtual cameras on the surface of the geometry and render potential camera centers as 3D points into these virtual cameras. The big advantage of this approach is that we can evaluate hundred thousands of potential camera poses in the same time it would take to evaluate hundreds camera poses in the direct approach.

Thus, we first randomly sample a fixed number N_p of 3D positions in the free space of the scene. For the representation of free space, we use a probabilistic octree representation [73]. As we only use this octree to keep a safe distance to the environment, we can use a very coarse maximum resolution. This makes the memory and computational effort of maintaining this data structure negligible. If

the operator marked his observation position in one of the images, we also check which position has a direct line of sight to the operator and only keep the ones which are visible from the operator's position.

Each sampled 3D position in free space represents the camera center of a surrogate camera. A surrogate camera has an unlimited field of view and thus also no orientation at this point (later we will transform this surrogate camera into a camera triplet). The usage of surrogate cameras allows us to reformulate the visibility estimation problem and to estimate which surrogate cameras are visible from a given triangle instead of the other way around. Thus, we generate a virtual camera for each triangle $t \in T$. The camera center of a virtual camera is set to the triangle's centroid and the optical axis to the triangle's normal. We set the focal length of this camera such that we get a fixed field of view ϕ. Now we use the virtual cameras for rendering the scene, i.e. the mesh and the 3D points that define the centers of the surrogate cameras. The resulting visibility links are stored in the surrogate cameras.

Finding the best camera triplet (M3). To find the best camera triplet at a low computational cost, we guide the transformation from surrogate cameras to camera triplets such that we only need to evaluate potentially useful and feasible camera constellations. Thus, we first compute the potential fulfillment gain $g_{pot}(t, s)$ of a surrogate camera s with respect to a linked triangle t as:

$$g_{pot}(t, s) = max_{\alpha}\{f(t, c_{\alpha}^{3}) - f(t), 0\}, \tag{4.6}$$

for a hypothetical equilateral camera triplet c_{α}^{3}, that has the surrogate camera in its center and where each camera directly faces towards the triangle. With *equilateral camera triplet*, we mean that that baseline/distance between each camera pair in the triplet is the same, which means that the three cameras together span an equilateral triangle perpendicular to the main view direction. The triangulation angle α defines the distance between the cameras in the b steps of the predicted MVS Confidence, which we evaluate with the confidence image of the closest already captured image (with respect to the surrogate camera) that observes the triangle.

Using this potential gain information, we determine in which direction the surrogate cameras should face. Therefore, we perform a weighted mean shift clustering on the rays towards the linked triangles. As a weight we use the fulfillment gain and

the bandwidth is set to the minimum camera opening angle (the minimum camera opening angle is defined by the smallest image dimension (height or width) and the principal point). The winning cluster (i.e. the cluster with the highest potential fulfillment gain) is chosen to define the general viewing direction of the surrogate camera. Then we update the visibility information of the now oriented surrogate cameras.

Given the orientation, we generate b equilateral camera triplets for each surrogate camera, one for each confidence bin. For each camera triplet c^3 we efficiently check the distance to obstacles [73] and compute the fulfillment gain of c^3 as

$$g(c^3) = \sum_{t \in T_{c^3}} max\{f(t, c^3) - f(t), 0\}, \tag{4.7}$$

where T_{c^3} are the triangles that are visible from c^3. Over all triplets, we find the best camera triplet as

$$c^3_{\text{best}} = \arg\max_{c^3 \in C^3} g(c^3), \tag{4.8}$$

where C^3 is the set of all generated camera triplets. For the implementation, we can drastically reduce the number of evaluations by using the potential gain as an upper bound.

If $g(c^3_{\text{best}})$ is greater than zero and we have not yet planned k camera triplets, we add c^3_{best} to the set of already acquired images (C) and plan a new camera triplet. Otherwise, we pass all planned camera triplets with positive gain on to the flight path optimization.

Flight path optimization (M4). This module minimizes the travel distance between the camera poses and ensures that the resulting images can be registered by the geometry estimation module. First, we reorder the camera poses with a greedy distance minimization using the last captured image as a starting point. Then we check if the taken images can be connected to the given set of images respecting the capture sequence. We assume that this is the case if an image has a minimum overlap o_{min} with at least one of the previously captured images. If this is not the case we sample camera poses which fulfill this property along the trajectory from the closest previously captured camera pose to the target camera pose. This results in a view plan that ensures a successful sequential registration of the planned image

set. The complete view planning strategy is summarized in Algorithm 3.

Data: A set C of already acquired images with poses and precomputed confidence images and a surface mesh with a set of triangles T_{all}, where a subset of the triangles $T_{ROI} \subseteq T_{all}$ were marked to be inside the region of interest.
Result: The view/path plan as an ordered list of camera poses $L_c = (c_1, c_2, ..., c_n)$.
Estimation of quality fulfillment (M1)
sample random triangle subset T_r from T_{ROI} with $|T_r| = N_t$;
assess fulfillment $f(t)$ with respect to C and compute the weight $w(t)$ $\forall t \in T_r$;
reduce T_r to T with $|T| = N_v$ by weighted random selection with $w(t)$;
Sampling and Visibility Estimation (M2)
sample a set of surrogate cameras S (as 3D points) in free space with $|S| = N_p$;
render T_{all} and S from virtual cameras placed on T and store links ($S \leftrightarrow T$);
$C_p \leftarrow$ empty ;
Finding the best camera triplet (M3)
while $\Big($*first iteration* **or** $\big($ $|C_p| < k$ **and** $c^3_{best} \neq$ *invalid* **and** $g(c^3_{best}) > 0\big)\Big)$ **do**
$c^3_{best} \leftarrow$ invalid; $g(c^3_{best}) \leftarrow 0$;
compute potential gain $g_{pot}(t, s)$ $\forall s \in S$ and $t \in T$;
compute and store $G_{pot}(s) = \sum_{t \in T} g_{pot}(t, s)$ $\forall s \in S$;
for all $s \in S$ **do**
if $G_{pot}(s) > g(c^3_{best})$ **then**
fix view direction of s with weighted mean shift based on $g_{pot}(t, s)$;
scoped update of visibility links ($s \leftrightarrow T$) with view frustum;
for all *triangulation bins* b **do**
generate equilateral camera triplet c^3 based on b;
compute fulfillment gain $g(c^3)$;
if $g(c^3) > g(c^3_{best})$ **then**
$g(c^3_{best}) \leftarrow g(c^3)$; $c^3_{best} \leftarrow c^3$;
end
end
end
end
if $g(c^3_{best}) > 0$ **then**
add all cameras of c^3_{best} to camera sets C_p;
update fulfillment $f(t) \forall t \in T$ analog to (M1) with the camera set $C \cup C_p$;
end
end
Flight path optimization (M4)
order all cams in C_p with greedy path planning to ordered list L_{tmp};
for $i = 1...|L_{tmp}|$ **do**
if *camera* c_i *has insufficient overlap with all* $c \in C$ **then**
sample cameras L_{c_i} from closest camera $c_{c_i} \in C$ to c_i to ensure overlap;
append L_{c_i} to L_c and insert all $c \in L_{c_i}$ in C;
end
append c_i to L_c and insert c_i in C;
end

Algorithm 3: View Planning Strategy

Chapter 5

Multi-View Stereo Prioritization

In this chapter, we aim to improve the efficiency of multi-view stereo (MVS) approaches based on depth maps. This type of approach is very popular (e.g. [35, 45, 108, 115, 142]) as it is inherently parallelizable and delivers state-of-the-art results. One drawback of such approaches is that they typically generate one depth map per image in the dataset. For modern cameras, this means that 3D points in the order of 10^7 are created per image. Within a few hours it is possible to acquire images that will result in billions of 3D points. While it is true that more data can only improve the completeness, practice shows that the acquired data is very often highly redundant. In many cases, it is possible to retain all important information with a small fraction of the acquired data.

The basic element of our prioritization [88] is a view cluster, which stands for a key view (i.e. an image used for generating a depth map) together with its matching partners. The aim of our MVS Prioritization is to establish an ordering of view clusters such that any subset of the ranked entries from entry 1 to n retains a maximum of information about the scene for any n. In this work, we approach the problem of view cluster prioritization in two steps.

In the first step, we aim to select suitable matching partners for each image in the image set; i.e. each image is considered as a potential key view in this step. To generate a depth map, we require other images that observe the same part of the scene from different positions. Choosing a subset of images for this task has a lot of impact on the quality of the resulting depth map. On the one hand, choosing images with a large baseline to the key view will lead to depth values with high accuracy. On the other hand, it will also make the matching task

(i.e. finding correspondences between the images) significantly harder, which will negatively influence the completeness of the depth map. Selecting more images will be beneficial for completeness and accuracy, however, the matching time also rises linear with the number of matching partners. So for increasing the efficiency, the number of matching partners is typically kept as low as possible. Thus, in this subproblem it is our aim to find the best subset of k matching partners such that k can be very small and we still obtain high quality results in terms of accuracy and completeness.

The second step is next-best view ranking. For this step, let us assume that we have already decided the set of matching partners for each potential key view. While it is possible to generate one depth map for each view, this approach leads to a massive amount of highly redundant data. Thus it is our aim to rank the view clusters (key views with matching partners) such that the most useful view clusters can be processed first. As by product, our approach delivers a fulfillment prediction (with respect to a desired ground resolution and 3D accuracy) for each added view cluster. This information makes it possible to determine how many and which view clusters are necessary to obtain a certain level of fulfillment prior to executing the MVS algorithm itself.

In Figure 5.1, we show an overview of the complete ranking procedure, which is explained in full detail in the remainder of this chapter.

5.1 Preprocessing

As input our approach requires the color images, a pre-trained MVS Confidence Predictor (Chapter 3) and the structure-from-motion output including the corresponding camera poses and a sparse 3D point cloud. For our metric quality constraints (ground sampling distance and 3D accuracy), we also require that the structure-from-motion output was transformed metrically. In our experiments, we used automatically detectable ground control points (GCPs) [110] for this purpose.

From this input, we then robustly extract a surface mesh with a Delaunay triangulation of the sparse point cloud [71] with the same parameters used in [89]. The amount of triangles of the resulting surface mesh is in the order of the number of sparse 3D points. For view cluster prioritization, the resulting geometry complexity is typically already unnecessarily high. Therefore, we first simplify the mesh

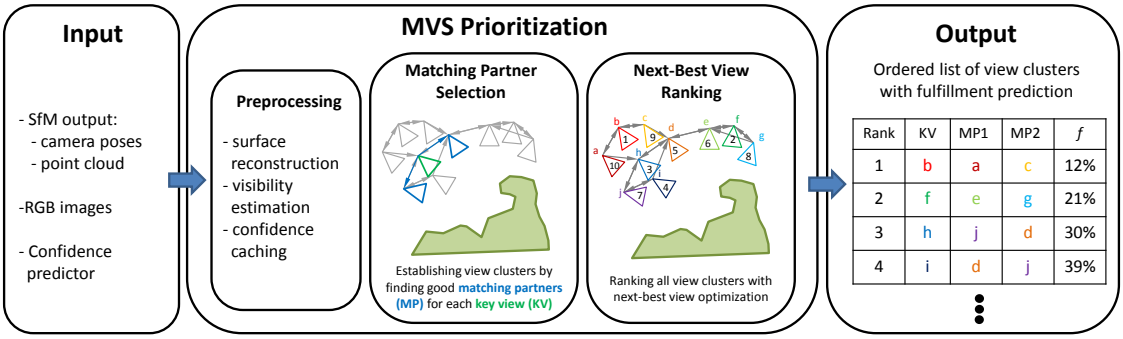


Figure 5.1: MVS Prioritization. As input our approach requires the SfM output (i.e. the camera poses and the sparse point cloud), the original color images and the pre-trained confidence predictor. From the SfM output, we extract a surface mesh, which is then used for visibility estimation (i.e. estimation of which camera sees which mesh triangles). With this information, we then precompute the MVS Confidence for each image as described in Chapter 3 and cache the predictions for each visible mesh triangle. Using this data, we then find good matching partners for each view, which results in a set of view clusters. These view clusters are then ranked by their importance for the overall reconstruction in a next-best view scheme. The output is ordered list of view clusters (i.e. key views with matching partners) together with a fulfillment prediction f for each entry.

reconstruction using an adaption of quadratic edge collapse decimation [40], which terminates when 95% of all triangle edges are above r times the desired ground sampling distance. In our experiments, we found $r = 20$ to be good value, as with this value all important 3D structures are still contained in the mesh, but the number of triangles is drastically reduced. To balance the triangle size independent of the 3D topology, we then refine large mesh triangles by iterative sub-division until all edge lengths are below e times the desired ground sampling distance. We found that $e = 5 \times r$ lead to a good trade-off between simplicity and balance. Note that the surface mesh is computed exactly once at the beginning of our approach and is not refined with the MVS output as our prioritization approach works before the MVS algorithm is executed.

5.2 Matching Partner Selection

For computing a depth map with MVS, each key view requires a set of matching partners which observe the same scene from different view points. In this step of our approach, we try to find a good set of k matching partners in the sense that

completeness and accuracy are optimized simultaneously. We say that we are aiming for a *good* set (opposed to the optimal set) as even the solution space for a single key view can be extremely large. The size of the solution space for this task is the same as for unordered sampling without replacement, i.e. the binomial coefficient $\frac{n!}{k!(n-k)!}$, where n is the size of the potential set of matching partners. Even for quite small numbers, like picking $k = 6$ matching partners out of $n = 23$ possibilities, the solution space is already larger than 100k.

For this purpose, we follow other works in the field (e.g. [6, 45]) and use the connectivity between the images for a dimensionality reduction. Thus, we first reduce the set of potential matching partners to the top n most connected images. We evaluated the connectivity based on the number of sparse points shared with the key view. From this set, we then draw y combinations. As the connectivity can be a very strong cue, we also want to ensure that the most connected images are included in the drawn y combinations. Thus, we first draw all possible combinations of the q most connected images, where q is the largest set size for which the solution space is less or equal $\frac{y}{4}$ (i.e. $\frac{q!}{k!(q-k)!} \leq \frac{y}{4}$). The rest of the $y - q$ combinations are drawn randomly from the larger set of the n most connected images. In this way, at least 75% of the combinations are drawn completely random, whereas we also consider the most connected combinations. For each drawn combination of matching partners (c^k), we now evaluate four fulfillment functions in relation to the key camera c_{key} and a triangle t observed from c_{key}. Note that these fulfillment functions are very similar to the fulfillment functions for image acquisition, as the tasks show many similarities. The main difference is that now the fulfillment functions are generalized to k matching partners, while the image acquisition approach could only handle image triplets.

This means that the first three fulfillment functions stay the same as in Section 4.2. I.e. (1) the *coverage* is modeled as a Boolean function f_{cov}, which is true if sufficient cameras ($\geq x$) observe the triangle t and false otherwise. (2) The *resolution* requirement f_{res} is defined as a truncated ratio with respect to a desired resolution r_d as in Equation 4.2 and (3) the *3D uncertainty* requirement f_{unc} a truncated ratio with respect to a desired accuracy a_d as in Equation 4.3.

Only the last fulfillment function (4), the MVS Confidence f_{conf}, is changed significantly. In previous chapters, we presented a way to learn and predict this confidence without any ground truth for two matching partners. Here, we present a

way to generalize the confidence prediction step to an arbitrary number of k matching partners. For this purpose, we assume that the MVS reconstruction process follows a stochastic process and can be modeled as a combination of unary MVS Confidence Predictions.

Under this assumption, we can use probability theory to reason about the overall probability of obtaining a successful measurement for a pixel of a key view given a set of matching partners. For a successful measurement, most MVS approaches require successful matches to at least two matching partners. Thus, we formulate the overall probability of obtaining a successful measurement, as the probability of getting successful matches to at least two matching partners. In Section 5.3, we derive this overall probability by growing a binary probability tree. In the following, we apply the general solution described in Section 5.3 to our specific problem.

In our problem, we are given a surface mesh with a set of triangles T and a set of images C. For a specific combination of a key view $c_{key} \in C$ with k matching partners $c^k \in C^k$, we want to predict the chances of successfully reconstructing the depth of the pixels in which a triangle $t \in T$ projects.

Thus, we now end up with the following equation for the overall probability of a successful match considering all matching partners:

$$f_{conf}(t, c_{key}, c^k) = \sum_{i=2}^{k} \left((-1)^i \cdot (i-1) \cdot \sum_{c^{(i)} \in C_k^{(i)}} \left(\prod_{j=1}^{i} f_{conf}\left(c_{key}, c_j^{(i)}\right) \right) \right), \quad (5.1)$$

where $C_k^{(i)}$ is the solution space for drawing subsets of i cameras from the available set of k matching partners, $c^{(i)}$ is one of these subsets and $c_j^{(i)}$ is one camera of this subset and $f_{conf}\left(c_{key}, c_j^{(i)}\right)$ is the pair-wise MVS Confidence for camera $c_j^{(i)}$ with the key camera c_{key}. We compute this pair-wise MVS Confidence by averaging the unary confidence values of the two cameras. We extract the unary confidence value for a triangle t by averaging the pixel-wise confidence predictions of all pixels that lie inside the projection of triangle t into the individual camera. The confidence predictor is trained and executed as described in Chapter 3. Note that for improved efficiency, we also precompute and store the unary confidence values for each camera-triangle pair in our preprocessing step.

Based on these four fulfillment functions, we define the fulfillment for a triangle

t as:

$$f(t, c_{key}, c^k) = (\alpha f_{res}(t, c_{key}) + (1-\alpha) f_{unc}(t, c_{key} \cup c^k)) \cdot f_{cov}(t) \cdot f_{conf}(t, c_{key}, c^k), \quad (5.2)$$

where α defines the relative weight between the resolution and uncertainty fulfillment.

With this basis function, we now compute a combined fulfillment score for each set of drawn matching partners as

$$f(c_{key}, c^k) = \sum_{t \in T_{c_{key}}} f(t, c_{key}, c^k), \quad (5.3)$$

where c_{key} is the key camera, c^k is the selected set of k matching partners, the triangle set $T_{c_{key}}$ is the set of triangles visible in c_{key} with $T_{c_{key}} \in T_z$. T_z is a fixed fraction z of all the available triangle set T_{all} (i.e. $|T_z| = |T_{all}|/z$).

Of all drawn combinations of matching partners, we now select the one that maximizes this combined fulfillment. The combination of key view and selected matching partners, we will further call *view cluster*. We execute this step for all available images.

5.3 Probabilistic MVS Confidence Extension

In this section, we extend the confidence prediction framework to an arbitrary number of matching partners. For this purpose, we use a traditional probability tree. In Figure 5.2 we show such a probability tree for 4 matching partners. In the following, we use such a general probability tree to derive a probabilistic formulation for obtaining a successful 3D measurement from k cameras.

Let us first formalize the probability tree for k cameras as a binary tree with a depth k. Every depth level d represent more or less what happens if we add a d^{th} camera to the previous set of $d-1$ cameras. Every vertex in the tree represents an event, which is successful with a probability $p(d)$, where d is the depth of the tree (also corresponding to the d^{th} matching partner). Let us call a path from the root vertex to a leaf vertex (i.e. a vertex without child vertices) simply "branch".

Now let us grow the binary tree. As we are only interested in having at least 2 successful matches, we can stop growing a branch when it has reached two positive

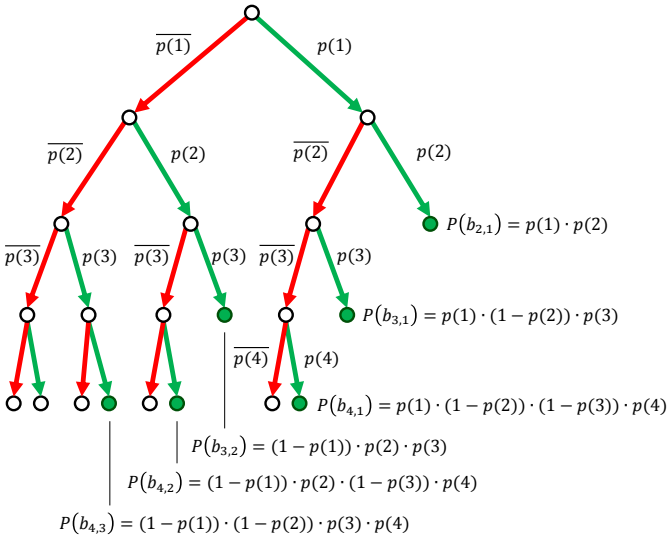


Figure 5.2: Probability tree. We formulate the MVS Confidence for a key view with k cameras as matching partners as a stochastic process (here $k = 4$). Each edge in the tree relates to an event. For a matching partner d, the event can either be positive (green color) with a probability $p(d)$ or negative (red color) with a probability $\overline{p(d)} = 1 - p(d)$. A positive event corresponds to a successful match between the matching partner d and the key view at a specific pixel location, which results in a 3D measurement that conforms with our uncertainty model. For obtaining a reliable 3D reconstruction, we require at least two matching partners with a successful match. Thus, each path in the tree with at least two positive events can be seen a successful branch. Each successful branch $b_{y,x}$ is defined by the first two positive events at level x and y respectively. The overall probability of a successful measurement can be obtained by summing all successful branches. Note that the order events in the tree is irrelevant for the overall probability, which can be better seen in Equation 5.7.

events (we will further call this kind of branch as "successful branch"). As such a successful branch has always exactly two successful events, we will further denote a successful branch as $b_{y,x}$, where y is the depth/id of the last successful event and x is the depth/id of the preceding successful event. This means that only branches with less than 2 successful events are grown in the next depth level. It follows that in the process of growing from level $d-1$ to d (i.e. adding a d^{th} camera), the tree gains exactly $d-1$ new successful branches (one for each camera already in the tree), while all old successful branches remain unchanged. It also follows that each new successful branch $(b_{d,i})$ has exactly one previous camera i (with $1 \geq i < d$)

with a successful event associated. Thus the probability along such a new successful branch $b_{d,i}$ is given by:

$$P(b_{d,i}) = p(d) \cdot p(i) \prod_{j \in \{1,2,\ldots,d-1\} \setminus i} (1 - p(j)), \tag{5.4}$$

where $p(x)$ is the probability of a successful match with camera x. If we now add up all branches with 2 successful events at level d, we end up with the following probability for obtaining at least 2 successful events:

$$P_{min2} = \sum_{i=2}^{d} \sum_{a=1}^{i-1} P(b_{i,a}) = \sum_{i=2}^{d} \sum_{a=1}^{i-1} p(i) \cdot p(a) \prod_{j \in \{1,2,\ldots,i-1\} \setminus a} (1 - p(j)) \tag{5.5}$$

Now we can expand the equation to

$$\begin{aligned} P_{min2} = \sum_{i=2}^{d} p(i) \cdot \Bigg(\left(\sum_{c=1}^{i-1} p(c) \right) - 2 \left(\sum_{c^{(2)} \in C_{i-1}^{(2)}} p\left(c_1^{(2)}\right) \cdot p\left(c_2^{(2)}\right) \right) \\ +3 \left(\sum_{c^{(3)} \in C_{i-1}^{(3)}} \prod_{a=1}^{3} p\left(c_a^{(3)}\right) \right) - \ldots + (-1)^i (i-1) \left(\sum_{c^{(i-1)} \in C_{i-1}^{(i-1)}} \prod_{a=1}^{i-1} p\left(c_a^{(i-1)}\right) \right) \Bigg), \end{aligned} \tag{5.6}$$

where $C_{i-1}^{(x)}$ is the solution space for drawing subsets of x cameras from the available set of $i-1$ cameras, $c^{(x)}$ is one of these subsets and $c_a^{(x)}$ is one camera of this subset. Now we can contract the equation again to

$$P_{min2} = \sum_{i=2}^{d} \left((-1)^i \cdot (i-1) \cdot \sum_{c^{(i)} \in C_d^{(i)}} \left(\prod_{a=1}^{i} p\left(c_a^{(i)}\right) \right) \right) \tag{5.7}$$

Note that this equation and Equation 5.1 are equivalent and that only the parameterization was changed; i.e. $d \mapsto k$, $P_{min2} \mapsto f_{conf}(t, c_{key}, c^k)$ and $p\left(c_a^{(i)}\right) \mapsto f_{conf}\left(c_{key}, c_a^{(i)}\right)$.

5.4 Next-Best View Ranking

In general, image acquisition for photogrammetric reconstruction is done with a lot of redundancy to ensure that the whole scene is sufficiently covered. However, a drawback of this acquisition technique is that some images do not contain any additional information compared to their neighbors. Finding the best and minimal subset for a sufficiently complete reconstruction is a very hard task. In fact, subproblems of this task are known to be NP-hard. E.g. if an oracle told us that the minimal number of key views for sufficiently covering the scene is n, we would still have to solve the NP-hard maximum coverage problem [27]. While solving this problem optimally is computationally intractable (unless P = NP), there exist approximation approaches with theoretic quality guarantees for the found solution if the function is a nonnegative monotone submodular function. Thus, let us define such a function with the following three conditions [69]:

Definition 5.4.1. (Submodularity) A set function $f : 2^V \to \mathbb{R}$ is *submodular* if for every subset $A, B \subseteq V$ (where V is a finite set) it holds that,

$$f(A \cap B) + f(A \cup B) \leq f(A) + f(B). \tag{5.8}$$

Definition 5.4.2. (Monotonicity) A set function $f : 2^V \to \mathbb{R}$ is *monotone* if for every subset $A \subseteq B \subseteq V$ (where V is a finite set) it holds that,

$$f(A) \leq f(B). \tag{5.9}$$

Definition 5.4.3. (Nonnegativity) A set function $f : 2^V \to \mathbb{R}$ is *nonnegative* if for every subset $A \subseteq V$ (where V is a finite set) it holds that,

$$f(A) \geq 0 \tag{5.10}$$

If an objective function f fulfills these three conditions, then a greedy algorithm is guaranteed to select n views such that the reached objective value is within 63% of the optimal objective value for the same amount of views [94] (see [69] for a good explanation of the proof and all implications). This fact (and the fact that n is not easily estimated) motivated us to solve this problem with a greedy algorithm and a nonnegative monotone submodular objective function.

Objective function. Now let us formalize the objective function which we aim to maximize, i.e. the overall fulfillment, as:

$$f_o(V,T) = \frac{1}{|T|} \sum_{t \in T} \max_{v \in V} f(t,v), \tag{5.11}$$

where T is the set of all mesh triangles, $|T|$ is the size of this set, t is one of these triangles, V is the set of view clusters and v is one of these view clusters. $f(t,v)$ is the triangle fulfillment as defined in Equation 5.2 with $v = \{c_{key}, c^k\}$.

Relation to other problems. If we consider that V and T are fixed, we see that this optimization problem is closely related to the well known facility location problem [69]. Here, the triangles t are the customers and the view clusters v are the facilities and each customer (triangle) selects the facility (view cluster) with the highest value (fulfillment). As $f(t,v) \geq 0$ for all t and v, it also follows that the objective function $f_o(V,T)$ is a nonnegative monotone submodular function [69]. This means that the quality guarantees of [94] for a greedy algorithm apply to our objective function. Note that we can also relate our problem to another NP-hard problem, i.e. the maximum coverage problem [27]. If we consider the special case that $f(t,v) = 1$ if t is visible from view cluster v and $f(t,v) = 0$ otherwise for all $v \in V$ and $t \in T$, our problem reduces to the maximum coverage problem (i.e. maximizing the number of all observed/covered triangles). This means our problem is a generalization of the maximum coverage problem, which means that our problem is still NP-hard despite having a nonnegative monotone submodular objective function.

Algorithm. We formulate our ranking procedure as a greedy algorithm. In each iteration, we select the next-best view cluster, i.e. the view cluster with the highest fulfillment gain. We define this gain for a view cluster v_i as:

$$g(v_i, V', T) = f_o(\{V' \cup v_i\}, T) - f_o(V', T), \tag{5.12}$$

where V' is the set of already selected view clusters. For an efficient solution to this problem, we propose the following algorithm.

Our algorithm first starts by estimating the visibility between each view/camera

and the triangle mesh. This visibility information is stored in each triangle (i.e. each triangle knows which camera has a direct line of sight to it). Based on this information, we precompute the fulfillments $f(t, v)$ for all triangles $t \in T$ with sufficient camera views ($\geq x$, related to the coverage fulfillment f_{cov}) and all view clusters. This fulfillment $f(t, v)$ is then stored within a map in each triangle t.

For each view cluster v, we now compute the initial fulfillment gain $g(v, V', T)$ with $V' = \{\}$ (i.e. an empty set of selected view clusters). Each pair of view cluster and gain value $\{v, g(v, V', T)\}$ is inserted into a priority queue, where now each view cluster is ranked by its fulfillment gain. While initializing this data structure has a complexity of $\mathcal{O}(n \log n)$ with $n = |V|$, we can now use lazy updates in each iteration and avoid a significant number of unnecessary computations.

In each iteration, we now select and pop the top element of the queue, i.e. the element with the highest fulfillment gain, and add the selected view cluster v_s to the set of selected view clusters V'. Then we update the current fulfillment for each triangle t that is observed by v_s. This current fulfillment is stored inside the triangle t and computed as $f'(t, V') = \max_{v \in V'} f(t, v)$. Based on this information, we now lazily update all elements of the priority queue. This means that we select and pop the top element of the queue and update the fulfillment gain of the corresponding view cluster v_u. We store all such temporarily removed view clusters V_u in a separate data structure and keep track of the maximum fulfillment gain g_{max} over all these values. We can stop the update procedure if $g_{max} \geq g(v_{top}, V', T)$, where v_{top} is the current top element of the priority queue. Then all temporarily removed view clusters V_u are reinserted into the priority queue. Now the next iteration can start. We terminate if $g(v_{top}, V', T) = 0$ or the priority queue is empty.

Note that all computational components (the visibility estimation, the confidence prediction and the fulfillment computation) are computed only once at the beginning of algorithm and the execution of the actual MVS algorithm is avoided altogether. This makes the ranking procedure very light weight in terms of computation time. The final output of this ranking procedure is an ordered list of view clusters (key views with matching partners) together with the estimated fulfillment up to this point.

Chapter 6

Experiments

We split our experiments into four parts. First, we validate our approach for automatic training data generation for two-view stereo. There are two main reasons for this choice. First of all, there exists more ground truth data for two-view stereo than for multi-view stereo. Second, the confidence measures in two-view stereo have a long standing tradition (see Section 2.3) and allow us to evaluate the quality of our training data in terms of confidence measuring performance.

In Section 6.2, we then apply our training scheme to MVS on the task of preserving prehistoric rock art in an environment dominated by vegetation. Within this scenario, we then analyze what the system learned about certain scene structures in relation to two very different MVS algorithms. In the same environment, we then also evaluate our approach for Automated Image Acquisition (Section 6.3) and MVS Prioritization (Section 6.4), and demonstrate that both applications strongly benefit from the MVS Confidence Prediction in terms of reconstruction quality and efficiency. In the latter experiment (Section 6.4), we also evaluate the generalization of the MVS Confidence Prediction across different domains to demonstrate the stability of our approach towards structures that were never seen in training.

6.1 Training Data Generation for Two-View Stereo Confidence Measures

In our experiments with two-view stereo confidence measures (published in [92]), we use three publicly available datasets, which are namely the KITTI2012 dataset [42], the Middlebury2014 dataset [112], and the Strecha fountain dataset [126].

The main focus of our experiments is on the KITTI2012 dataset [42], because it is well-suited to demonstrate our approach and has already been used for the evaluation of confidence measures before [46, 97]. The KITTI2012 dataset does not only let us evaluate the coverage and accuracy of our approach, but also lets us highlight the usefulness of our approach in boosting the performance of confidence measures by simply training them on the automatically generated training data.

6.1.1 General Setup

For all experiments, we used the same set of parameters. The parameter $\alpha_{\min}$ ($= 10°$) can be used to adjust the trade-off between coverage and label error. As a general rule, we can say that if one increases this parameter, the false positive rate becomes lower, but at the same time the label coverage decreases as well. The parameter $\sigma_{\max}$ ($= 2$) can be used to express desired accuracy of a query algorithm as a multiple of the σ bound.

As query algorithms, we use two different stereo algorithms. The first algorithm is a Semi-Global Matching (SGM) [54] implementation by Rothermel et al. [108] which uses the census transform for computing the matching cost. As a second algorithm, we chose the Slanted Plane Smoothing (SPS) approach of Yamaguchi et al. [139]. We chose this approach because it shows a very good performance on the KITTI datasets [42, 81], and gives a completely different output than a SGM (piece-wise planar super pixels vs. unrestricted transitions).

For analyzing the benefit of our approach for learning, we have chosen three different machine-learning based confidence measures [46, 97, 125]. All three approaches use random forests, which made it possible to re-implement them in a common framework. The difference between the approaches lies in which hand-crafted features they feed to the random forest. Ensemble learning [46] uses the peak ratio, entropy of disparities, perturbation, left-right disparity difference, hor-

trees	max depth	min leaf size	split criterion	#samples/node	#thresholds
20	20	100	standard entropy	2000	500

Table 6.1: Random Forest Parameters.

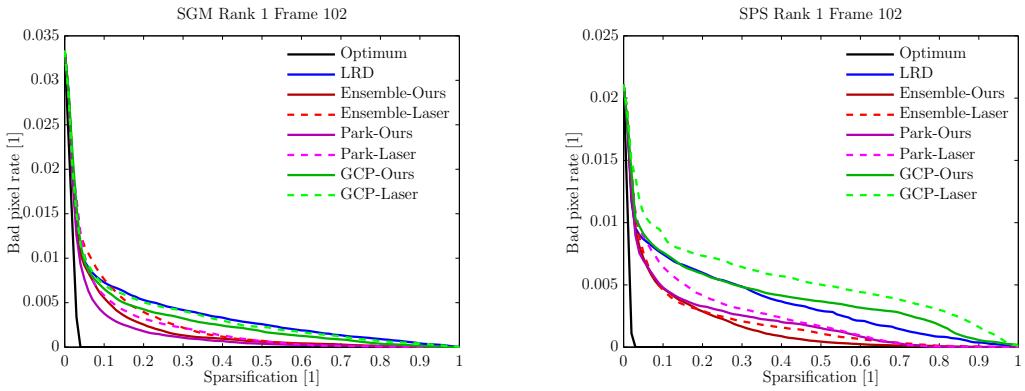
izontal gradient, disparity map variance, disparity ambiguity, zero mean sum of absolute differences and the local SGM energy, which results through consideration of multiple scales in a feature vector of 23 dimensions. Ground Control Point (GCP) learning [125] uses eight features, which are the matching cost, distance to border, maximum margin, attainable maximum likelihood, left-right consistency, left-right difference, distance to discontinuity and difference with median disparity. Park et al. [97] use a feature vector with 22 dimensions, which contains the peak ratio, naive peak ratio, matching score, maximum margin, winner margin, maximum likelihood, perturbation, negative entropy, left-right difference, local curvatures, local variance of disparity values, distance to discontinuity, median deviations of disparities, left-right consistency, magnitude of image gradient and the distance to border.

For the implementation, we used the publicly available random forest framework of Schulter et al. [118]. For training the forest, we used the same settings in all our experiments. We used 20 trees with a maximum depth of 20 and a minimum leaf size of 100. For choosing a split function we use the standard entropy and draw 2000 random samples per node and 500 random thresholds per feature channel. All parameters are summarized in Table 6.1. For every training setup, we balanced the dataset on image basis. This means that every image contributed as many positive training examples as negative examples. For the final evaluation, we always considered the complete image. For obtaining the pose estimation on the KITTI2012 dataset, we use the Library for Visual Odometry 2 (LIBVISO2) [43].

6.1.2 KITTI Dataset

We use the KITTI2012 dataset [42] to evaluate three properties of our ground truth generation, which are namely accuracy, coverage and training performance. The first two, we obtain by comparing our automatically generated label images to label images produced with the laser ground truth provided for the training dataset.

For the SGM [108] data, we reach an **accuracy of 97.3**% (STD: 1.4%) at an average coverage of the laser ground truth of 47.8% (STD: 11.8%). For the SPS [139]



(a) SGM Sequence 102.

(b) SPS Sequence 102.

Figure 6.1: Sparsification curves for sequence 102 of the **KITTI** training dataset. We display all combinations of query algorithm (SGM [108] and SPS [139]), confidence measure (Ensemble [46], GCP [125], Park [97]) and training data (Laser and Ours). As a baseline method we also show the Left-Right disparity Difference (LRD).

data, we obtain an **accuracy of 95.3%** (STD: 5.7%) at an average coverage of 48.6% (STD: 13.4%). Note that the coverage mostly depends on the camera motion. The ideal case to demonstrate our approach would be a circular motion around an object, whereas no motion will result in no labeled images. As the KITTI dataset contains some sequences with very little motion, this results in a high standard deviation of the coverage.

While accuracy and coverage are relevant, the much more interesting factor is how well the data is suited for training an algorithm. To analyze this factor, we benchmark the change of the confidence measuring performance of three recent confidence measures, which we further refer to as Ensemble [46], GCP [125] and Park [97]. For benchmarking this performance, we evaluate the Area Under the Sparsification Curve (AUSC) as in [46, 60, 97]. A sparsification curve plots the bad pixel rate over the sparsification factor. For drawing the curve, the pixels are sorted by confidence values and always the lowest values are removed. Sparsification curves for frame 102 of the dataset are shown in Figure 6.1.

For training on the laser ground truth, we follow the evaluation protocol of [46, 97]. This means that we select the frames 43, 71, 82, 87, 94, 120, 122 and 180 of the KITTI *training* dataset for training. The labels correct/incorrect are set by comparing the query depth maps with the laser ground truth using the standard

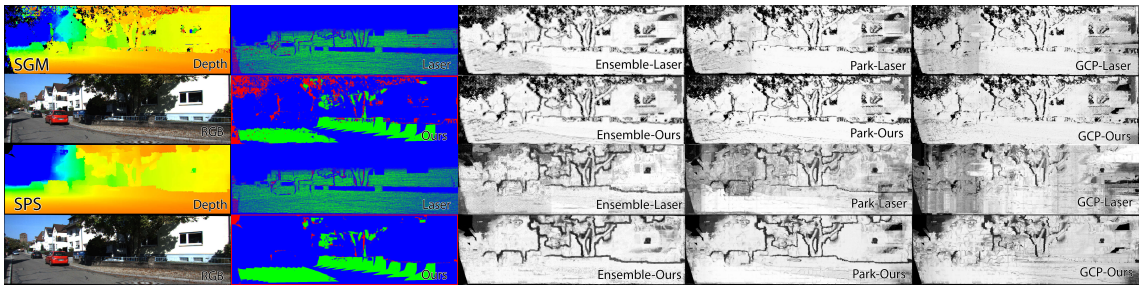


Figure 6.2: Qualitative results for sequence 102 of the **KITTI** training dataset. In the first column,we show the depth maps of SGM [108] and SPS [139] together with the color input image. The second column shows the resulting label images once produced with the laser ground truth (Laser) and once with our approach (Ours). Note that our approach only assigns a positive label to parts of the scene that are observed under significantly different view points (the car is making a turn to the left in the sequence). The remaining 3 columns show the confidence output of Ensemble [46], GCP [125] and Park [97] once trained on Laser and once on Ours. The confidence ranges from low (black) to high (white). Note the confidence output is much smoother for Ours and contains less artifacts (especially for GCP).

three pixel disparity threshold. Further on, we will mark a confidence measure trained on this data with the suffix "Laser". As our approach requires multiple images that view the same scene, we use the 195 sequences of 21 stereo pairs of the KITTI *testing* dataset for automatically generating our label images. Further on, we will mark a confidence measure trained on this data with the suffix "Ours". Example label images can be found in Figure 6.2. For testing, we once again follow the protocol of [46, 97] and evaluate the confidence measuring performance on the KITTI *training* dataset minus the eight sequences that were used for training on the laser ground truth. Thus, there is no overlap between training and testing for Laser as well as Ours. Also note that Ours has not seen a single ground truth laser scan. In training, we used all available training samples from the laser ground truth and roughly ten times this number from our automatically generated data. Note that this is less than one percent of all available training data. With this setup our implementation used ∼20GB of memory for training.

Results. In Figure 6.3 we show the mean, minimum and maximum AUSC values of the three confidence measures for all combinations of query algorithm and training data. In Tab. 6.2 we show the AUSC for each approach divided by the optimal AUSC over all evaluated sequences of the KITTI dataset.

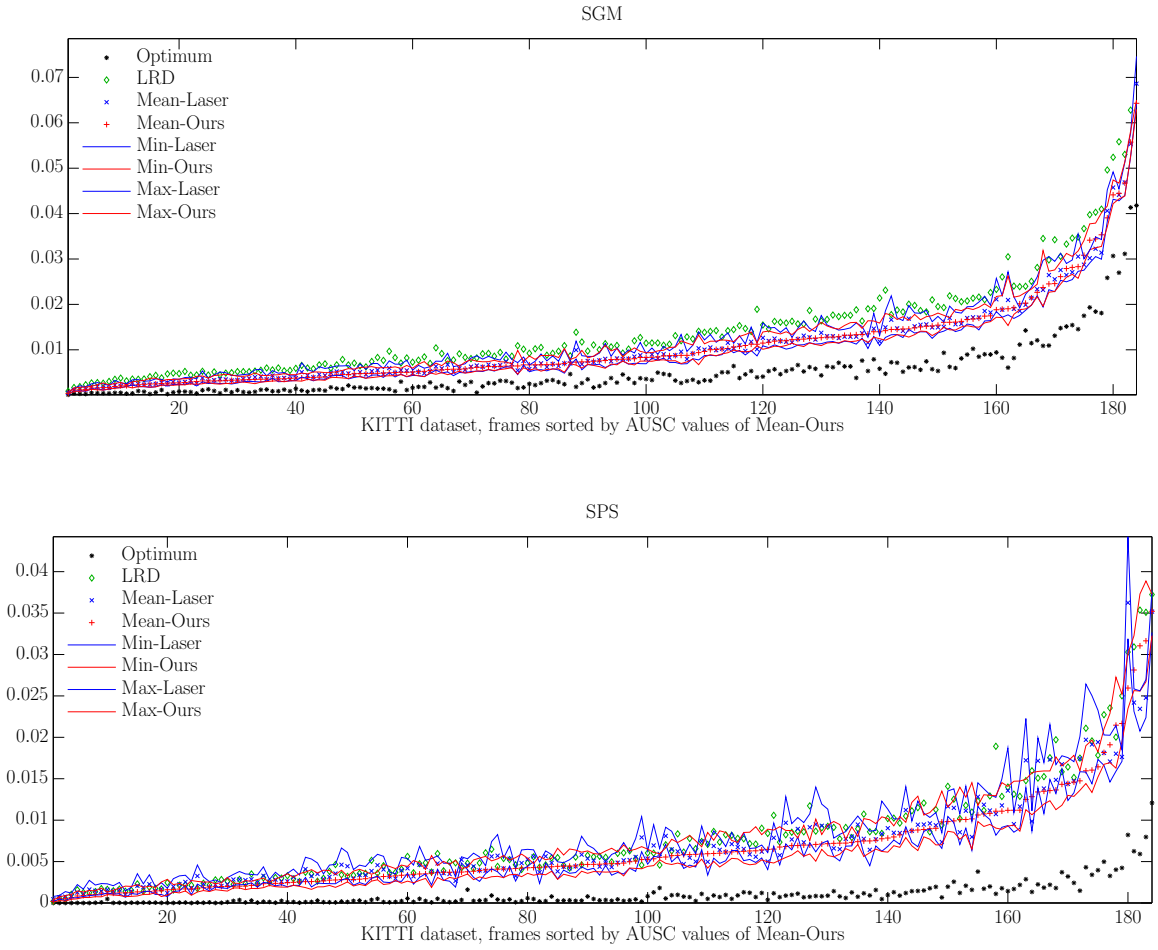


Figure 6.3: Mean, minimum and maximum AUSC values over the three confidence measures (Ensemble [46], GCP [125], Park [97]) for all frames of the **KITTI** training dataset minus the eight frames used for training. We display all combinations of query algorithm (SGM [108] and SPS [139]) and training data (Laser and Ours). The frames were sorted according to mean AUSC value of Ours. As a baseline method we also show the Left-Right disparity Difference (LRD). Note that Ours (red) is lower than Laser (blue) in most cases. For SGM, all approaches perform always better than LRD if they are trained on Ours, while if they are trained on Laser they sometimes perform worse (e.g. 142). For SPS stereo, the number of severe errors is significantly higher for Laser than for Ours (compare blue versus red peaks above 160).

	LRD	Ens.[46]	Park[97]	GCP[125]
SGM-Laser	2.81	1.97	1.93	2.50
SGM-Ours	2.81	1.95	**1.92**	2.45
Reduction	-	0.94%	0.78%	**2.02%**
SPS-Laser	7.60	5.86	6.23	8.28
SPS-Ours	7.60	**5.43**	5.61	7.95
Reduction	-	7.28%	**9.93%**	3.98%

Table 6.2: Area under the sparsification curve divided by optimal area on the **KITTI** dataset. We display all combinations of query algorithm (SGM [108] and SPS [139]), confidence measure (Ensemble [46], GCP [125], Park [97]) and training data (Laser and Ours). The reduction is computed as $1 - AUSC_{Ours}/AUSC_{Laser}$.

In all cases, using our training data resulted in a performance boost. In some cases the AUSC even dropped by 10%. A visual comparison of the difference in the confidence output can be found in Figure 6.2. Note that our training data leads to a smoother confidence output with significantly fewer artifacts.

As a matter of completeness, we executed our training data generation only on the eight same sequences that were used for training Laser. One has to note that the coverage of our approach depends on the camera motion and one of the sequences (180) contains no useful motion, which leaves our approach with 7 sequences. Using only this limited amount of training data, the AUSC increased by ∼10% for all approaches compared to using the 195 testing sequences. This is not surprising, as each of our training images can be considered as weaker compared to the laser ground truth, in the sense that consistency alone cannot uncover all errors and that the coverage of our labeling depends on the camera motion. But this experiment clearly shows that using ten times more "weak" training samples, which can be cheaply generated with our method, still leads to a better performance than fewer "strong" training samples.

6.1.3 Middlebury Dataset

The Middlebury2014 [112] dataset contains a set of 23 high resolution stereo pairs for which known camera calibration parameters and ground truth disparity maps obtained with a structured light scanner are available. The set is divided into 10 stereo pairs for training and additional 13 stereo pairs that we used for testing. The images in the Middlebury dataset all show static indoor scenes with varying diffi-

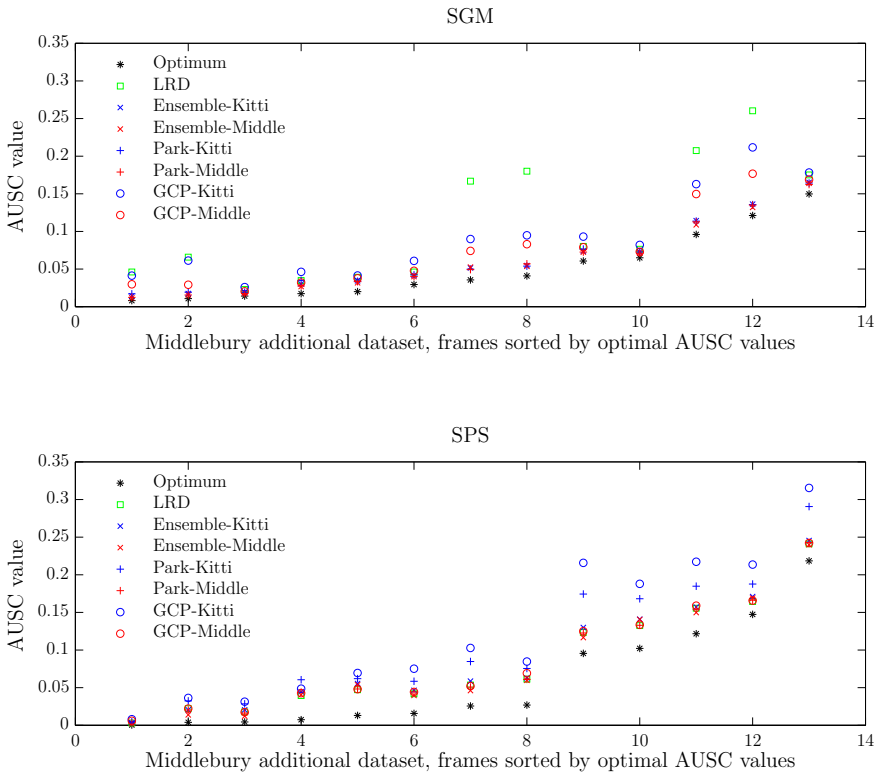


Figure 6.4: Area under the Sparsification Curve (AUSC) values for all 13 frames of the additional **Middlebury** dataset. The frames were sorted according to the optimal area under the curve value. We display all combinations of query algorithm (SGM [108] and SPS [139]), confidence measure (Ensemble [46], GCP [125], Park [97]) and training data (Kitti [42] and Middle [112]). As a baseline method we also show the Left-Right disparity Difference (LRD). Note that the red symbols (Middle) are in many cases drastically lower than their blue counter parts (Kitti).

culties including repetitive structures, occlusions, wiry objects as well as untextured areas.

Due to the limitation that only stereo pairs and no multi-view sequences are provided, we are not able to evaluate the accuracy performance of our ground truth generation. But we can still evaluate the performance of the confidence measures previously learned on the KITTI to evaluate their generalization performance from outdoor to indoor scenes. Figure 6.4 shows the resulting AUSC curve for SGM [108] and SPS [139], respectively. In Tab. 6.3 we show the AUSC over the optimal values.

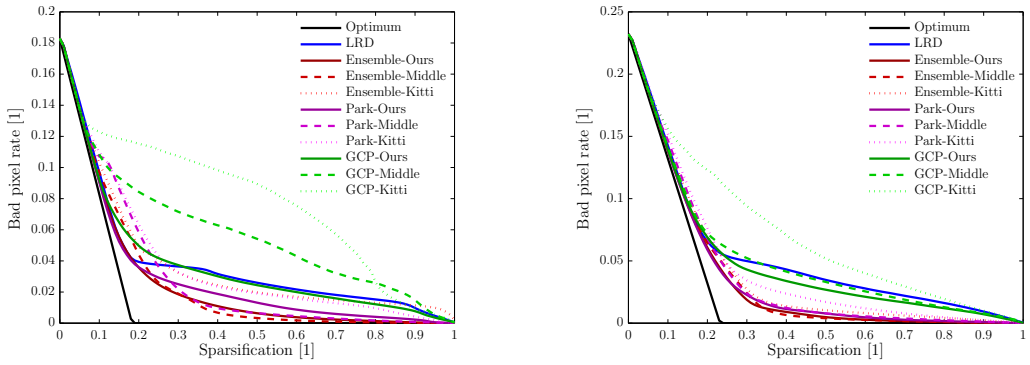
	LRD	Ens.[46]	Park[97]	GCP[125]
SGM-Kitti	2.10	1.24	1.25	1.78
SGM-Middle	2.10	**1.19**	1.20	1.50
Reduction	-	3.29%	3.30%	**15.86%**
SPS-Kitti	1.41	1.48	1.81	2.05
SPS-Middle	1.41	**1.39**	1.42	1.44
Reduction	-	6.32%	21.63%	**29.82%**

Table 6.3: Area under the sparsification curve divided by optimal area on the **Middlebury** dataset. We display all combinations of query algorithm (SGM [108] and SPS [139]), confidence measure (Ensemble [46], GCP [125], Park [97]) and training data (Kitti [42] and Middle [112]). The reduction is computed as $1 - AUSC_{Middle}/AUSC_{Kitti}$.

Results. For all combinations of query algorithm and confidence measure, training on the Middlebury increased the performance compared to training on the KITTI and evaluating on the Middlebury. The percentage of area reduction strongly depends on the used confidence measure. We assume that the large variation in area reduction (3%-30%) is caused by features which are very setup specific (e.g. distance to border). Despite the large reduction variation, all approaches benefit from training on the Middlebury rather than the KITTI. This means that tuning towards a special setup can make a large difference in performance.

6.1.4 Strecha Dataset

To further demonstrate the value of our approach, we analyze the sparsification performance in a completely different setup. For this experiment, we used the multi-view stereo dataset of Strecha et al. [126]. This dataset provides images together with camera poses and two ground truth meshes. From the two available meshes, the Herz-Jesu mesh is a good example that also active sensors have their limitations. In this mesh, all the thin structures (hand rails and bars) are simply missing. As these errors would cause problems in the evaluation, we only used the second dataset (Fountain), which does not contain any thin structures. This dataset consists of 11 images aligned to the ground truth mesh. For this experiment, we split the images into a training set containing 3 image pairs and a test set with 2 image pairs. The training pairs are made of images 0+1, 4+5 and 8+9 and the testing pairs of 2+3 and 6+7. Each pair was then rectified using [108]. As the SPS implementation [139] failed to produce any reasonable output on this kind of data, we limit this experiment



(a) SGM Pair 2+3.

(b) SGM Pair 6+7

Figure 6.5: Sparsification curves for testing stereo pair on the **Strecha** fountain dataset. We display all combinations of confidence measure (Ensemble [46], GCP [125], Park [97]) and training data (Kitti [42], Middle [112] and Ours) for the SGM output [108]. As a baseline method we also show the Left-Right disparity Difference (LRD).

	LRD	Ens.[46]	Park[97]	GCP[125]
Kitti RA	2.12	1.81	1.91	3.54
Middle RA	2.12	1.43	1.59	2.60
Ours RA	2.12	**1.40**	1.51	2.01
Kitti Red	-	22.34%	21.04%	**45.30%**
Middle Red	-	1.86%	5.02%	**23.53%**

Table 6.4: Area under the sparsification curve divided by optimal area (Relative Area RA) on the **Strecha** fountain dataset. We display all combinations of confidence measure (Ensemble [46], GCP [125], Park [97]) and training data (Kitti [42], Middle [112] and Ours) for the SGM output [108]. The reduction is computed as $1 - AUSC_x/AUSC_{Ours}$ for each confidence measure.

to the SGM [108] reconstruction.

Results. In this setup our ground truth generation reached an **accuracy of 95.1%** (STD: 2.6%) at a coverage 30.4% (STD: 5.0%). In Figure 6.5 we show the resulting two sparsification curves and the AUSC reduction statistics in Tab. 6.4. All combinations of query algorithms and confidence measures performed better trained on the Middlebury than on the KITTI. In all cases, the performance was further increased by tuning them specifically to this scene in using our automatically generated training data.

6.1.5 Discussion

In this series of experiments, we have shown that confidence measures work better if they are trained and executed in the same domain. Our method provides a way of obtaining the necessary training data at a low cost, which makes it possible to acquire training data in high diversity and quantity. Although we can make no guarantees for the correctness of each generated training sample, our experiments demonstrate that average quality of the generated samples is high enough to obtain strong confidence measures even in the complete absence of traditional ground truth. This property is very important, as in our following experiments we will enter a domain, where it is close to impossible to obtain reliable ground truth for all present scene structures.

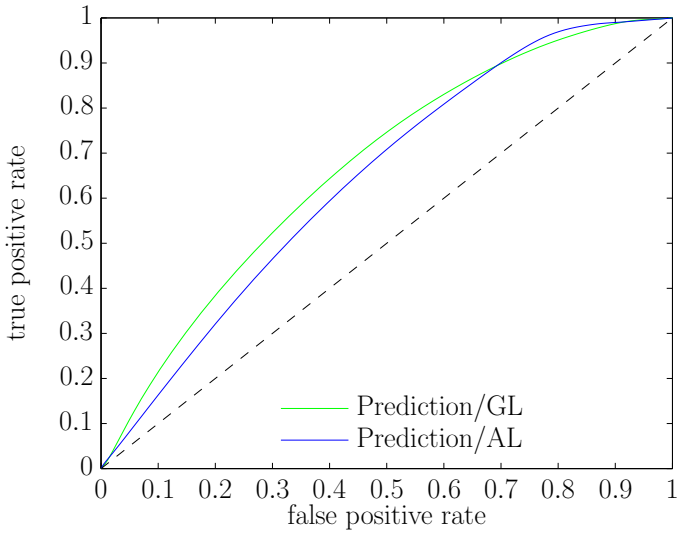
6.2 MVS Confidence Prediction

In this section, we move from two-view stereo confidence measures to MVS Confidence Prediction. We would like to emphasize the difference between the two terms. A confidence measure estimates the reliability of a depth measurement in a fixed setup given both images, the complete cost volume and two complete depth maps. In contrast, MVS Confidence Prediction aims to predict the reliability of a future depth measurement – without having the real measurement or even all necessary images at prediction time. In the first part of this section, we show what this difference means in hard numbers on the KITTI dataset [42]. In this experiment, we use our complete MVS Confidence Prediction framework (including Semantic Texton Forest (STF) [122]) instead of two-view stereo confidence measures. In the second part, we use a challenging multi-view dataset to evaluate what the system can learn about two different MVS approaches in relation to a variety of scene structures and camera constellations without any ground truth.

6.2.1 KITTI2012 Dataset

For learning, we follow the same procedure as in the previous chapter and use the 195 sequences of 21 stereo pairs of the testing dataset for automatically generating our label images. We treat each stereo pair as a distinct cluster and use a semi-global matcher with left-right consistency check (SURE [108]) as the query algorithm. As before, we evaluate the label accuracy and the average Area Under the Sparsification Curve (AUSC), although with a slightly different setup. While stereo confidence measures try to decide which depth values cannot be trusted from an already computed depthmap, our aim is to predict which kind of structures cause more problems than others. Thus, we remove all regions from the Lidar ground truth, which are not visible in both color images (including object occlusions).

Results. In Figure 6.6 we show the resulting receiver operator characteristic (ROC) of the confidence prediction as well as the label accuracy. With this setup we reach a labeling accuracy of **98.7%** while labeling 35% of the ground truth pixels (which is very similar to the results in the previous chapter). Although the Lidar ground truth and our approach label very different regions of the image, the accuracy and the receiver operating characteristic of the predicted confidence (Figure 6.6) are very



Qry/Ref	ACC
AL/GL	98.7%
PL/GL	69.5%
PL/AL	68.7%

Figure 6.6: Two-View Stereo Experiment. On the left side, we show the receiver operating characteristic of the predicted confidence in relation to the Ground truth Labels (GL) and the Automatically generated Labels (AL). On the right side, we show the obtained classification accuracy (ACC) for three different combinations of query labeling (Qry) and reference labeling (Ref). PL stands for Predicted Labels.

similar for both reference labelings. This means that the missed regions do not seem to have a significant impact on what the system learns. For the sparsification, we obtain a relative AUSC of 3.15 (obtained AUSC divided by optimal AUSC). This means that the AUSC is **39%** lower than random sparsification with 5.15. This is a strong indication that the system learned to predict regions which are difficult to reconstruct for the semi-global matcher. In Figure 6.7, we show an example of a systematic problem learned by our approach.

For the matter of completeness, we also analyze the sparsification performance of the STF [122] with the exact same setup as in the previous chapter (including the training data generation). With this setup, STF reaches a relative AUSC of 6.63. It is not surprising that STF cannot reach the sparsification performance of stereo specific sparsification approaches (e.g. left-right difference with 2.81), as the STF only uses color information of a single image and thus has no chance to reason about occlusions. Nevertheless, the STF was able to extract some high level knowledge in which regions the chances of failure are higher and thus still obtains a 31.4% lower AUSC value than random sparsification (9.65).

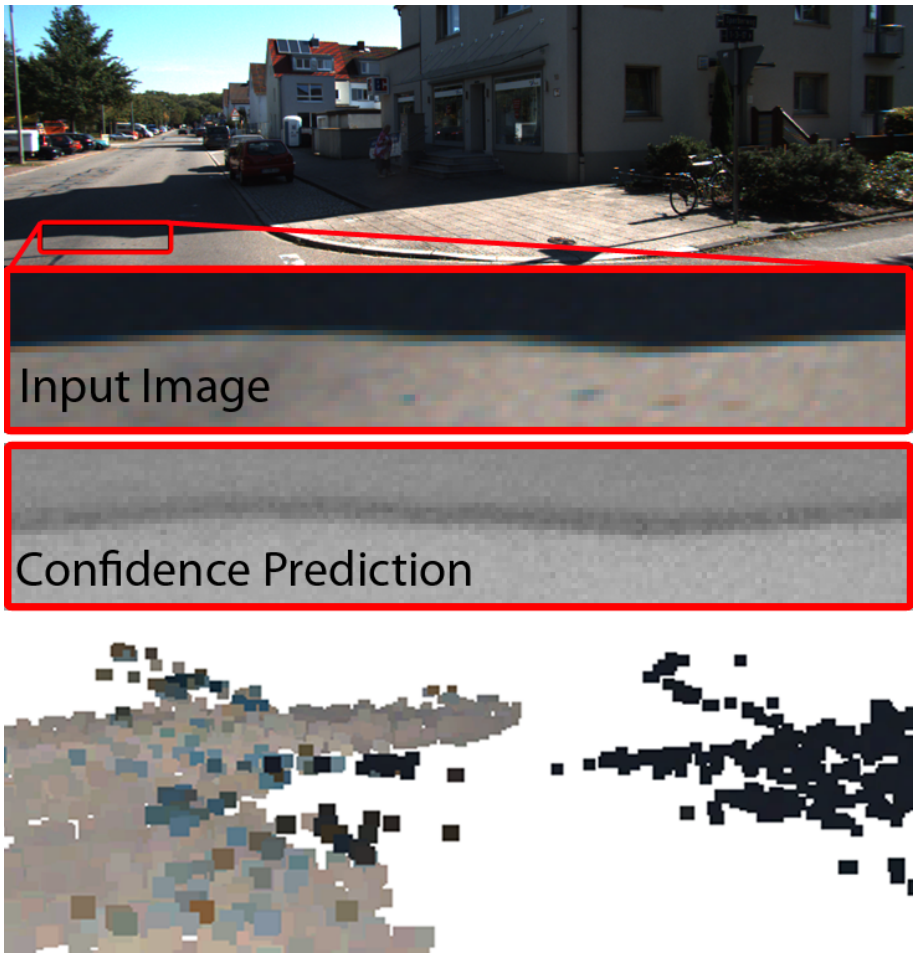


Figure 6.7: Example of problems learned with our MVS Confidence Prediction approach on the KITTI dataset. The used algorithm (SURE [108]) seems to have a systematic problem at shadow edges, where the high gradient at the shadow edge leads to wrong depth estimates in the shadow region. In the bottom image, we visualize this problem by unprojecting the depth map and showing the resulting point cloud. Instead of a flat and continuous road, the algorithm underestimates the depth in the transition to the shadow region. This leads to wrong measurements that "float" over the road surface. Note that the confidence prediction in this region is very low (dark), which indicates that our approach successfully learned to detect this systematic problem from a single color image before the reconstruction occurs.

6.2.2 Valcamonica Dataset

For the second dataset, we have chosen a reconstruction scenario in a closed real-world domain, where the task is the 3D reconstruction of prehistoric rock art sites in the Italian valley of Valcamonica. The recorded dataset consists of over 5000

Figure 6.8: Examples of the Valcamonica Dataset. In the top row, we show three images from far away, in the middle, three slanted views and ,in the bottom row, three close-ups. All sites contain a limited set of 3D structures (mainly rock, grass, trees, bridges and markers).

images of 8 different sites (see Figure 6.8). The images contain a great variety of viewing angles and acquisition scales. The camera to scene distance varies from 2 to 50 meters. The whole environment contains a well-defined set of 3D structures (mainly rock, grass, trees, bridges, signs and markers). These structures dominate nearly all sites in the region (hundreds), which makes this a perfect example for learning and predicting domain specific properties of a query algorithm.

For generating camera triplets, we used $t = 5$ triangulation bins and $i = 20000$ iterations. The lowest triangulation angle bin starts at a minimum angle of 4° and ranges to double that value, where the next bin starts. On each resulting triplet, we execute a query algorithm three times at different image resolutions (levels 1, 2 and 3 of an image pyramid). For defining sufficiently different view points, we used the a minimum angle of $\alpha_{min} = 10^\circ$ and a minimum scale difference of $s_{min} = 2$. For assessing if measurements support each other, we used a two sigma bound (i.e.

$\sigma_{max} = 2$).

We evaluate two query algorithms for the dense 3D reconstruction. The first query algorithm is based on semi-global matching SURE-3 [108], but can use more than two views for improving the reconstruction accuracy. In contrast, our second query algorithm PMVS-3 [32] tries to densify an initial sparse 3D reconstruction through iterative expansion. As it enforces the visibility consistency already in the reconstruction procedure, many parts of the scene are simply missing in the reconstruction output. We noted that the number of seed points with only 3 cameras is very low, and thus we also executed PMVS with its own clustering algorithm [31] and a much higher cluster size of 40 cameras (denoted as PMVS-40). Due to insufficient overlap, we assume that all measurements are correct and only used the missing data detection for obtaining negative training data in this setup.

Results. For the quantitative evaluation of this experiment, we performed leave-one-out cross validation across the 8 sites, i.e. we train on 7 sites and test on the remaining. This led to the following classification accuracies (with standard deviation): PMVS-40: 76.4% (STD: 5.1%), PMVS-3: 81.1% (STD: 4.2%) and SURE-3: 65.3% (STD: 6.1%).

Within this context, we also analyzed the influence of regular grid sampling on the prediction performance. For small grid sizes, the classification error stays nearly the same (relative error increase is below 1% for 4 pixels), while for larger grid sizes it declines gradually (below 3% for 16 pixels and below 7% for 64 pixels). This means that regular sampling can drastically reduce the computational load of the prediction with only a small decrease of the prediction performance.

But much more important than these numbers seems the question: What did the system learn about the different algorithms? To answer this question, we display the confidence prediction for certain details in the dataset in Figure 6.9.

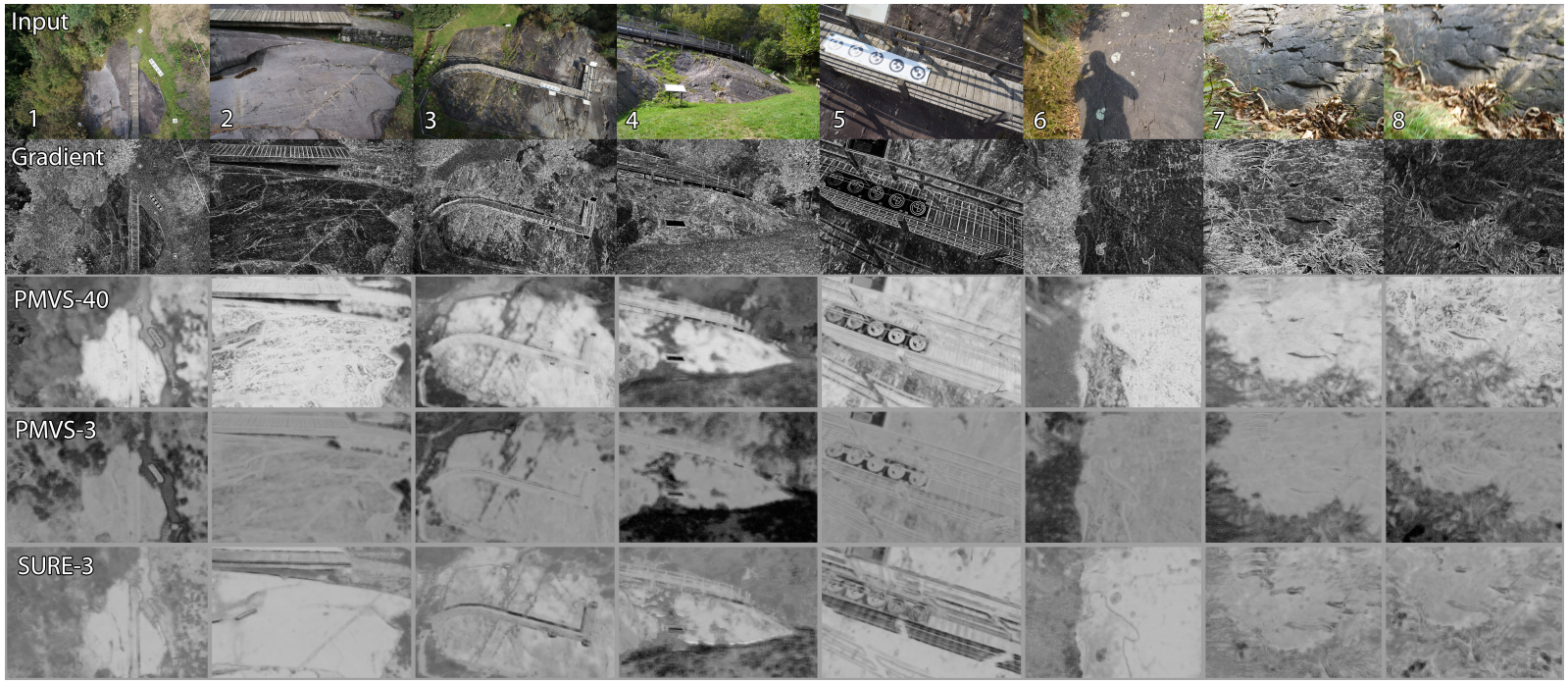


Figure 6.9: MVS Confidence Prediction. From top to bottom: Zoomed input images, gradient images and the predicted confidence (PMVS-40, PMVS-3, SURE-3). If we compare the gradient images with the confidence prediction, it is apparent that much more than simple gradients have been learned. While PMVS seems to require a minimum image gradient (sharp image (7) vs. motion-blurred image (8)), the consistency of SURE does not depend so strongly on this property. This seems reasonable as PMVS requires the image gradient to constrain its local patch optimization, whereas SURE can consistently interpolate regions with low gradient through semi-global matching. For both approaches, the confidence of smooth surfaces (e.g. rock and bridge) is significantly higher than for high frequency structures like vegetation. *Grass* seems to be a very special case. If it is shortly trimmed and viewed from far above (3), it behaves similar to random texture and can be nicely reconstructed. However, if it is too long or viewed more closely (4), the high-frequency structure of grass is more dominant and it becomes nearly impossible to reconstruct. This property was captured for both approaches. One of the most remarkable differences between the two approaches is that SURE's confidence is low at shadow boundaries (5+6), whereas PMVS' confidence rises. For SURE even object boundaries seem to be implicitly encoded in the confidence (2), as the output at discontinuities is often incorrect.

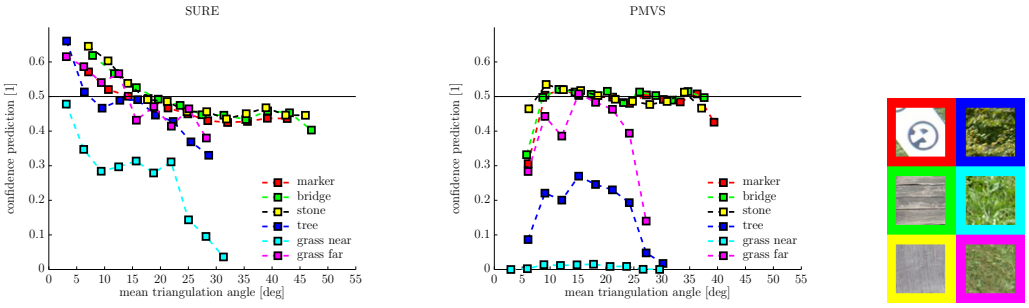


Figure 6.10: Dependence of the confidence prediction on the triangulation angle and the 3D structure. On the right, we display the patches (50 × 50px) which we used to produce the curves. These curves show the confidence prediction within angular bins (20 bins between min and max). The curves stop if less than 1% of the collected triangulation angles fall within a bin. For both approaches (SURE-3 and PMVS-3), there is a significant difference between smooth surfaces (marker, bridge, stone) and high frequency structures (tree, grass). The predicted confidence is to some extent correlated with the degree of non-planarity of a structure. While grass viewed from far away is quite easy to reconstruct, the same grass viewed close up becomes very hard to reconstruct. For both approaches, the chance for reconstructing highly non-planar structures above 30° is virtually zero.

For all setups, our approach has learned that vegetation, despite having a strong image gradient, is harder to reconstruct than smooth surfaces (e.g. rock or bridge). If we compare PMVS to SURE, we see at first glance that PMVS has overall a lower confidence. This is mainly due to the fact that PMVS was developed for a larger number of cameras and the small number of seed points between the three cameras leads to many missing parts. The most outstanding difference between PMVS and SURE seems to be shadow object boundaries. For SURE, the confidence values for shadow boundaries are low, despite the huge gradient at this position. If we look at images (5) and (6), the both approaches outline the shadow boundaries. While the SURE confidence is lowered, the PMVS Confidence is higher. In general, it is quite reasonable to lower the confidence values at shadow boundaries, as the shadows tend to move quite significantly over the acquisition period. Thus, matching shadow boundaries leads to wrong depth estimates. We assume that the difference in the output density is the reason why only SURE seems to have learned this very useful property.

Now let us analyze what the system learned about the two algorithms in relation to scene structures and triangulation angle. In Figure 6.10, we show the confidence

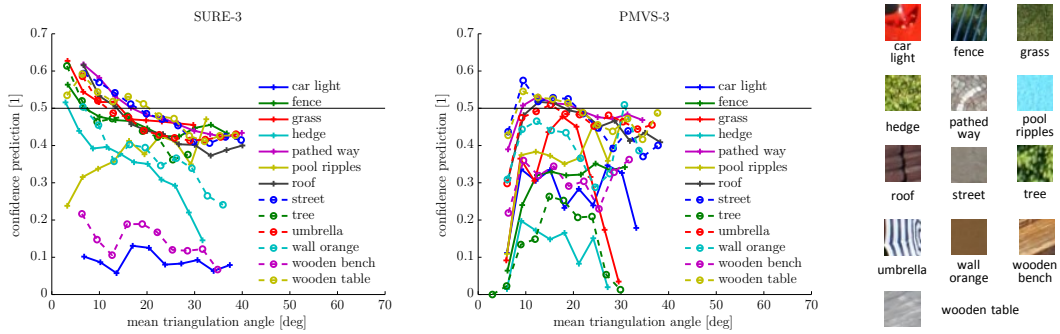


Figure 6.11: Dependence of the confidence prediction on the triangulation angle and the 3D structure with patches of the House Dataset. On the right, we display the patches (50×50px) which we used to produce the curves for SURE-3 and PMVS-3. These curves show the confidence prediction within angular bins (20 bins between min and max). The curves stop if less than 1% of the collected triangulation angles fall within a bin.

prediction for six different structures. From this experiment we can draw several conclusions. First, the 3D structure of the scene has a significant influence on how well something can be reconstructed under a given triangulation angle. The more non-planar a structure is, the harder it is to reconstruct at large triangulation angles. Second, the two analyzed approaches react very differently to a change in triangulation angle. While for SURE the confidence is always highest for very small angles, PMVS' confidence stays constant for smooth surfaces. In the case of non-planarity, SURE is clearly more robust than PMVS.

While it is highly interesting to see what the system "thinks" about structures it has already seen, we found it also interesting to evaluate the same for "unseen" structures. Thus, Figure 6.11 shows the same evaluation for patches on a completely different scenario, i.e. the reconstruction of single family houses in an sub-urban environment (Note that we will also evaluate our MVS Prioritization approach in the same environment - see Chapter 6.4). If we compare the curves of both datasets, we would like to observe a similarity between structures which are similar in 3D. For most structures, this works very well. For example, for both MVS approaches the random forest correctly classifies smoother structures (street, grass, paved way,wooden table, roof and umbrella) and high frequency structures (tree and hedge). Note that in training the random forest actually never seen the classes street, paved way, roof or umbrella. For some other structures, where nothing remotely

similar was observed in training, the random forest predicts something unexpected. E.g., the car lights, the orange wall and wooden bench seem to interpreted as a high frequency structure (similar to tree or hedge). For pool ripples, both random forest think that a higher triangulation gives better chance of a good reconstruction. Overall however, the correct predictions seem to outweigh the incorrect predictions, which indicates a reasonable generalization performance. This indication is further strengthened by our experiments on MVS Prioritization in Chapter 6.4.

6.2.3 Discussion

In this section, we first evaluated the difference between two-view confidence measures and our MVS Confidence Prediction with Semantic Texton Forests (STFs). This evaluation has shown two things. First, that two-view confidence measures, which have both images, the complete cost volume and two complete depth maps at their disposal, are much more powerful than our MVS Confidence Predictor which only uses a single color image for its prediction. Second, despite this extremely limited input, the MVS Confidence Predictor was able to detect systematic problems of the stereo algorithm. E.g. it correctly learned that there are problems with the depth estimation at shadow boundaries. This means that even from a single color image, it is possible to learn general problems of an algorithm.

In the second part of this section, we analyzed what the MVS Confidence Prediction can learn about two different MVS algorithms and what this tells us about these two algorithms. First of all, this experiment has shown that there is a significant difference between the two MVS algorithms. Second, the MVS Confidence seems to be strongly influenced by the triangulation angle and the individual scene structures. PMVS, which tries to locally optimize patches in 3D, seems to have significantly more problems with reconstructing high frequency structures (such as vegetation) than SURE, which is based on semi-global matching with a census transform cost. However, through the local patch optimization PMVS' behavior is more stable on well-textured planar structures. In general, we see that high frequency structures (such as vegetation) are more likely to be correctly reconstructed at a low triangulation angle, while smooth surfaces (such as wood planks) can also be reconstructed at much larger angles.

In the following experiments, we will use the MVS Confidence Prediction to

improve the input for the MVS reconstruction step (i.e. the images and their constellation). This approach allows us to avoid bad image constellations with respect to the scene structures and the given MVS algorithm and thus allows us to obtain high quality 3D reconstructions with a high efficiency.

Figure 6.12: In this experiment, we aim to reconstruct the archaeological site *Seradina Rock 12C* with some of the surrounding vegetation. The region of interest is marked in red. Related to this experiment, we show the work flow from a user's perspective in Video [86].

6.3 Automated Image Acquisition for Multi-View Stereo

The main aim of our acquisition approach is to record images in such a way that the MVS algorithm can reconstruct the scene of interest with as much detail as possible and at the same time fulfill a set of quality constraints, such as ground resolution and 3D accuracy. One of the targeted application areas of the resulting 3D model is a virtual reality environment, where a user can visit the archaeological sites from a remote location. To enable an immersive user experience, we are not only interested in completely reconstructing the smooth surfaces, but we also want the approach to reconstruct as much as possible of the surrounding environment (including the vegetation).

6.3.1 Experimental Procedure

For evaluating our image acquisition approach, we focus on one site in Valcamonica, namely Seradina Rock 12C. The rock surface (17×13 m) is covered with prehistoric

rock carvings and is partly occluded by the surrounding vegetation (Figure 6.12). We placed 7 fiducial markers in a circle around the rock of interest and measured them with a Leica total station. These markers can be automatically detected in the images and are used for geo-referencing the offline reconstructions [109]. Additionally, a ground truth mesh of the rock (not the surroundings) was obtained through terrestrial laser scanning (TLS) in the same coordinate system two years before. The mesh has a resolution of 8 mm edge length and the accuracy of the laser scanner (Riegl VZ-400) is 5 mm. We use this mesh to evaluate the resulting 3D accuracy.

To evaluate our image acquisition approach, we first run different view planning algorithms on-site and then analyze the effective reconstruction output, which is computed off-site. As we also desire a reconstruction of the surrounding environment (which is dominated by vegetation), we use SURE [108] as MVS algorithm.

For this experiment, we run three versions of the proposed approach. The first version is our full approach (F5x4), where we let the algorithm plan 4 camera triplets per iteration for a total of 5 iterations. In the second version (F1x20), we let our approach plan the same number of total triplets (20) but in a single iteration, i.e. we disable the incremental geometry updates. The third version (NP5x4) is exactly the same as F5x4 but without the prediction to constrain the triangulation angle. As a baseline method, we use grid planning with 80 percent overlap. For the grid computation, we estimate the dominant plane in the region of interest of the mesh reconstruction. All approaches share the same set of parameters. The quality requirements were set to $x = 3$, $g_d = 8\,\text{mm}$ and $a_d = 8\,\text{mm}$ with $\alpha = 0.5$. The safety distance was set to 5 m at a maximum octree [73] resolution of 2 m and the minimum camera overlap for registration to $o_{min} = 50\%$. The triangulation angle was binned in $b = 9$ steps of 5° from 0° to $\gamma_{max} = 45°$. For the inverse visibility estimation we set the parameters such that the planning approximately takes 5 seconds per planned triplet, i.e. $N_t = 2000$, $N_p = 5000$ and $N_v = 200$ with $\phi = 120°$. This parameters resulted in an effective execution time per triplet of 5.98 seconds (STD: 2.19) over all experiments on a HP EliteBook 8570w. The confidence was evaluated on a regular grid with a step size of 8 pixels, which resulted in a confidence prediction time of ∼2 sec/image. We acquire the images with a Sony Nex-5 16Mpx camera mounted on an Asctec Falcon8 octocopter.

To evaluate the coverage and the quality fulfillment, we first obtain a geo-

	Init	Grid	F5x4	F1x20	NP5x4	G+F5x4	G+F1x20	G+NP5x4
cov	53.5 ± 1.2	56.0 ± 1.2	$\mathbf{65.6} \pm 1.6$	$\mathbf{66.6} \pm 1.4$	56.7 ± 1.4	$\mathbf{69.5} \pm 1.5$	67.0 ± 1.5	57.2 ± 1.2
f_{res}	17.9 ± 1.3	43.9 ± 2.6	42.2 ± 2.6	$\mathbf{47.5} \pm 2.7$	29.3 ± 2.3	$\mathbf{52.8} \pm 2.7$	$\mathbf{55.3} \pm 2.7$	46.8 ± 2.6
f_{unc}	15.5 ± 0.3	$\mathbf{22.8} \pm 0.4$	21.2 ± 0.5	20.7 ± 0.5	19.9 ± 0.5	$\mathbf{27.7} \pm 0.5$	26.2 ± 0.4	25.6 ± 0.4
f	16.7 ± 0.8	$\mathbf{33.4} \pm 1.5$	31.7 ± 1.5	$\mathbf{34.1} \pm 1.6$	24.6 ± 1.4	$\mathbf{40.2} \pm 1.6$	$\mathbf{40.7} \pm 1.6$	36.2 ± 1.5

Table 6.5: Fulfillment statistics in percent. We show the coverage of the region of interest *cov*, the resolution fulfillment f_{res} and the uncertainty fulfillment f_{unc}, as well as the overall fulfillment f as defined in Sec. 3.3. We display the mean value and the standard deviation over the three surface meshes. We mark all results within the standard deviation of the best method with a bold fond. In the first column, we show the results with only the 19 initialization images, then we show the four standalone approaches. The last three columns show a combination of the standard grid approach (Grid) with the other approaches.

referenced sparse reconstruction from all flights on the day of the experiment (∼500 images). Then we obtain three meshes, one based on [71, 135] and the two others as described in Section 3.3. As we know that these meshes will contain errors, we only use these meshes as a guideline for the evaluation. Within the region of interest, we split all triangles to have a maximum edge length of 8 cm. For each taken image, we first compute the triangle visibility. Then we produce a depthmap from all SURE 3D points linked to the image. If the measured depth is either larger than or within 24 cm of the triangle depth, we accept the 3D point as a valid measurement of the triangle. Based on the links of the 3D measurement, we then compute the fulfillment of the triangle analog to Section 3.3. Instead of the MVS Confidence f_{conf}, we use the actual coverage of the triangle. This means for a given triangle, we project this triangle into the key view and analyze the percentage of pixels with a valid measurement within the 2D projection of the triangle. Finally, this results in a set of fulfillment and coverage scores over all triangles in the region of interest.

In field, all approaches were initialized with 19 images taken in grid at a height of 50 m above the lowest point of the site. The region of interest was marked in one of the initialization images, such that it is centered on the rock and includes a few meters of the surrounding vegetation (Figure 6.12). Landing and take-off are performed manually, while the view plans are executed autonomously by the UAV.

6.3.2 Results

In Figure 6.13, we show the resulting view plans for each approach. For each of our approach variants, we executed SURE only on the three images of the triplets. Like

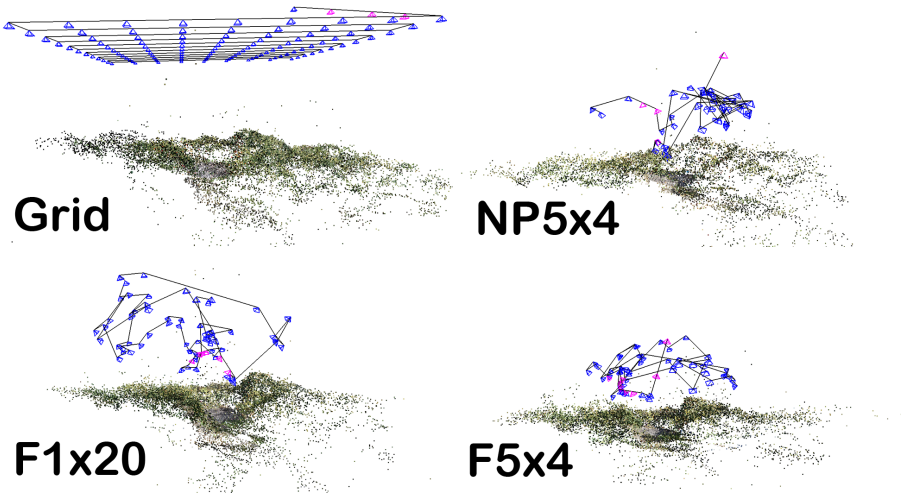


Figure 6.13: Resulting view plans for all approaches. We show traditional grid planning (Grid), our approach without prediction (NP5x4), our approach with prediction without geometry update (F1x20) and our full approach (F5x4). The blue cameras are regular or triplet cameras, while the pink cameras ensure sufficient overlap for sequential registration.

this we can evaluate the general success rate of view planning variants in analyzing on which triplets SURE succeeded to produce any 3D output. Without the confidence prediction the success rate is very low (**18%** for **NP5x4**). This shows the gap between theory and practice. While in theory a large triangulation leads to a small 3D uncertainty, the matching becomes much more difficult and only flat surfaces survive. However, with the proposed confidence prediction we were able to reach a perfect success rate for our full approach (**100%** for **F5x4**), and still reached an acceptable success rate without the reconstruction updates (**80%** for **F1x20**).

In Table 6.5 we display the effective fulfillment statistics of all approaches in the region of interest. Of the standalone approaches, F1x20 and Grid take the lead, but are closely followed by F5x4. The worst performance was reached by NP5x4. While the dense grid performs well on the overall fulfillment, we can see a **10% gap** in the scene coverage, where F5x4 and F1x20 lead with nearly equal results. F1x20 performs slightly better than F5x4, because F1x20 found a sweet spot in the center above the rock for a single triplet, where it was able to drop below the tree line and acquire a close up of the rock.

If we combine the results of the dense grid (Grid) with the proposed approach, we achieve the overall best results. All evaluated measures improve significantly, which is an indication of a symbiosis between the approaches. This suggests that for the

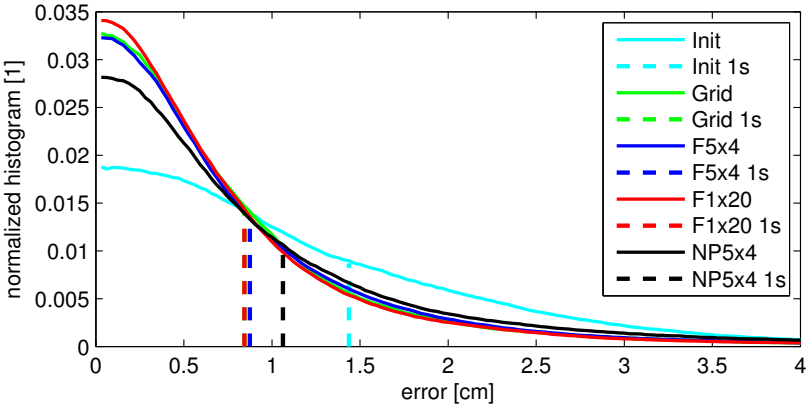


Figure 6.14: Error histogram on the rock surface. We show the normalized histograms of the error distribution and the 1 σ bound in which 68.3% of all measurements lie. Grid and F1x20 share the same error bound.

given scene (which is quite flat for many scene parts) an initial grid reconstruction with a subsequent refinement with the proposed approach is recommended. Note that if the scene complexity increases and a grid plan can no longer be executed safely (e.g. underneath a forest canopy or indoors), our planning approach is still applicable.

If we take a look at the error distribution in relation to the ground truth of the rock surface (Figure 6.14), we can see that our approach and grid planning achieve very similar results. Note the Grid only covered 87.4% of the rock surface, while all others covered significantly more: F5x4 covered 97.9%, F1x20 94.7% and NP5x4 94.0% (see Figure 6.15 for more details). This is a very promising result, as we only allowed our approach to use the planned triplets and no combination between them, while we put no such restrictions on the Grid approach. Furthermore, many of the triplets focused on the surrounding vegetation and the overall number of acquired images by our approach is lower than for the Grid approach (60 vs. 108 images). Thus, our approach achieved a high accuracy at a higher coverage with fewer images, which can also be observed visually in Figure 6.16 or in Video [85].

6.3.3 Discussion

In this experiment, we applied our image acquisition approach to a challenging scene, which is made up of a combination of smooth surfaces (rock and short grass) and high frequency structures (such as trees and fences). Compared to traditional grid

planning, we achieve a significantly higher scene coverage (while retaining the desired 3D accuracy) requiring only half of the images. The main key to this significant performance gain is our MVS Confidence Prediction. This prediction allows us to plan image constellations, which are perfectly suited for the used MVS stereo algorithm and thus lets us minimize the number of require images. If we switch off the MVS prediction, we can see a significant drop in performance. The reason for this drop is that 3D accuracy term in our fulfillment function pushes the cameras too far away from each other, such that the theoretical 3D accuracy is high, but the MVS algorithm is unable to find reliable correspondences. With the MVS prediction, the system knows the limits of the MVS algorithm with respect to the scene structure and only plans feasible camera constellations. In the following experiment, we show that this idea cannot only be used for Automated Image Acquisition, but also to improve the efficiency of MVS on a fixed set of images.

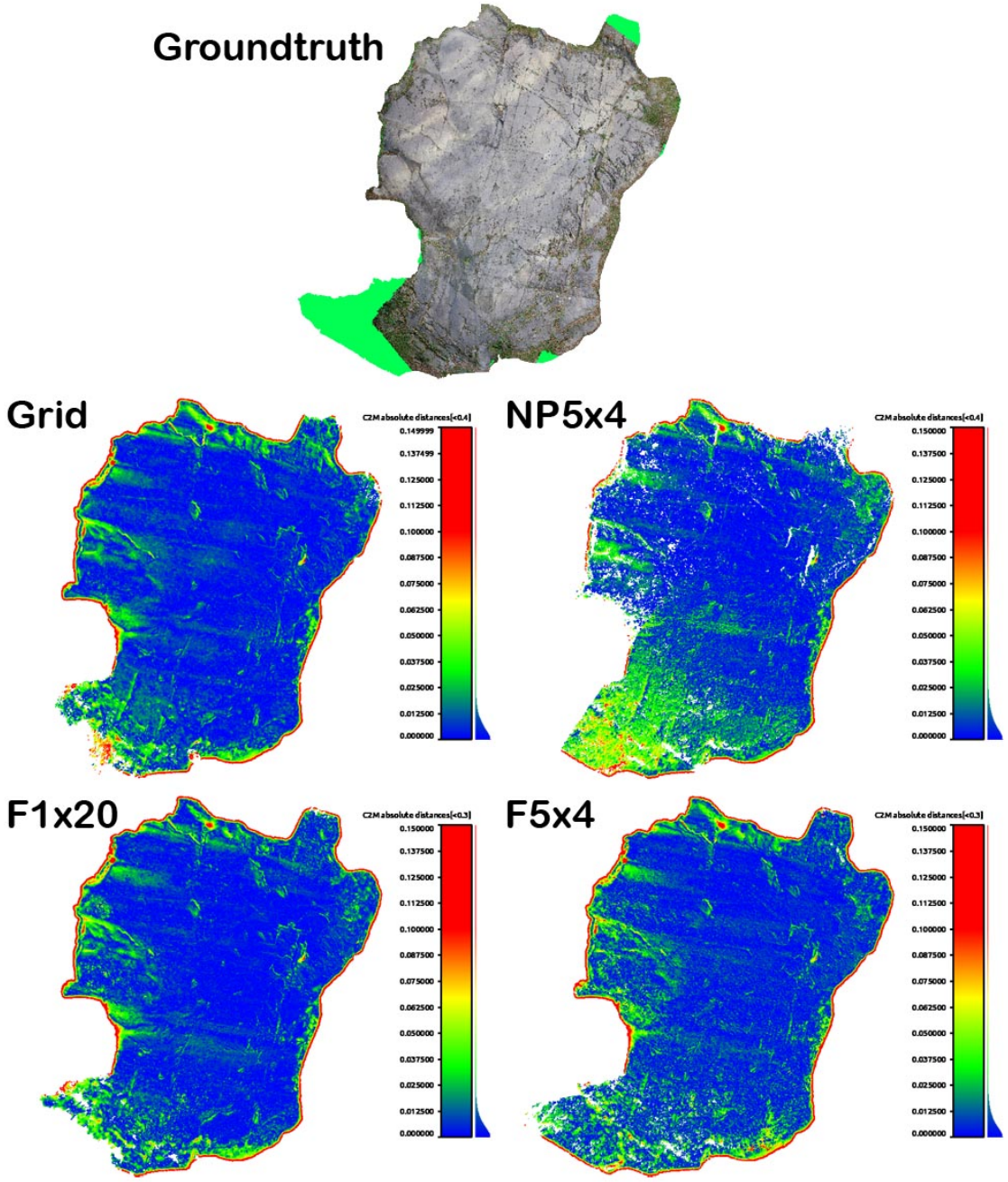


Figure 6.15: Distance to the ground truth. On the top we show the ground truth mesh which has been acquired with terrestrial laser scanning and was partly textured with UAV acquired images. The other images show the color coded distance of the final reconstructions to the ground truth mesh. Note that the error of Grid, F1x20 and F5x4 is very similar, while NP5x4 has a much larger error in the bottom part of the rock. Further notice that F5x4 has the largest coverage of the rock surface in the lower left corner of the rock.

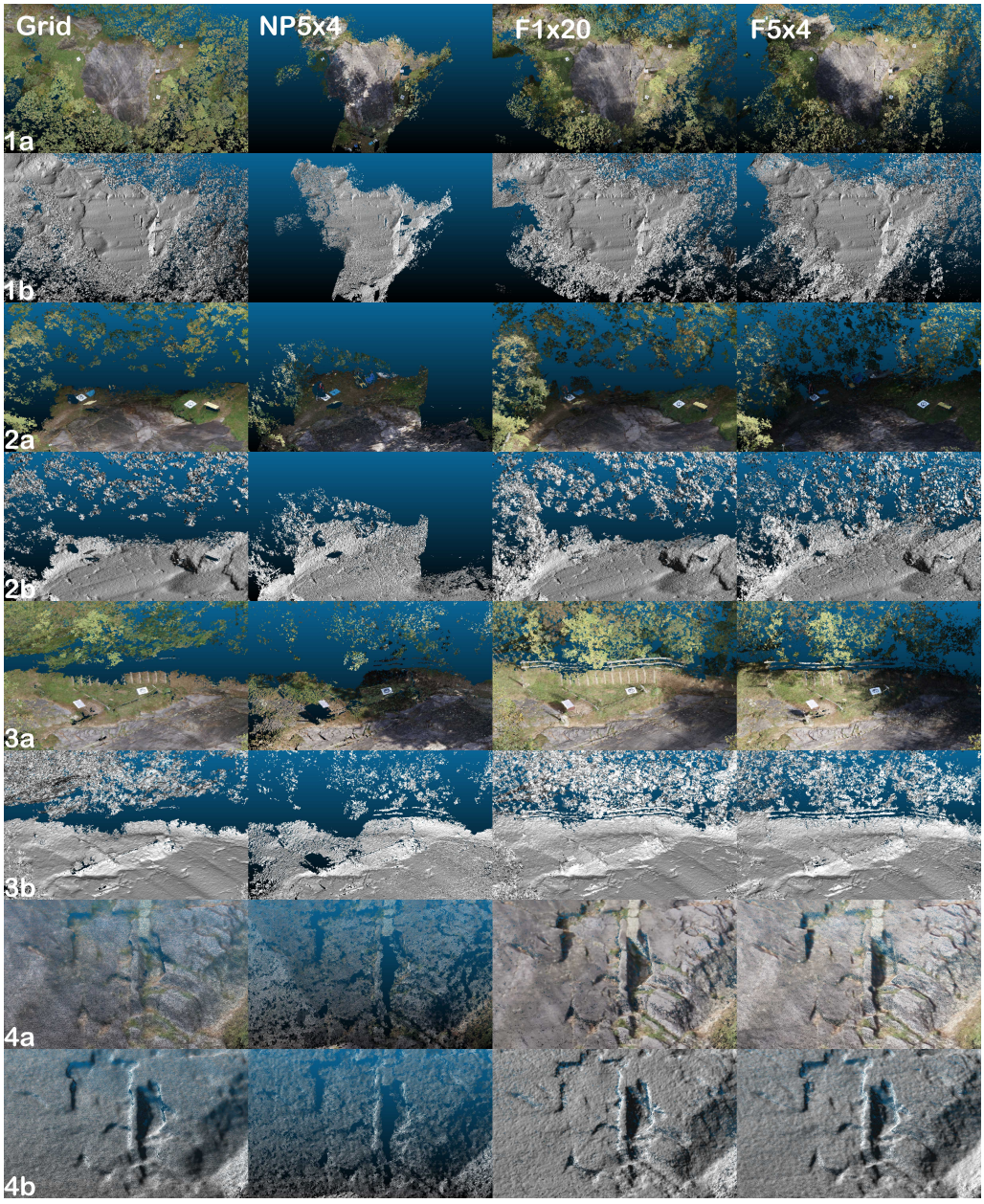


Figure 6.16: We show the resulting reconstructions for the four different approaches. To eliminate the influence of color in the depth perception, we computed the normals of the point cloud and show the resulting point clouds also without color. For all triplet based approaches (NP5x4, F1x20, F5x4) we only show the output of the planned triplets (without initialization images or additional images for registration). Note that the coverage underneath the trees is significantly higher with confidence prediction (F1x20 and F5x4). The bottom row (4a-b) shows a close-up of the rock surface. Note that Grid is overly smooth, while F1x20 and F5x4 have much sharper edges. NP5x4 has a lot of missing parts caused by self-occlusion of the rock surface.

6.4 Multi-View Stereo Prioritization

In this experiment, we move from active vision (where we influence the way images are acquired) to passive vision (where a fixed set of images is given, which cannot be enlarged further). In this experiment, we aim to use the MVS Confidence Prediction to increase the efficiency of the MVS reconstruction process. The main lever for this improved efficiency is the fact that most MVS datasets are acquired in an over-complete way. This means that not all images and view points are required to satisfy the user's requirements.

In our experiments, we use two different environments. The first environment contains cultural heritage sites in the valley of Valcamonica, Italy (the same environment used in the previous experiments). For our experiments, we use our learning-based confidence predictor (Section 5.2), which was specially trained for this kind of environment (Section 6.2.2). For evaluation, we then use a test site which was not included in the training. This allows us to demonstrate the full potential of our approach in the same environment in which is was trained.

The second environment, we use for evaluation, is a suburban setting of single family houses. We will use this scenario only for evaluation and not for training. With this setting, we evaluate the domain generalization properties of our confidence predictor. I.e. how does the predictor react if it is confronted with structures it has never seen before? Does it break down or still behave reasonable?

Using these two datasets, we structure our experiments in three main parts. First, we evaluate only the matching partner selection in Section 6.4.2 and then only the ranking performance in Section 6.4.3. From both of these experiments, we then take the best performing baseline approach and evaluate this combination against our full approach in Section 6.4.4. Finally, we summarize and discuss the outcome of our experiments in Section 6.4.5.

6.4.1 Evaluation Details

For all our experiments, we use the same confidence predictor. This confidence predictor was trained for the environment of Valcamonica on 5000 images of 8 different sites with SURE [108] as MVS algorithm (Section 6.2.2). Note that this dataset does not have any overlap with the two datasets used for evaluation. We selected SURE as our main MVS algorithm as it a widely accepted photogrammetric soft-

ware, which yields high quality results.

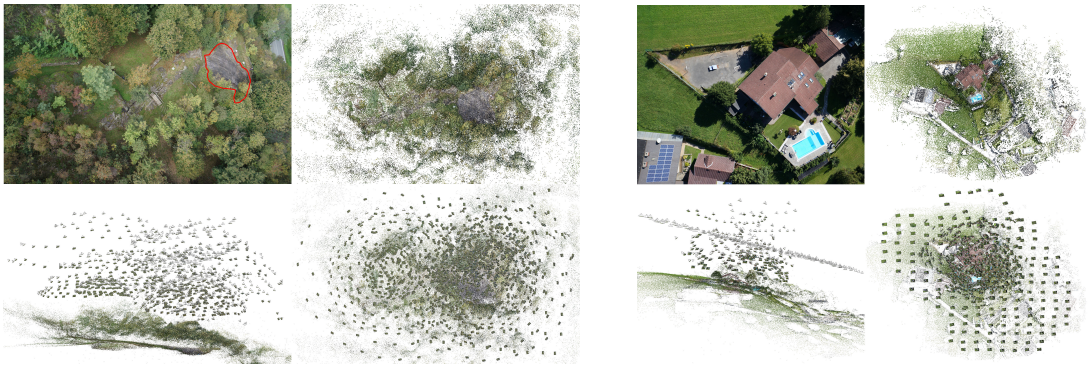
For the matching partner selection, we use the $n = 22$ most connected images, draw $y = 100$ combinations and set the triangle fraction factor to $z = 10$.

For our fulfillment functions, we set the necessary number of cameras to $x = 3$ in line with the default parameter of SURE [108] (i.e. 2 matching partners). We set the desired ground sampling distance g_d and desired accuracy a_d to 1cm and the weighting parameter α to 0.5. For the mesh balancing procedure, this means that 95% of all edges have an edge length longer than 20cm and no edges are longer than 1m.

Valley Dataset. The dataset consists of 1236 registered images of a complex scene in the valley of Valcamonica (Figure 6.17). The images were acquired on 3 consecutive days in regular patterns (i.e. grids and domes) and in semi-structured ways using Automated Image Acquisition (Chapter 4). The grid was acquired with 80% overlap and 8mm GSD. The domes were centered on two separate locations and the GSD was varied approximately from 8mm to 16mm. For the semi-structured Automated Image Acquisition (Chapter 4), the quality parameters were set to 8mm ground sampling distance and accuracy. The images were processed with structure-from-motion pipeline of [110] and geo-referenced using the same fiducial markers as in Section 6.3. This resulted in a sparse point cloud of approximately 480k points, where nearly all cameras contain 3D points with 100+ connections.

House Dataset. The House Dataset contains 485 registered images of a detached house with 3000 m^2 garden. The images were taken in a regular grid and several iterations of our Automated Image Acquisition approach (Chapter 4). The grid was acquired with 80% overlap and 1cm GSD, and for our approach with 1cm for ground resolution and accuracy. Structure-from-Motion and geo-referencing were done in the same way as for the Valley dataset. We use this scenario to evaluate the generalization performance of our approach to new environments. While some scene structures are similar to the Valley environment (grass and trees), most structures have never been seen in training (roofs, house walls, swimming pool, cars, etc.).

Computation Time. We ran all our experiments on the same desktop computer with a Intel Core i7-4771 CPU (with 4 physical and 8 virtual cores), 32GB of RAM



(a) Valley Dataset (b) House Dataset

Figure 6.17: Valley and House Dataset. Top left: One image of the dataset shows an overview of the scene. Top right: A nadir view of the sparse point cloud. Bottom right: The camera poses with the sparse point cloud from the same nadir view point. Bottom left: A side view of the camera poses with the sparse point cloud. For the Valley Dataset, ground truth of the rock formation Seradina 12C (marked in red) was acquired with a laser scanner.

Num Matching Partners	2	3	5	11
Matching Partner Selection [s]				
Valley Dataset	0.7	1.0	1.8	23.8
House Dataset	0.7	1.0	1.8	18.2
Next-Best View Ranking [s]				
Valley Dataset	3.5	3.6	4.0	10.8
House Dataset	2.8	3.1	3.4	8.5
Overall Average [s]				
(Valley+House)/2	3.9	4.4	5.5	30.7
Prioritization Time / Matching Time [%]				
(Valley+House)/2	4.13%	3.11%	2.36%	5.98%

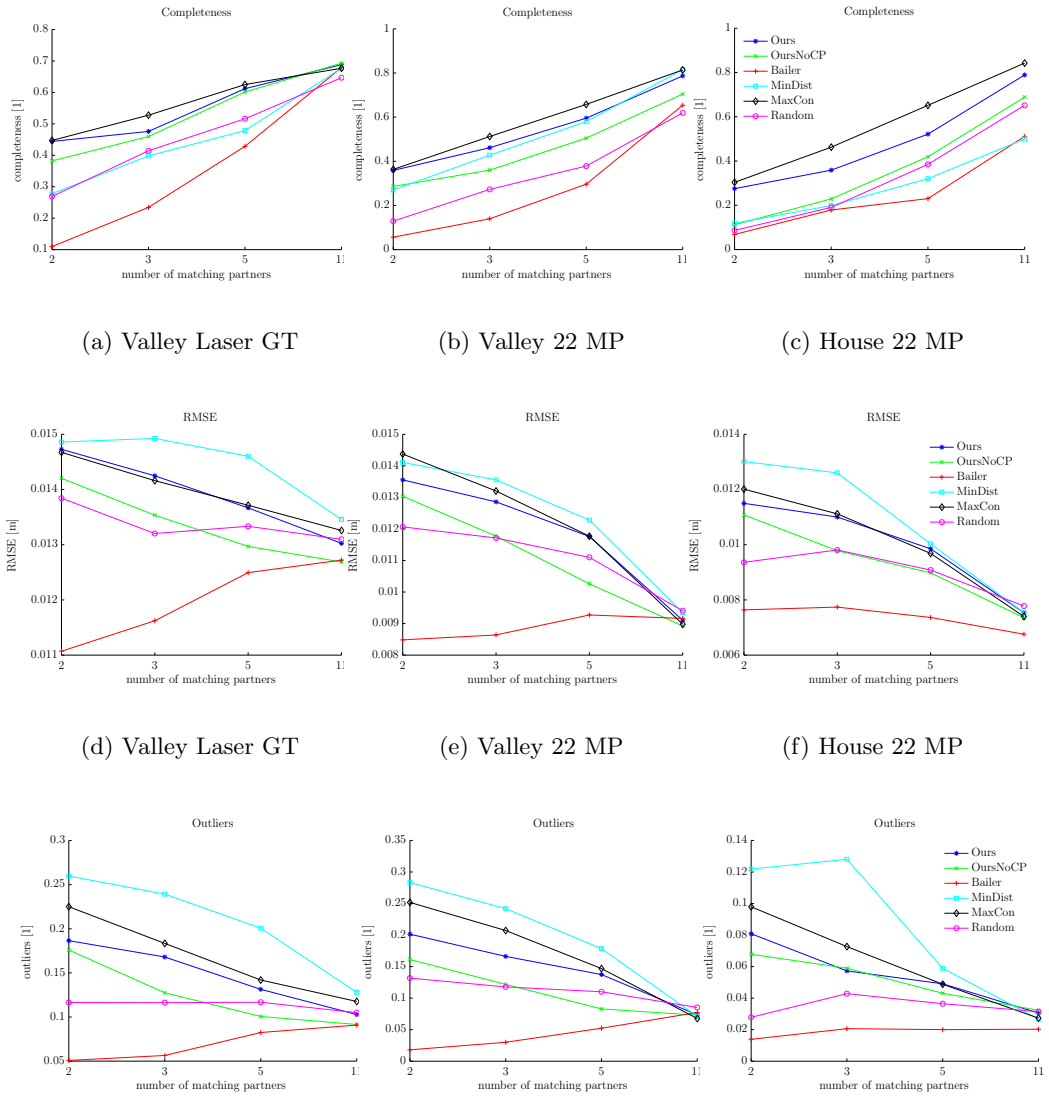
Table 6.6: We show the average time consumption of the two main parts of our approach (i.e. Matching Partner Selection and Next-Best View Ranking) in seconds per key view. All initialization steps are included in the timings. The last two row show the overall average and the relative time consumption between our prioritization method and the matching time of SURE [108].

and GeForce GTX 770. We use the linux version of SURE [108], which only uses the CPU. With this version of SURE the average depth map computation time per matching partner on our datasets is 46.6s. In our approach, the mesh computation and the visibility casting are negligible compared to the more computationally intense parts of our approach (both parts together are finished in less than two minutes for both datasets). Our confidence prediction takes a constant time of 1s per image. In Table 6.6, we show the time consumption for the matching partner selection and the next-best view ranking for each dataset and number of matching partners separately. Note that our prioritization approach is very light weight and only requires 2%-6% of the time the matching procedure of SURE [108] itself. This means that if we reduce the number of required key views to 5%, we effectively reduce the overall run-time to 7%-11%, which speeds up the full MVS reconstruction process by a full order of magnitude.

6.4.2 Selecting the k-best Matching Partners

In this experiment, we evaluate the first subtask of our view cluster prioritization, i.e. the matching partner selection. In this subtask, the algorithm is given a fixed set of key frames and for each key frame the k-best matching partners should be selected. This task is part of many MVS pipelines and has a great impact on the overall performance of the pipeline as k represents the tradeoff between execution speed and quality of the 3D reconstruction. If wrong matching partners are selected, many parts of the scene might either be missing or the estimated 3D points might show a undesirable amount of noise. Thus, we evaluate three factors in this experiment, i.e. completeness, outlier percentage and RMSE (root mean square error). However, this evaluation requires reference data. For the Valley Dataset, we have a traditional ground truth only for a small part of the scene, i.e. a rock formation called Seradina Rock 12C (Figure 6.17). This ground truth was captured with a Riegl VZ-400 Laser Scanner and has an accuracy of 5mm. While we also use this ground truth in our evaluation (denoted as "Laser GT"), it is limited to a mostly flat rock surface.

In order to evaluate the three quality factors also for the other parts of the scene (including the vegetation), we have to use a more unconventional approach. For this purpose, let us first formally define the task of the matching partner selection. In the task of matching partner selection, an algorithm shall select a subset M_k of



(a) Valley Laser GT (b) Valley 22 MP (c) House 22 MP

(d) Valley Laser GT (e) Valley 22 MP (f) House 22 MP

(g) Valley Laser GT (h) Valley 22 MP (i) House 22 MP

Figure 6.18: Matching partner selection experiment on the **Valley** and **House** Datasets. We show completeness, RMSE (Root-Mean-Square Error) and outlier ratio compared to the output with the 22 most connected matching partners (22 MP) and the laser ground truth of a rock formation (Laser GT).

k matching partners (images) for a specific MVS algorithm such that the resulting depth map maximally resembles the depth map produced with the same algorithm and a much larger set of matching partners M_{large} with $M_k \subset M_{large}$ and $|M_k| \ll |M_{large}|$. With this definition, we can use the depth map produced with M_{large} as reference data for assessing the quality parameters. We will further refer to this reference data as "22 MP" as we use the most connected 22 matching partners.

Using these two kinds of reference data, we evaluate the three quality parameters in the following manner. First, we detect outliers with respect to the reference data and the desired accuracy. I.e. we classify all depth estimates which are more than 3 times the desired accuracy (i.e. $> 3 \cdot 1$cm) as outliers. Note that for the traditional ground truth, we additionally exclude occlusions that are more than 24cm from the rock surface from this evaluation. Then we evaluate the completeness and RMSE with respect to the reference data using only the valid measurements (without outliers and occluded values).

From the large set of possible approaches for matching partner selection (see Section 2.5), we select the most relevant approaches for the given algorithm and application of photogrammetric reconstruction. Thus, we evaluate our approach against two photogrammetric standard approaches, which are both implemented by SURE [108], i.e. the k closest images (MinDist), the k most connected images (MaxCon). Note that the default parameters of SURE are MinDist with $k = 5$. For evaluating the connectivity, we count the shared 3D points in the sparse reconstruction. Further, we select k random images as proposed by [35] (Rand) and k images with the carefully hand-crafted approach of [6] (Bailer). Finally, we run our approach once with confidence prediction (Ours) and once without prediction (OursNoCP), i.e. $f_{conf} = 1$. For each key frame, all approaches have the task of selecting $k = 2, 3, 5, 11$ out of the 22 most-connected matching partners.

For computational reasons, we do not computed all possible combination on all images of the datasets, but use a representative subset. For the *22 MP* evaluation, we will use 100 randomly selected key views for evaluation. For the *Laser GT* evaluation (of Rock 12C), we select the 50 views (of the 100) in which Seradina 12C is most prominently visible (i.e. covers the largest area in pixel).

Results. In Figure 6.18, we show the averaged results (over 100 views for *22 MP* and over 50 views for *Laser GT*) in completeness, the root mean square er-

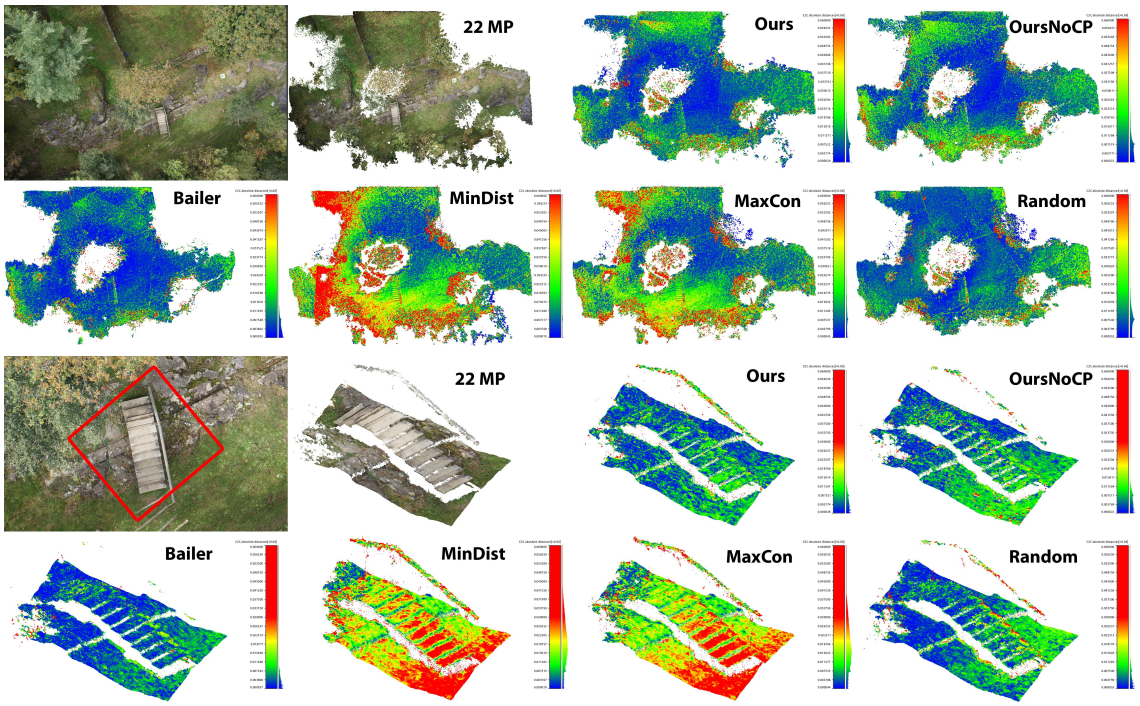


Figure 6.19: Impact of matching partner selection (selecting 5 out of 22) on the **Valley Dataset**. All shown reconstructions are computed with the same image as key view (top left). All evaluated approaches used 5 matching partners. The top two row show the whole point clouds where the color encodes the distance to the reference (22MP); from no error (blue) to *6cm* error (red). The bottom row shows a cutout of the same reconstruction (here red indicates an error larger than *3cm*). Notice that our approach leads to a reconstruction with low error that still preserves fine details such as the hand rail. Other approaches have either a low error but also low completeness (Bailer) or have a high completeness but also a high error (MinDist and MaxCon). Only our approach performs well in both aspects.

ror (RMSE) and the ratio of outliers.

Now let us first compare the results of *Valley Laser GT* and *Valley 22 MP* to establish the relation between traditional ground truth and the reference data with a large number of matching partners. In this regard, the number of outliers seems to be most important. If we compare sub-figures (g) and (h), we can observe that the relative ordering between the approaches stays the same for 2,3 and 5 matching partners. Only for 11 matching partners, this ordering seems to change. From this we conclude that our approximative reference data seems to work reasonably up to 5 matching partners.

Now let us compare the individual approaches on the Valley Dataset. As ex-

pected, for all approaches more matching partners lead to higher completeness. Similarly, more matching partners lead to a lower outlier ratio for most approaches. Only for Bailer, the number of outliers seems to increase (we think this is due to the very low completeness). For all approaches, the RMSE and the outlier ratio seem to be highly correlated.

In general, we can observe a tradeoff between completeness and accuracy (outliers and RMSE). None of the evaluated approaches leads in both completeness and accuracy. Some approaches are rather tuned for completeness (MaxCon and Ours), while others more for accuracy (Bailer and OursNoCP). In contrast to only flat surfaces (Laser GT), we can see a clear gap between the two versions of our approach on the complete scene (22 MP). Further, we can see that our approach seems to be tuned very much towards completeness (very close to MaxCon), but at the same time shows a significantly lower outlier ratio than MaxCon (especially for a low number of matching partners).

If we now take a look at the results of the House Dataset, the relative ordering between MaxCon, Ours and OursNoCP is very similar. Ours stays between the other two approaches (for completeness and accuracy), but exhibits a much higher completeness than OursNoCP.

Overall we conclude that our approach with learning leads to a significantly higher completeness than our approach without learning and at the same time keeps the outlier ratio significantly lower than approaches of similar completeness (especially for a low number of matching partners). This can also be observed in the example shown in Figure 6.19.

6.4.3 Ranking Only

In the previous experiment, we fixed the set of key views and evaluated the matching partner selection. In this experiment, we fix the matching partner selection and evaluate the view cluster ranking performance. For each view, we run our full approach with 5 matching partners. This leads to one view cluster per view.

The task in this experiment is to maximize the *real* fulfillment with as few view clusters as possible. We evaluate the real fulfillment analogue to Equation 5.11, with the difference that all estimated and predicted values are replaced by measurements as follows. For evaluating the coverage, we use the projected depth map together

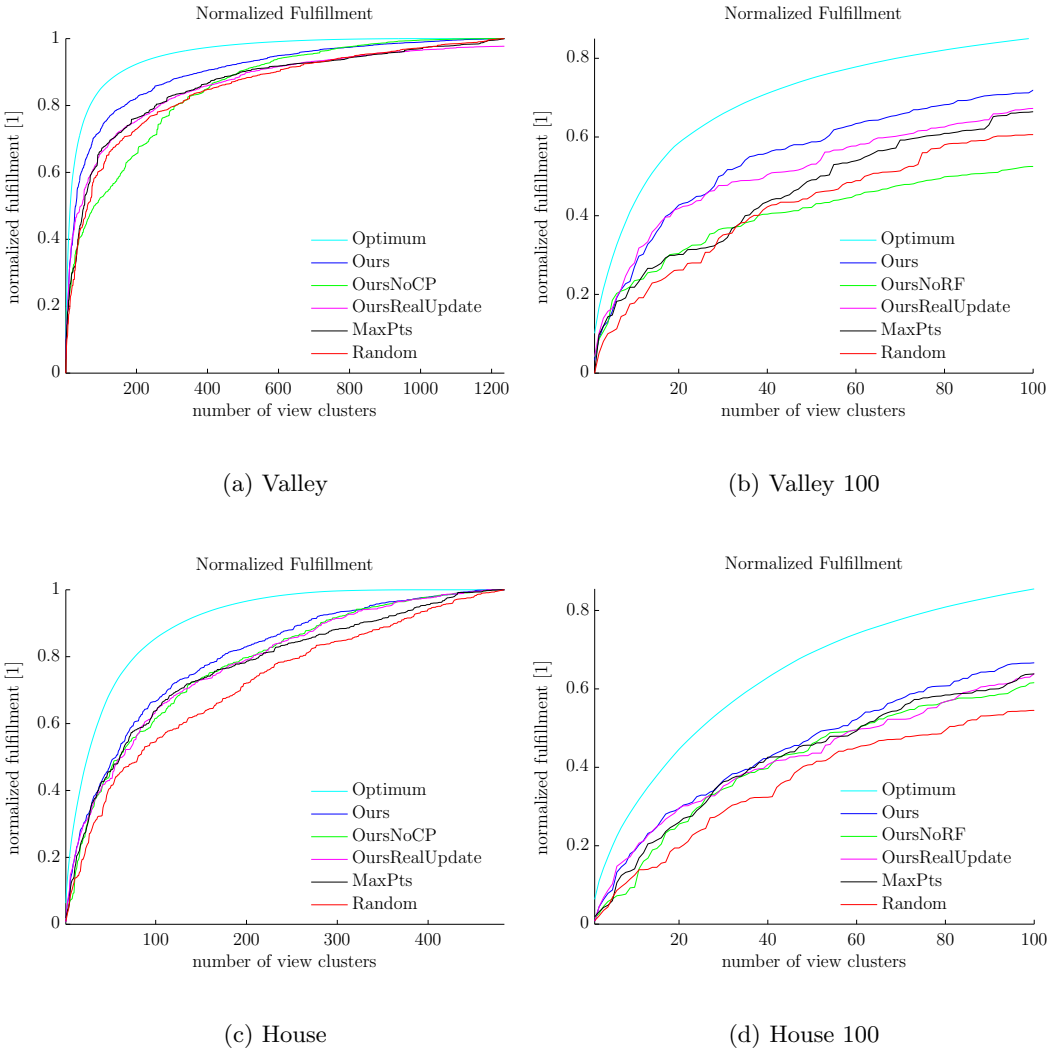
Fulfillment	Opt	Ours	OursNoCP	OursRU	MaxPts	Random
Threshold	Required View Clusters [%]					
10%	0.08	**0.24**	**0.24**	**0.24**	**0.24**	0.32
20%	0.24	0.57	**0.49**	0.57	0.73	1.05
30%	0.49	0.97	1.46	**0.89**	1.62	2.18
40%	0.73	**1.46**	3.07	1.62	2.83	2.99
50%	1.13	**2.43**	6.63	3.72	4.21	5.02
60%	1.78	**4.37**	12.78	5.99	6.07	7.36
70%	3.16	**7.12**	19.17	10.92	10.60	13.03
80%	5.66	**14.00**	25.16	21.60	20.71	25.16
90%	12.70	**30.91**	39.56	42.07	39.97	47.98

Table 6.7: View Cluster Ranking on the Valley Dataset. We show the percentage of view clusters that are required for reaching a certain percentage of the maximal achievable fulfillment. Note that our approach only requires two times the optimal number of view clusters, whereas the second best approach requires between 30-90% more than ours.

with the set of successful matching partners reported by SURE. For judging whether a measurement represents a valid measurement of a triangle, we use the same criterion as in the last experiment. I.e. we treat a measurement as valid if the distance to the corresponding measurement with 22 matching partners is smaller or equal to three times the desired accuracy ($\leq 3 \cdot 1cm$). For evaluating the theoretical uncertainty (Equation 4.3), we use the set of reported matching partners by SURE and check for self occlusions for each triangle. Finally, we replace f_{conf} with the actual coverage of the triangle. This means for a given triangle, we project this triangle into the key view and analyze the percentage of pixels with a valid measurement within the 2D projection of the triangle.

In this experiment, we evaluate our approach with prediction (Ours) against two variants of our approach. The first variant (OursNoCP) does not use the confidence prediction (i.e. f_{conf} is fixed to 1). The second variant (OursRealUpdate) uses the confidence prediction, but instead of also predicting the update of the objective function, it uses the real fulfillment for the update. This variant can be seen as our approach with the MVS approach in the loop.

Additionally to random ranking (Random), we also tried to come up with a good baseline algorithm that only operates on the sparse point cloud. We denote this algorithm as "MaxPts". This algorithm always greedily selects the view cluster of the camera with the highest number of connected 3D points. A 3D point is



(a) Valley

(b) Valley 100

(c) House

(d) House 100

Figure 6.20: Ranking Performance on the **Valley** and **House** Datasets. We show the normalized fulfillment over the number of used view clusters. The left side shows the plot over all view clusters and the right side the first 100 steps of the same plot.

connected to a certain camera if this camera is contained in the feature track of the 3D point. After selecting the camera with the maximum number of points, all these points are removed from the sparse reconstruction. Through this removal operation, the algorithm naturally tries to explore the reconstruction. Finally, we also compare our approach to the optimal greedy solution (Optimum). This algorithm requires all reconstructions to be available in every iteration and uses the real objective function for all its decisions.

Results. In Figure 6.20, we show the normalized fulfillment for all approaches, such that the highest fulfillment value is set to one. The normalized fulfillment is computed as the current fulfillment divided by the maximal achievable fulfillment (i.e. the fulfillment with all 1236 view clusters). In Table 6.7, we also show the percentage of view clusters, which are needed to reach a certain fulfillment level in steps of 10%.

If we analyze the results of the Valley Dataset, we can see the following. First of all, our full approach performs very well. Up to roughly 80% fulfillment our approach roughly requires two times more than the Optimum. Between 30% and 70% (which is one of the most interesting regions for our task), the second best approach requires approximately 70% more view clusters than our approach. What seems really fascinating at first glance is that in this region our approach with prediction (Ours) actual performs better than with the MVS algorithm in the loop (OursRealUpdate). The reason for this astonishing result is that, with the real update, the algorithm does not have any notion of what it has tried in the past. This means, if for some reason the prediction says that there is a chance to reconstruct this object from this view point and the reconstruction actually fails, the algorithm will try the same thing with the view directly next to the last one. However, if we only use the prediction instead of the real update, the algorithm will rather try a completely different view point next. With real updates, the performance of our approach slowly converges to the simple exploration baseline (MaxPts). If we completely remove the prediction from our approach (OursNoCP), the performance degrades very soon below the baseline method. This point of failure marks the moment where the algorithm thinks it has explored most parts of the scene geometrically (just using the mesh), but in reality it is still only at 40% fulfillment. If we take a closer look at Table 6.7, we can see that the first 60% of fulfillment are quite cheap with our approach. For 60% we only require 4.4% of all poses (i.e. 45 view clusters). However, each further added 10% of fulfillment roughly doubles the required number of view clusters.

If we analyze the results of the House Dataset (Figure 6.20), we can see that our approach still performs better than the baseline approaches from 40% upwards.

6.4.4 Full MVS Prioritization

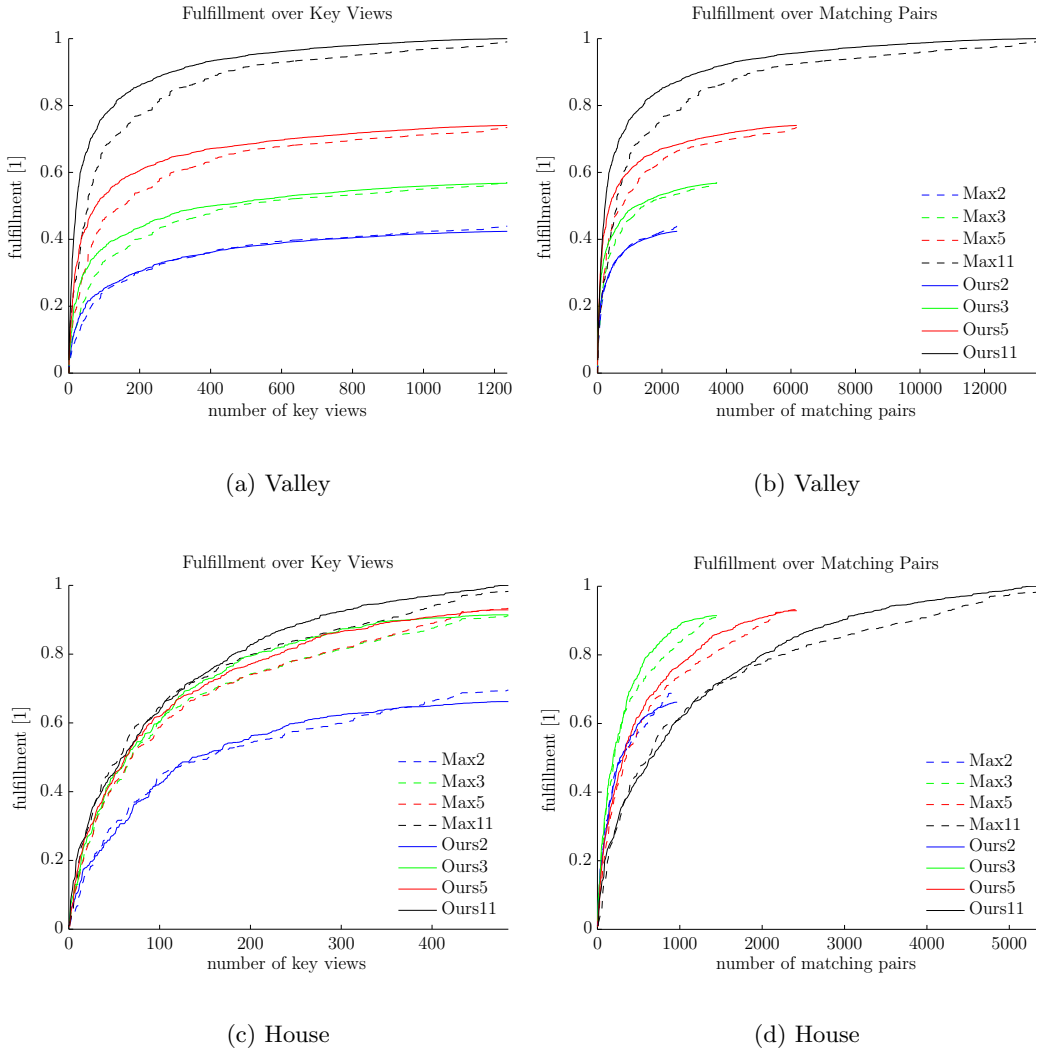
In the previous two experiments, we analyzed each of our two steps separately. In this experiment, we evaluate our full approach against the best combination of the baseline methods. We denote this approach as "Max", as it consists of the matching partner selection based on the maximum connectivity (MaxCon) and the ranking procedure based on the maximum number of visible 3D points (MaxPts). For both approaches, we evaluate the performance for different numbers of matching partners, i.e. $k = 2, 3, 5, 11$.

Results. In Figure 6.21, we show the fulfillment curves for both datasets. Note that the curves are normalized such that maximally achieved fulfillment over all approaches (i.e. Ours11 with all key views) is set to one. On a first glance, we can see that there is a significant difference in the datasets. For the Valley Dataset, there is a significant difference in the maximally reached fulfillment for different numbers of matching partners. E.g. our approach with 3 matching partners is only able to reach half of the fulfillment of the approach with the maximum number of matching partners. For the House Dataset, this gap is significantly smaller and our approach with 3 matching partners is able to reach 90% of the fulfillment. We think that the main reason for this discrepancy is the difficulty of the dataset. While the Valley Dataset is strongly dominated by trees which are exceedingly hard to reconstruct, the House Dataset contains mostly flat structures such as roads, short grass or roofs.

For the Valley Dataset, our approach significantly outperforms right from the start. In Table 6.8, we see that between 30% and 70% fulfillment the baseline approach with 11 matching partners requires 2 to 3 times more matching pairs than our approach for obtaining the same fulfillment level.

If we take a look at the House Dataset (Figure 6.21 and Table 6.9), we can also see a clear gap between our approach and the baseline. However, this clear gap only starts to form between 60% and 70% fulfillment. What seems interesting is that in terms of computational efficiency, three matching partners seem to be the best choice for the House Dataset. Once again, we think that this is the case because the House environment seems to easier than the Valley environment.

If we take a look at the prediction performance (Figure 6.22), we see a significant difference between the training environment (Valley) and the unseen test environ-



(a) Valley

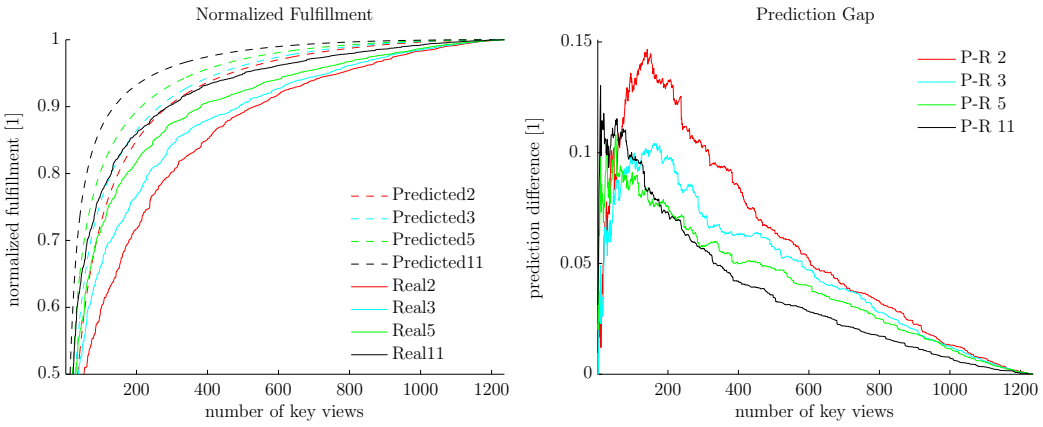
(b) Valley

(c) House

(d) House

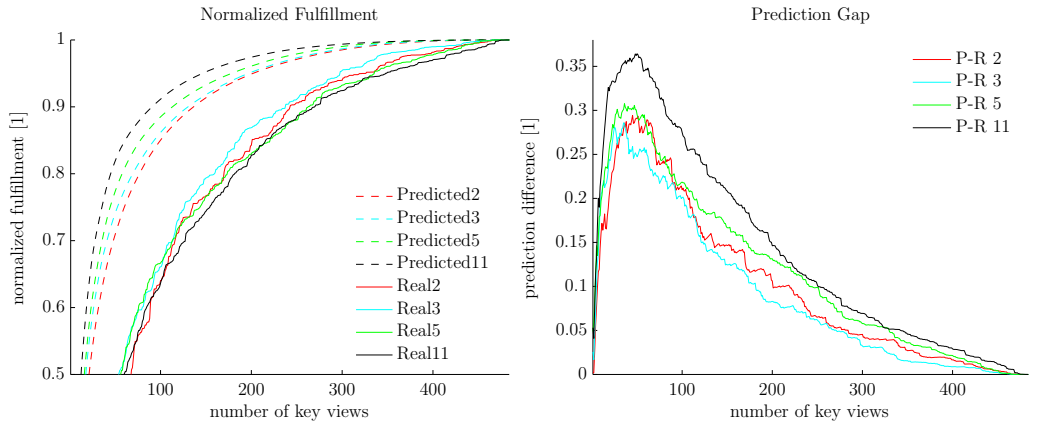
Figure 6.21: Fulfillment development of our full approach on the **Valley Dataset** (top) and **House Dataset** (bottom). We show our full approach (Ours) and the baseline method (Max) for four different numbers of matching partners (i.e. $k = 2, 3, 5, 11$). The left side shows the fulfillment over the number of key views, whereas the right side shows the fulfillment over the number of matching pairs.

ment (House). While there is a significant gap between predicted fulfillment and real fulfillment on the House Dataset, the gap is a lot smaller on the Valley Dataset. In fact, the gap closes below 10% after only 55 key views for 5 matching partners (87 for 11). This means in a known environment, the predicted fulfillment can indeed be used for estimating the actual fulfillment before executing the actual MVS algorithm. With just using a 10% buffer, it is possible to estimate the necessary



(a) Valley

(b) Valley



(c) House

(d) House

Figure 6.22: Prediction performance on the **Valley** and **House** Dataset. On the left side, we show the normalized fulfillment once of predicted with the random forest (Predicted) and once measured with the actual depth maps (Real) for 2,3,5 and 11 matching partners. On the right side, we show the prediction gap as the difference of the normalized fulfillment (i.e. Predicted minus Real) for the **Valley** and **House** Dataset. Note that the prediction gap on the Valley dataset is not very wide (especially for 5 and 11 matching partners). Also note that the relative ordering between different numbers of matching partners is preserved. On the House Dataset (which contains mostly objects that were never seen in training) the gap is significantly wider.

	Max				Ours			
	2	3	5	11	2	3	5	11
Fulfillment	Required Matching Pairs [%]							
10%	0.29	0.26	0.22	0.24	**0.15**	**0.15**	0.18	0.16
20%	0.94	0.82	0.92	0.89	0.71	**0.35**	0.40	0.49
30%	2.88	1.83	1.54	2.18	2.77	1.04	**0.70**	0.81
40%	9.91	4.30	2.57	2.99	10.61	2.91	1.29	**1.13**
50%	-	10.55	5.63	4.45	-	8.94	2.83	**1.78**
60%	-	-	10.70	5.99	-	-	7.02	**2.67**
70%	-	-	30.78	10.03	-	-	22.76	**5.02**
80%	-	-	-	19.26	-	-	-	**10.52**
90%	-	-	-	36.25	-	-	-	**23.46**

Table 6.8: Required number of matching pairs to reach a certain level of fulfillment on the **Valley Dataset** . We compare our full approach (Ours) to the best baseline combination (Max) for a varying number of matching partners. The best values of each row are marked in bold font.

	Max				Ours			
	2	3	5	11	2	3	5	11
Fulfillment	Required Matching Pairs [%]							
10%	0.49	0.39	0.66	1.24	0.30	**0.28**	0.56	0.62
20%	1.05	0.90	1.60	2.48	1.01	**0.79**	1.22	1.86
30%	1.77	1.63	2.72	4.75	2.10	**1.52**	2.25	4.34
40%	3.38	2.54	4.32	7.23	3.34	**2.31**	3.85	7.64
50%	5.90	3.94	6.57	11.36	5.11	**3.61**	5.82	12.40
60%	11.34	5.58	9.86	16.53	9.50	**5.58**	8.64	17.56
70%	-	9.07	15.40	26.03	-	**7.61**	13.71	26.45
80%	-	15.72	26.01	41.94	-	**11.66**	21.88	37.60
90%	-	24.40	39.07	72.52	-	**21.07**	35.41	55.79

Table 6.9: Required number of matching pairs to reach a certain level of fulfillment on the **House Dataset**. We compare our full approach (Ours) to the best baseline combination (Max) for a varying number of matching partners. The best values of each row are marked in bold font.

number of key views for a certain fulfillment level.

Visual Interpretation. In Figure 6.23, we show the early exploration behavior for our approach and the baseline for 44-45 matching pairs. At this point, Ours3 reaches approximately 25% fulfillment, while all others still have a significantly lower fulfillment. The reason for this gap is the following. With 45 matching pairs, Ours3

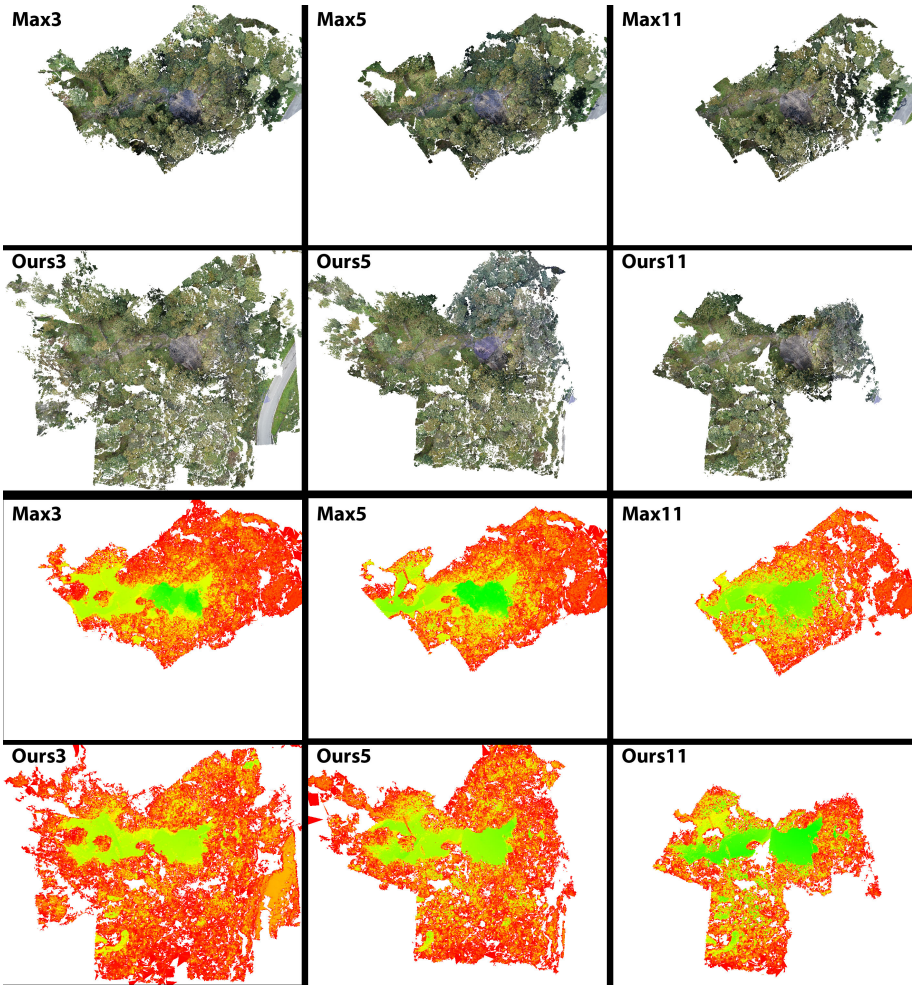


Figure 6.23: Comparison of the early reconstruction behavior (44-45 matching pairs) on the **Valley Dataset**. The two top rows show the point clouds and the two bottom rows the color coded fulfillment on the evaluation mesh. Red is a low (but existing) fulfillment and green a high fulfillment. White means that this part is not covered at all. We compare our approach with the Max baseline for different numbers of matching partners (3,5,11).

and Max3 can select 15 view clusters. While Max3 stays there where most images were taken, Ours3 explores nearly all parts of the scene. At this point, Max11 and Ours11 could only select 4 view clusters. While both approaches can clearly not cover the whole scene with 4 views, we can see a difference in how the views are selected. While Max11 selects highly connected views from a far distance, Ours11 already prefers closer views which are able to reach the desired resolution and accuracy on smooth surfaces (rock).

In Figure 6.24, we show the exploration behavior for our approach and the base-

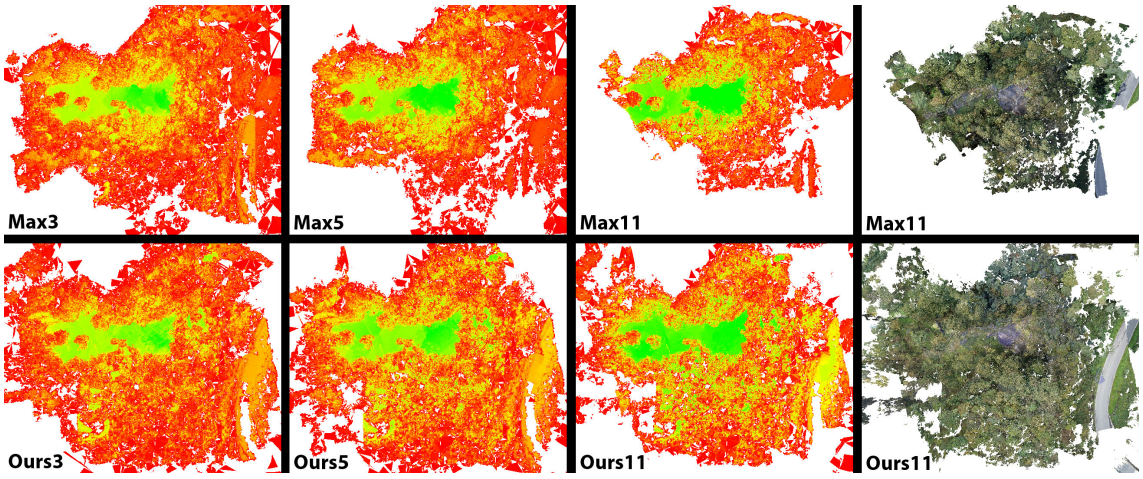


Figure 6.24: Comparison of the exploration behavior for 220 matching pairs. The right column shows point clouds, while the others show the fulfillment on the evaluation mesh. The colors of the mesh range from close to zero fulfillment (red) over yellow to complete fulfillment (green) and white means zero fulfillment.

line for 219-220 matching pairs. While all variants of our approach explored the scene well, a lot of the outer parts of the scene are missing for the baseline. Note that the compactness of the representation with a higher number of matching partners is much better than for lower numbers. With 11 matching partners the point cloud size is approximately 150 million points, while it is already 500 million points for 5 matching partners and exceeds 1 billion points for 3 matching partner. We conclude that in this environment, fewer views with more matching partners lead to a better fulfillment at a higher compactness for SURE.

In Figure 6.25, we show the advantage of our mesh-based approach compared to only using sparse points. To make the difference more obvious, we only marked the main house as the region of interest; i.e. the task now is to reconstruct only the house and the other parts of the scene do not matter. For this experiment, we also removed all sparse points outside the the region of interest from the point cloud.

If we take a look at the results, we can see that with our approach the facades of the houses are included in the model at a very early stage, while they are completely missing for the Max approach. The reason for this is that Ours starts to select oblique views at a very early stage as it gives facades and roof equal priority. In contrast, the Max approach is led by the number of sparse points and only a very low number of sparse points lie on the facades. This biases the Max approach strongly

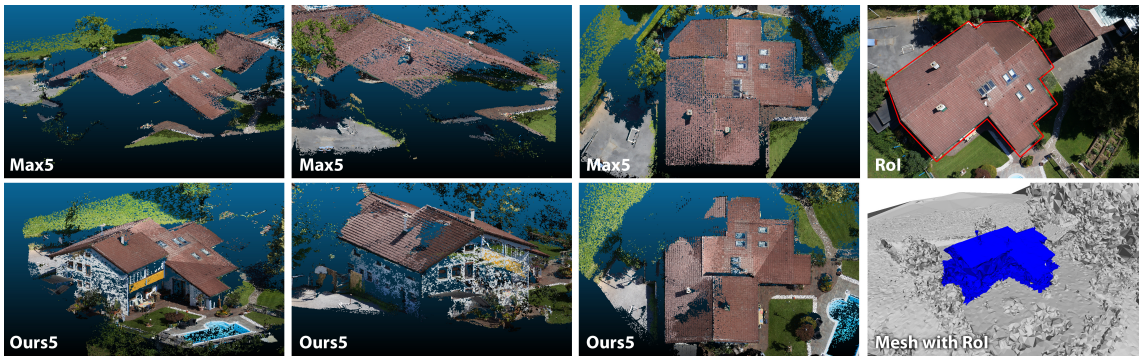


Figure 6.25: Benefit of mesh (Ours) versus point cloud only (Max). In this experiment, we only marked the house itself as region of interest (RoI) (top right); the corresponding interesting triangles are shown in blue (bottom right). We show the MVS output after 8 view clusters for five matching partners. The Max approach tends to only select nadir views, as most of the sparse features lie on the roof of the house. In contrast, our approach gives the facades and the roof equal priority. Thus, our approach also selects oblique views so that the facades are also represented in the 3D reconstruction.

towards selecting nadir views, which nicely cover the well textured roof.

6.4.5 Discussion

In all our experiments, we have shown that the MVS Confidence Prediction allows us to consistently outperform the exact same approach without prediction. We see this as proof that our prioritization approach was able to use the accumulated knowledge of the confidence predictor to significantly improve the performance. In this sense, we think that our approach is a great benefit to reoccurring photogrammetric tasks in difficult environments.

Potential Benefits. On the Valley Dataset (i.e. the dataset which was captured in the same domain as the training data), we were able to achieve significantly better results for the view cluster prioritization than any other baseline. For the same level of fulfillment our approach requires 2 to 3 times less key views than the best baseline. Further, we demonstrated that the real fulfillment only lacks approximately 10% behind the predicted fulfillment. This means that the fulfillment prediction is a reasonable approach for approximating the necessary number of key views to get a good coverage of the scene. E.g. if we aim for 70% fulfillment with 11 matching partners, we can use our approach to predict a fulfillment of 80%. In this experiment,

this method would lead to 69.3% fulfillment with only 60 key views (which is already highly complete – see Video [87]). If we analyze the computation time for this example in relation to using each view as key view (as done by conventional MVS), we end up with a speed up factor of 9.2 (from $n/(s + n \cdot r)$ with number selected of key views n, number selected of key views s and the ratio r of prioritization time over MVS time). If we are only interested in the saved memory, we even achieve a memory reduction factor of 20.1 (430M points vs. 8650M points). This means that our approach has a large potential to save computational time and memory with a very small overhead.

Generalization Performance. Our experiments on the House dataset demonstrated that our approach still performs reasonable even if confronted with a scene that contains mostly objects that were never seen in training. Our approach still has some benefit over the baseline methods at higher levels of fulfillment, however, the margin is significantly less than for the Valley dataset. On the House dataset, we reach 70% of the achievable fulfillment with 7.6% of the available matching pairs, while the corresponding baseline requires 9.1%. However, if we compare our the actual run-time to the best baseline, we see that our method does not lead to a run-time improvement for the same fulfillment level due to the higher computational overhead. In this sense, we can conclude that our method will not break down in an unknown environment, but the main benefit over non-learning approaches will be significantly diminished.

Chapter 7

Conclusion and Outlook

In this thesis, we introduce the concept of MVS Confidence Prediction. In contrast to traditional stereo confidence measures, this new concept aims to predict the chances of a successful 3D reconstruction *before* the actual MVS algorithm is executed. This concept gives us active control over the images presented to the MVS algorithm and thus grants us the opportunity to actively avoid suboptimal or redundant image constellations.

One of the special attributes of our method is that we are able to obtain the necessary training data in a fully automated manner – without ground truth or manual interaction. The key to our training scheme is that we evaluate the consistency between 3D reconstructions obtained with the same algorithm from significantly different view points. This scheme allows us to detect systematic problems of an MVS algorithm (with respect to certain scene structures) and allows us to use such cases for training a MVS Confidence Predictor. This means that the only input our method requires is an over-complete set of images that observes a scene from different view points, which makes our training scheme highly scalable and flexible to the application scenario.

We demonstrate the usefulness of MVS Confidence Prediction in two applications. The first application is the image acquisition itself. Here, we use the MVS Confidence to actively adapt the acquisition pattern in a way that the resulting image constellations are perfectly suited for a specific MVS algorithm. In a challenging environment, this allows us to achieve a highly complete and accurate 3D reconstruction with a significantly lower number of images compared to traditional acquisition methods.

In the second application, we take a fixed (but over-complete) set of images and use the MVS Confidence to rank the given images depending on their impact on the 3D reconstruction quality. In our experiments, our approach is thus able to reach a certain level of quality fulfillment (in terms of completeness with respect to a desired accuracy and ground resolution) with up to three times less key views than the best baseline approach. In contrast to other approaches, our approach has the distinct advantage that it provides the opportunity to predict the relative fulfillment level in complex scenes. This means that the user can decide whether doubling the amount of computation time and memory consumption for reaching 80% fulfillment instead of 70% (which is hardly visible – see Video [87]) is necessary or not. On the topic of domain generalization, we have shown that our approach still performs reasonably well in an environment that has never been seen in training, however, the advantage over non-learning based approaches is significantly diminished. In this sense, we see the main application area of our approach in reoccurring photogrammetric tasks in challenging environments, where our approach can be applied with little additional cost as the whole learning procedure is fully automated.

Possible Improvements. In this work, we showed that self-supervised learning can be a great benefit for improving the efficiency of MVS, however, our approach has its limits. In our approach, the prediction is done with a very light-weight random forest that only has an inference window of 27×27 pixels. With this restriction our approach can only make decisions based on a small local window and does not have any context information. In this regard, we think that convolutional neural networks (CNNs) have the potential to significantly improve the prediction performance.

For our approach for automated image acquisition, we see room for improvement in the number of planned images. In the related experiments, we only generate three images out of each surrogate camera, because we used the exact same setup in training. However, our later experiments on MVS Prioritization have shown that using only 3 images is suboptimal for all scenarios. Thus, we recommend to generate a larger number of images out of each surrogate camera; at least 4 cameras for easy scenes and even more for difficult scenes.

This brings us to another point, where we see room for improvement. I.e. our approach currently does not select the number of required matching partners depending on the observed scene. In our approach, this decision has to be still made

by the user. While we think that this can be incorporated in our framework, we did not yet have time to explore this option. In this regard, the main difficulty will be to find the right kind and amount of regularization to tradeoff computing time, amount of resulting data and fulfillment.

Outlook. In this thesis, we focused on improving the efficiency and quality of the MVS reconstruction process without changing the algorithm itself. This makes it possible to integrate our approach in a standard MVS pipeline and boost its efficiency and performance with very little effort. In the future, however, we also see the potential that ideas similar to ours are used within the MVS algorithm itself to adapt its internal parameters to the presented scene structure.

In the short term, it could replace hand-crafted heuristics such as the triangulation angle prior of Schönberger et al. [115] for improving the performance. However, if we look a little bit farther in the future, we see a lot of potential in transferring ideas of two-view confidence measures to MVS. As demonstrated in our experiments, two-view confidence measures are exceedingly more powerful than our MVS Confidence Predictor. The main reason for the significant performance difference is that our MVS Confidence Predictor only uses very little information (i.e. a single color image) for the confidence estimation, while two-view confidence measures have a lot more information available for the confidence estimation (i.e. a rectified image pair, two depth maps and/or the cost volume). Thus, we think that also for MVS a very powerful confidence measure could be learned, if the necessary input data is provided. Such a confidence measure – within the reconstruction procedure – could detect which images or parameter sets (e.g. the matching window size or similarity measure) will more likely lead to a successful reconstruction and smoothly adapt the algorithm's behavior to the presented scene structure.

If we once again look farther into the future and take CNNs for single view depth estimation (e.g. [39]) and 3D CNNs (e.g. [66]) into account, we think that measurements (in the traditional sense) and learned hallucinations (supported by previously seen data) will be continuously merging with each other. While future dense 3D reconstruction algorithms will also have to fit some data term, we see a lot of freedom in how regions between data-supported measurements will be interpolated and occluded parts will be completed. For certain applications (e.g. content generation for virtual reality), many scene parts will be only mildly supported by measurements

and then fully automatically completed with learned shapes. The challenges will be how to achieve this consistently and in a scalable fashion, for which the ideas presented in this thesis could be a valuable starting point.

Appendix A

Acronyms

List of Acronyms

AUSC	Area Under the Sparsification Curve
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DSM	Digital Surface Model
FOV	Field Of View
GCP	Ground Control Point
GPS	Global Positioning System
GPU	Graphics Processing Unit
GSD	Ground Sampling Distance
MVS	Multi-View Stereo
NBV	Next-Best-View
RAM	Random-Access Memory
RANSAC	Random Sample Consensus
RF	Random Decision Forest
RGB	Red, Green, Blue (color space)
RMSE	Root Mean Square Error
ROC	Receiver Operator Characteristic
SfM	Structure from Motion
SGM	Semi-Global Matching
SIFT	Scale Invariant Feature Transform

STD	STandard Deviation
STF	Semantic Texton Forest
UAV	Unmanned Aerial Vehicle

Appendix B

List of Publications

In chronological order, we list all peer-reviewed publications of Christian Mostegel and note how they are related to this thesis.

B.1 2014

B.1.1 Active Monocular Localization: Towards Autonomous Monocular Exploration for Multirotor MAVs

Christian Mostegel, Andreas Wendel and Horst Bischof.
In: IEEE International Conference on Robotics and Automation (ICRA).
June 2014, Hong Kong, China.
(full paper - oral presentation)
[Best Student Paper Award - Finalist]
(5 finalists of 1001 accepted papers (48% general acceptance rate)).

Abstract. The main contribution of this paper is to bridge the gap between passive monocular SLAM and autonomous robotic systems. While passive monocular SLAM strives to reconstruct the scene and determine the current camera pose for any given camera motion, not every camera motion is equally suited for these tasks. In this work we propose methods to evaluate the quality of camera motions with respect to the generation of new useful map points and localization maintenance. In our experiments, we demonstrate the effectiveness of our measures using a low-cost quadrocopter. The proposed system only requires a single passive camera as exte-

roceptive sensor. Due to its explorative nature, the system achieves autonomous way-point navigation in challenging, unknown, GPS-denied environments.

Related Chapter(s): -

B.2 2015

B.2.1 Graz Griffins Solution to the European Robotics Challenges 2014

Jesus Pestana Puerta, Rudolf Prettenthaler, Thomas Holzmann, Daniel Muschick, **Christian Mostegel**, Friedrich Fraundorfer and Horst Bischof.
In: Austrian Robotics Workshop (ARW).
May 2015, Klagenfurt, Austria.
(short paper - oral presentation)

Abstract. An important focus of current research in the field of Micro Aerial Vehicles (MAVs) is to increase the safety of their operation in general unstructured environments. An example of a real-world application is visual inspection of industry infrastructure, which can be greatly facilitated by autonomous multicopters. Currently, active research is pursued to improve real-time vision-based localization and navigation algorithms. In this context, the goal of Challenge 3 of the EuRoC 20144 Simulation Contest was a fair comparison of algorithms in a realistic setup which also respected the computational restrictions onboard an MAV. The evaluation separated the problem of autonomous navigation into four tasks: visual-inertial localization, visual-inertial mapping, control and state estimation, and trajectory planning. This EuRoC challenge attracted the participation of 21 important European institutions. This paper describes the solution of our team, the Graz Griffins, to all tasks of the challenge and presents the achieved results.

Related Chapter(s): -

B.2.2 Performance Evaluation of Vision-Based Algorithms for MAVs

Thomas Holzmann, Rudolf Prettenthaler, Jesus Pestana Puerta, Daniel Muschick, **Christian Mostegel**, Friedrich Fraundorfer, Horst Bischof and Gottfried Graber.

In: Workshop of the Austrian Association for Pattern Recognition (OAGM).
May 2015, Salzburg, Austria.
(full paper - oral presentation)

Abstract. An important focus of current research in the field of Micro Aerial Vehicles (MAVs) is to increase the safety of their operation in general unstructured environments. Especially indoors, where GPS cannot be used for localization, reliable algorithms for localization and mapping of the environment are necessary in order to keep an MAV airborne safely. In this paper, we compare vision-based real-time capable methods for localization and mapping and point out their strengths and weaknesses. Additionally, we describe algorithms for state estimation, control and navigation, which use the localization and mapping results of our vision-based algorithms as input.

Related Chapter(s): -

B.3 2016

B.3.1 Using Self-Contradiction to Learn Confidence Measures in Stereo Vision

Christian Mostegel, Markus Rumpler, Friedrich Fraundorfer, Horst Bischof.
In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
June 2016, Las Vegas, USA.
(full paper - poster presentation)

Abstract. Learned confidence measures gain increasing importance for outlier removal and quality improvement in stereo vision. However, acquiring the necessary training data is typically a tedious and time consuming task that involves manual interaction, active sensing devices and/or synthetic scenes. To overcome this problem, we propose a new, flexible, and scalable way for generating training data that only requires a set of stereo images as input. The key idea of our approach is to use different view points for reasoning about contradictions and consistencies between multiple depth maps generated with the same stereo algorithm. This enables us to generate a huge amount of training data in a fully automated manner. Among

other experiments, we demonstrate the potential of our approach by boosting the performance of three learned confidence measures on the KITTI2012 dataset by simply training them on a vast amount of automatically generated training data rather than a limited amount of laser ground truth data.

Related Chapter(s): 3, 6.1

B.3.2 UAV-based Autonomous Image Acquisition with Multi-View Stereo Quality Assurance by Confidence Prediction

Christian Mostegel, Markus Rumpler, Friedrich Fraundorfer, Horst Bischof.
In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.
June 2016, Las Vegas, USA.
(full paper - spotlight and poster presentation)

Abstract. In this paper we present an autonomous system for acquiring close-range high-resolution images that maximize the quality of a later-on 3D reconstruction with respect to coverage, ground resolution and 3D uncertainty. In contrast to previous work, our system uses the already acquired images to predict the confidence in the output of a dense multi-view stereo approach without executing it. This confidence encodes the likelihood of a successful reconstruction with respect to the observed scene and potential camera constellations. Our prediction module runs in real-time and can be trained without any externally recorded ground truth. We use the confidence prediction for on-site quality assurance and for planning further views that are tailored for a specific multi-view stereo approach with respect to the given scene. We demonstrate the capabilities of our approach with an autonomous Unmanned Aerial Vehicle (UAV) in a challenging outdoor scenario.

Related Chapter(s): 3, 4, 6.2, 6.3

B.3.3 The 3D-PITOTI Project with a Focus on Multi-Scale 3D Reconstruction using Semi-Autonomous UAVs

Christian Mostegel, Georg Poier, Christian Reinbacher, Manuel Hofer, Friedrich Fraundorfer, Horst Bischof, Thomas Höll, Gert Holler, Axel Pinz.
In: 1st OAGM-ARW Joint Workshop (Vision Meets Robotics).
May 2016, Wels, Austria.
(extended abstract - oral presentation)

Abstract. In this talk, we showcase our outcome of the ambitious 3D-PITOTI project, which involves a multi-disciplinary team of over 30 scientists from across Europe. The project focuses on the 3D aspect of recording, storing, processing and visualizing prehistoric rock art in the UNESCO World Heritage site in Valcamonica, Italy. The rock art was pecked into open-air rock formations thousands of years ago and has an inherent 3D nature. After a project overview, we present the results of the Graz University of Technologyâs contributions in 3D acquisition and processing with a focus on our novel autonomous UAV system. We elaborate the challenges of 3D reconstruction across vastly different scales, from a valley wide reconstruction down to individual peckings on the rock surface. Within this context, we first present a novel 3D scanning device with sub-millimeter accuracy. Aside from correctly scaled 3D information, the scanning device also provides the surface radiometry without the need for artificial shrouding. Additionally, we point out one application for which this highly accurate 3D data has shown to be crucial: The interactive segmentation of the individually pecked figures.

Finally, we present a novel autonomous UAV system for acquiring high-resolution images at a few meters distance. The system optimizes scene coverage, ground resolution and 3D uncertainty, while ensuring that the acquired images are suitable for a specific dense offline 3D reconstruction algorithm. There are three main aspects that set this system apart from others. First, the system operates completely on-site without the need for a prior 3D model of the scene. Second, the system iteratively refines a surface mesh, predicts the fulfillment of requirements and can thus correct for initially wrong geometry estimates and imperfect plan execution. Third, the system uses the already acquired 2D images to predict the chances of a successful reconstruction with a specific offline 3D densification algorithm depending on the

observed scene and potential camera constellations. We demonstrate the capabilities of our system in the challenging environment of the prehistoric rock art sites and then register the individual reconstructions of all scales in one consistent coordinate frame.

Related Chapter(s): 3, 4, 6.2, 6.3

B.4 2017

B.4.1 Evaluations on multi-scale camera networks for precise and geo-accurate reconstructions from aerial and terrestrial images with user guidance

Markus Rumpler, Alexander Tscharf, **Christian Mostegel**, Shreyansh Daftry, Christof Hoppe, Rudolf Prettenthaler, Friedrich Fraundorfer, Gerhard Mayer, Horst Bischof.
In: Computer Vision and Image Understanding.
April 2017, Volume 157.
(journal paper)

Abstract. During the last decades photogrammetric computer vision systems have been well established in scientific and commercial applications. Recent developments in image-based 3D reconstruction systems have resulted in an easy way of creating realistic, visually appealing and accurate 3D models. We present a fully automated processing pipeline for metric and geo-accurate 3D reconstructions of complex geometries supported by an online feedback method for user guidance during image acquisition. Our approach is suited for seamlessly matching and integrating images with different scales, from different view points (aerial and terrestrial), and with different cameras into one single reconstruction. We evaluate our approach based on different datasets for applications in mining, archaeology and urban environments and thus demonstrate the flexibility and high accuracy of our approach. Our evaluation includes accuracy related analyses investigating camera self-calibration, georegistration and camera network configuration.

Related Chapter(s): -

B.4.2 Der Einsatz unbemannter Flugsysteme zur Charakterisierung von gesprengtem Haufwerk

Alexander Tscharf, **Christian Mostegel**, Andreas Gaich, Gerhard Mayer, Friedrich Fraundorfer, Horst Bischof.
In: 18. Geokinematischer Tag des Institutes für Markscheidewesen und Geodäsie am 10. und 12. Mai 2017 in Freiberg.
May 2017, Freiberg, Germany.
(full paper - oral presentation)

Abstract. The fragmentation and the shape of the muck pile are the two major outcomes of open pit mine and quarry blasts. Fast information about the muck pile properties will help to improve the production scheduling and furthermore this information could be used to optimize the blasting patterns of future production blasts. The combined use of unmanned aerial vehicles (UAVs) and modern machine learning and computer vision systems offers a new way of acquiring spatial data to determine on-site fragment size distribution, while at the same time enabling integration into common work flows and mitigating the weaknesses of ground-based systems with special regard to completeness and representativeness. In the present paper, we will discuss the relevant related work, present the planned path for system development and give examples of first work.

Related Chapter(s): -

B.4.3 Scalable Surface Reconstruction from Point Clouds with Extreme Scale and Density Diversity

Christian Mostegel, Rudolf Prettenthaler, Friedrich Fraundorfer, Horst Bischof.
In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
July 2017, Honolulu, USA.
(full paper - poster presentation)

Abstract. In this paper we present a scalable approach for robustly computing a 3D surface mesh from multi-scale multi-view stereo point clouds that can handle extreme jumps of point density (in our experiments three orders of magnitude). The

backbone of our approach is a combination of octree data partitioning, local Delaunay tetrahedralization and graph cut optimization. Graph cut optimization is used twice, once to extract surface hypotheses from local Delaunay tetrahedralizations and once to merge overlapping surface hypotheses even when the local tetrahedralizations do not share the same topology. This formulation allows us to obtain a constant memory consumption per sub-problem while at the same time retaining the density independent interpolation properties of the Delaunay-based optimization. On multiple public datasets, we demonstrate that our approach is highly competitive with the state-of-the-art in terms of accuracy, completeness and outlier resilience. Further, we demonstrate the multi-scale potential of our approach by processing a newly recorded dataset with 2 billion points and a point density variation of more than four orders of magnitude - requiring less than 9GB of RAM per process.

Related Chapter(s): -

B.5 2018

B.5.1 Prioritized Multi-View Stereo Depth Map Generation using Confidence Prediction.

Christian Mostegel, Friedrich Fraundorfer, Horst Bischof.
In: ISPRS Journal of Photogrammetry and Remote Sensing.
(journal paper - **in revision**)

Abstract. In this work, we propose a novel approach to prioritize the depth map computation of multi-view stereo (MVS) to obtain compact 3D point clouds of high quality and completeness at low computational cost. Our prioritization approach operates before the MVS algorithm is executed and consists of two steps. In the first step, we aim to find a good set of matching partners for each view. In the second step, we rank the resulting view clusters (i.e. key views with matching partners) according to their impact on the fulfillment of desired quality parameters such as completeness, ground resolution and accuracy. Additional to geometric analysis, we use a novel machine learning technique for training a confidence predictor. The purpose of this

confidence predictor is to estimate the chances of a successful depth reconstruction for each pixel in each image for one specific MVS algorithm based on the RGB images and the image constellation. The underlying machine learning technique does not require any ground truth or manually labeled data for training, but instead adapts ideas from depth map fusion for providing a supervision signal. The trained confidence predictor allows us to evaluate the quality of image constellations and their potential impact to the resulting 3D reconstruction and thus builds a solid foundation for our prioritization approach. In our experiments, we are thus able to reach more than 70% of the maximal reachable quality fulfillment using only 5% of the available images as key views. For evaluating our approach within and across different domains, we use two completely different scenarios, i.e. cultural heritage preservation and reconstruction of single family houses.

Related Chapter(s): 5, 6.4

Bibliography

[1] Aanæs, H., Jensen, R. R., Vogiatzis, G., Tola, E., and Dahl, A. B. (2016). Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*.

[2] Ahmadabadian, A., Robson, S., Boehm, J., and Shortis, M. (2014a). Stereo-imaging network design for precise and dense 3D reconstruction. *The Photogrammetric Record*, 29(147).

[3] Ahmadabadian, A. H., Sargeant, B., Erfani, T., Robson, S., Shortis, M., Hess, M., and Boehm, J. (2014b). Towards fully automatic reliable 3d acquisition: From designing imaging network to a complete and accurate point cloud. *Robotics and Autonomous Systems*, 62(8).

[4] Alexis, K., Papachristos, C., Siegwart, R., and Tzes, A. (2015). Uniform coverage structural inspection path-planning for micro aerial vehicles. In *Intelligent Control (ISIC), 2015 IEEE International Symposium on*.

[5] Almadhoun, R., Taha, T., Seneviratne, L., Dias, J., and Cai, G. (2016). A survey on inspecting structures using robotic systems. *International Journal of Advanced Robotic Systems*, 13(6).

[6] Bailer, C., Finckh, M., and Lensch, H. P. A. (2012). Scale robust multi view stereo. In *European Conference on Computer Vision (ECCV)*, Berlin, Heidelberg. Springer Berlin Heidelberg.

[7] Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B. (2009). Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3).

[8] Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. (2008). Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110.

[9] Bircher, A., Alexis, K., Burri, M., Oettershagen, P., Omari, S., Mantel, T., and Siegwart, R. (2015). Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics. In *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*.

[10] Bircher, A., Kamel, M., Alexis, K., Oleynikova, H., and Siegwart, R. (2016). Receding horizon path planning for 3d exploration and surface inspection. *Autonomous Robots*.

[11] Bleyer, M., Rhemann, C., and Rother, C. (2011). Patchmatch stereo-stereo matching with slanted support windows. In *British Machine Vision Conference (BMVC)*, volume 11.

[12] Brown, M., Hua, G., and Winder, S. (2011). Discriminative learning of local image descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1).

[13] Butler, D. J., Wulff, J., Stanley, G. B., and Black, M. J. (2012). A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision (ECCV)*.

[14] Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). BRIEF: binary robust independent elementary features. In *European Conference on Computer Vision (ECCV)*, Hersonissos, Greece.

[15] Chang, A. X., Funkhouser, T. A., Guibas, L. J., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F. (2015). Shapenet: An information-rich 3d model repository. *arXiv*, abs/1512.03012.

[16] Civera, J., Davison, A. J., and Montiel, J. M. M. (2007). Inverse depth to depth conversion for monocular slam. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*.

[17] Collins, R. T. (1996). A space-sweep approach to true multi-image matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[18] Crandall, D., Owens, A., Snavely, N., and Huttenlocher, D. (2011). Discrete-continuous optimization for large-scale structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[19] Dornhege, C., Kleiner, A., and Rolling, A. (2013). Coverage search in 3D. In *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*.

[20] Egnal, G., Mintz, M., and Wildes, R. P. (2004). A stereo confidence metric using single view imagery with comparison to five alternative approaches. *Image and Vision Computing*, 22(12). Proceedings from the 15th International Conference on Vision Interface.

[21] Egnal, G. and Wildes, R. (2002). Detecting binocular half-occlusions: empirical comparisons of five approaches. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(8).

[22] Engel, J., Sturm, J., and Cremers, D. (2013). Semi-dense visual odometry for a monocular camera. In *IEEE International Conference on Computer Vision (ICCV)*.

[23] Englot, B. and Hover, F. (2011). Planning complex inspection tasks using redundant roadmaps. In *Int. Symp. Robotics Research*.

[24] European Machine Vision Association (2016). EMVA Standard 1288: Standard for Characterization of Image Sensors and Cameras. EMVA 1288 3.1, European Machine Vision Association.

[25] Everingham, M., van Gool, L., Williams, C., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2).

[26] Fan, X., Zhang, L., Brown, B., and Rusinkiewicz, S. (2016). Automated view and path planning for scalable multi-object 3d scanning. *ACM Trans. Graph.*, 35(6).

[27] Feige, U. (1998). A threshold of ln n for approximating set cover. *J. ACM*, 45(4).

[28] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6).

[29] Forster, C., Pizzoli, M., and Scaramuzza, D. (2014). Appearance-based active, monocular, dense reconstruction for micro aerial vehicles. In *Proceedings of Robotics: Science and Systems*, Berkeley, USA.

[30] Fuhrmann, S., Langguth, F., and Goesele, M. (2014). Mve-a multiview reconstruction environment. In *Proceedings of the Eurographics Workshop on Graphics and Cultural Heritage (GCH)*, volume 6.

[31] Furukawa, Y., Curless, B., Seitz, S. M., and Szeliski, R. (2010). Towards internet-scale multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[32] Furukawa, Y. and Ponce, J. (2010). Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8).

[33] Galceran, E., Campos, R., Palomeras, N., Ribas, D., Carreras, M., and Ridao, P. (2015). Coverage path planning with real-time replanning and surface reconstruction for inspection of three-dimensional underwater structures using autonomous underwater vehicles. *Journal of Field Robotics*, 32(7).

[34] Galceran, E. and Carreras, M. (2013). A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12).

[35] Galliani, S., Lasinger, K., and Schindler, K. (2015). Massively parallel multiview stereopsis by surface normal diffusion. In *IEEE International Conference on Computer Vision (ICCV)*.

[36] Gallup, D., Frahm, J. M., Mordohai, P., and Pollefeys, M. (2008). Variable baseline/resolution stereo. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*.

[37] Gallup, D., Frahm, J. M., Mordohai, P., Yang, Q., and Pollefeys, M. (2007). Real-time plane-sweeping stereo with multiple sweeping directions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[38] Gao, X.-S., Hou, X.-R., Tang, J., and Cheng, H.-F. (2003). Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8).

[39] Garg, R., B.G., V. K., Carneiro, G., and Reid, I. (2016). Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision (ECCV)*, Cham. Springer International Publishing.

[40] Garland, M. and Heckbert, P. S. (1997). Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

[41] Gehrig, S. and Scharwachter, T. (2011). A real-time multi-cue framework for determining optical flow confidence. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*.

[42] Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[43] Geiger, A., Ziegler, J., and Stiller, C. (2011). StereoScan: Dense 3D Reconstruction in Real-time. In *Intelligent Vehicles Symposium (IV)*.

[44] Godard, C., Mac Aodha, O., and Brostow, G. J. (2017). Unsupervised monocular depth estimation with left-right consistency. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[45] Goesele, M., Snavely, N., Curless, B., Hoppe, H., and Seitz, S. M. (2007). Multi-view stereo for community photo collections. In *IEEE International Conference on Computer Vision (ICCV)*.

[46] Haeusler, R., Nair, R., and Kondermann, D. (2013). Ensemble learning for confidence measures in stereo vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[47] Han, X., Leung, T., Jia, Y., Sukthankar, R., and Berg, A. C. (2015). Matchnet: Unifying feature and metric learning for patch-based matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[48] Häne, C., Heng, L., Lee, G. H., Sizov, A., and Pollefeys, M. (2014). Real-time direct dense matching on fisheye images using plane-sweeping stereo. In *Proceedings of the 2014 2nd International Conference on 3D Vision (3DV)*, 3DV '14, Washington, DC, USA. IEEE Computer Society.

[49] Haner, S. and Heyden, A. (2011). Optimal view path planning for visual SLAM. In *Proceedings of the 17th Scandinavian Conference on Image Analysis*, SCIA'11, Ystad, Sweden. Springer-Verlag.

[50] Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition.

[51] Hartmann, W., Galliani, S., Havlena, M., Van Gool, L., and Schindler, K. (2017). Learned multi-patch similarity. In *IEEE International Conference on Computer Vision (ICCV)*.

[52] Hartmann, W., Havlena, M., and Schindler, K. (2014). Predicting matchability. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[53] Hiep, V. H., Keriven, R., Labatut, P., and Pons, J. P. (2009). Towards high-resolution large-scale multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[54] Hirschmüller, H. (2008). Stereo processing by semiglobal matching and mutual information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2).

[55] Hollinger, G. A., Englot, B., Hover, F., Mitra, U., and Sukhatme, G. S. (2012). Uncertainty-Driven View Planning for Underwater Inspection. In *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*.

[56] Hoppe, C., Klopschitz, M., Donoser, M., and Bischof, H. (2013). Incremental surface extraction from sparse structure-from-motion point clouds. In *British Machine Vision Conference (BMVC)*.

[57] Hoppe, C., Klopschitz, M., Rumpler, M., Wendel, A., Kluckner, S., Bischof, H., and Reitmayr, G. (2012a). Online feedback for structure-from-motion image acquisition. In *British Machine Vision Conference (BMVC)*, volume 2.

[58] Hoppe, C., Wendel, A., Zollmann, S., Pirker, K., Irschara, A., Bischof, H., and Kluckner, S. (2012b). Photogrammetric camera network design for micro aerial vehicles. In *Proceedings of the Computer Vison Winterworkshop*, Mala Nedelja, Slovenia.

[59] Hornung, A., Zeng, B., and Kobbelt, L. (2008). Image selection for improved multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[60] Hu, X. and Mordohai, P. (2012). A quantitative evaluation of confidence measures for stereo vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11).

[61] Irschara, A., Zach, C., Frahm, J. M., and Bischof, H. (2009). From structure-from-motion point clouds to fast location recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[62] Jancosek, M. and Pajdla, T. (2011). Multi-view reconstruction preserving weakly-supported surfaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[63] Jancosek, M. and Pajdla, T. (2012). Hallucination-free multi-view stereo. In *Trends and Topics in Computer Vision*, Berlin, Heidelberg. Springer Berlin Heidelberg.

[64] Jing, W., Polden, J., Lin, W., and Shimada, K. (2016a). Sampling-based view planning for 3d visual coverage task with unmanned aerial vehicle. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[65] Jing, W., Polden, J., Tao, P. Y., Lin, W., and Shimada, K. (2016b). View planning for 3d shape reconstruction of buildings with unmanned aerial vehicles. In *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*.

[66] Kar, A., Häne, C., and Malik, J. (2017). Learning a multi-view stereo machine. In *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc.

[67] Karaszewski, M., Adamczyk, M., and Sitnik, R. (2016). Assessment of next-best-view algorithms performance with various 3d scanners and manipulator. {*ISPRS*} *Journal of Photogrammetry and Remote Sensing*, 119.

[68] Kim, S., Min, D., Kim, S., and Sohn, K. (2017). Feature augmentation for learning confidence measure in stereo matching. *IEEE Transactions on Image Processing*, 26(12).

[69] Krause, A. and Golovin, D. (2014). Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*. Cambridge University Press.

[70] Kuhn, A., Hirschmüller, H., Scharstein, D., and Mayer, H. (2016). A tv prior for high-quality scalable multi-view stereo reconstruction. *International Journal of Computer Vision*.

[71] Labatut, P., Pons, J.-P., and Keriven, R. (2007). Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts. In *IEEE International Conference on Computer Vision (ICCV)*.

[72] Ladicky, L., Sturgess, P., Russell, C., Sengupta, S., Bastanlar, Y., Clocksin, W., and Torr, P. (2012). Joint optimization for object class segmentation and dense stereo reconstruction. *International Journal of Computer Vision*, 100(2).

[73] Lau, B., Sprunk, C., and Burgard, W. (2013). Efficient grid-based spatial representations for robot navigation in dynamic environments. *Robotics and Autonomous Systems*, 61(10). Selected Papers from the 5th European Conference on Mobile Robots (ECMR 2011).

[74] Laussedat, A. (1899). *La métrophotographie*. Enseignement supérieur de la photographie. Gauthier-Villars.

[75] Long, G., Kneip, L., Alvarez, J. M., Li, H., Zhang, X., and Yu, Q. (2016). *Learning Image Matching by Simply Watching Video*. Springer International Publishing, Cham.

[76] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60.

[77] Mac Aodha, O., Humayun, A., Pollefeys, M., and Brostow, G. (2013). Learning a confidence measure for optical flow. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(5).

[78] Martin, R. A., Rojas, I., Franke, K., and Hedengren, J. D. (2016). Evolutionary view planning for optimized uav terrain modeling in a simulated environment. *Remote Sensing*, 8(1):26.

[79] Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., and Brox, T. (2016). A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[80] Mendez, O., Hadfield, S., Pugeault, N., and Bowden, R. (2016). Next-best stereo: extending next best view optimisation for collaborative sensors. In *British Machine Vision Conference (BMVC)*.

[81] Menze, M. and Geiger, A. (2015). Object scene flow for autonomous vehicles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[82] Merrell, P., Akbarzadeh, A., Wang, L., Mordohai, P., Frahm, J.-M., Yang, R., Nister, D., and Pollefeys, M. (2007). Real-time visibility-based fusion of depth maps. In *IEEE International Conference on Computer Vision (ICCV)*.

[83] Meydenbauer, A. (1892). *Das photographische Aufnehmen zu wissenschaftlichen Zwecken, insbesondere das Messbild-Verfahren*. Untes Verlags-Anstalt, Berlin.

[84] Morris, D. D. (2001). *Gauge freedoms and uncertainty modeling for 3D computer vision*. PhD thesis, Carnegie Mellon University.

[85] Mostegel, C. (2018a). Video of resulting reconstruction of our full image acquisition approach with prediction (F5x4) versus our approach without prediction (NP5x4). `https://www.dropbox.com/s/y4tk08mrrekev52/Image_acquisition_F5x4_vs_NP5x4.avi?dl=1`.

[86] Mostegel, C. (2018b). Video of user interaction with image acquisition software. `https://www.dropbox.com/s/jbl6hu4whomb5h4/Image_acquisition_GUI.avi?dl=1`.

[87] Mostegel, C. (2018c). Video showing raw point clouds for different fulfillment levels with our MVS prioritization approach. `https:`

//www.dropbox.com/s/q5ylfd2xbwezg9i/Prioritized_MVS_Valley_Ours11_Raw_PointClouds.mp4?dl=1.

[88] Mostegel, C., Fraundorfer, F., and Bischof, H. (2018). Prioritized multi-view stereo depth map generation using confidence prediction. *ISPRS Journal of Photogrammetry and Remote Sensing*, [In Revision].

[89] Mostegel, C., Prettenthaler, R., Fraundorfer, F., and Bischof, H. (2017). Scalable surface reconstruction from point clouds with extreme scale and density diversity. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[90] Mostegel, C. and Rumpler, M. (2012). Robust Surface Reconstruction from Noisy Point Clouds using Graph Cuts. Technical report, Graz University of Technology, Institute of Computer Graphics and Vision. https://www.tugraz.at/institute/icg/Media/mostegel_2012_techreport.

[91] Mostegel, C., Rumpler, M., Fraundorfer, F., and Bischof, H. (2016a). Uav-based autonomous image acquisition with multi-view stereo quality assurance by confidence prediction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[92] Mostegel, C., Rumpler, M., Fraundorfer, F., and Bischof, H. (2016b). Using self-contradiction to learn confidence measures in stereo vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[93] Mostegel, C., Wendel, A., and Bischof, H. (2014). Active Monocular Localization: Towards Autonomous Monocular Exploration for Multirotor MAVs . In *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*.

[94] Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions – I. *Mathematical Programming*, 14(1).

[95] Nister, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6).

[96] O'rourke, J. (1987). *Art gallery theorems and algorithms*, volume 57. Oxford University Press Oxford.

[97] Park, M.-G. and Yoon, K.-J. (2015). Leveraging stereo matching with learning-based confidence measures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[98] Pathak, D., Girshick, R., Dollar, P., Darrell, T., and Hariharan, B. (2017). Learning features by watching objects move. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[99] Peris, M., Maki, A., Martull, S., Ohkawa, Y., and Fukui, K. (2012). Towards a simulation driven stereo vision system. In *International Conference on Pattern Recognition*.

[100] Philbin, J., Isard, M., Sivic, J., and Zisserman, A. (2010). Descriptor learning for efficient retrieval. In *European Conference on Computer Vision (ECCV)*, volume 6313 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg.

[101] Poggi, M. and Mattoccia, S. (2016a). Learning a general-purpose confidence measure based on o(1) features and a smarter aggregation strategy for semi global matching. In *2016 Fourth International Conference on 3D Vision (3DV)*.

[102] Poggi, M. and Mattoccia, S. (2016b). Learning from scratch a confidence measure. In *British Machine Vision Conference (BMVC)*.

[103] Poggi, M. and Mattoccia, S. (2017). Learning to predict stereo reliability enforcing local consistency of confidence maps. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[104] Poggi, M., Tosi, F., and Mattoccia, S. (2017). Quantitative evaluation of confidence measures in a machine learning world. In *IEEE International Conference on Computer Vision (ICCV)*.

[105] Ranftl, R., Vineet, V., Chen, Q., and Koltun, V. (2016). Dense monocular depth estimation in complex dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[106] Roberts, M., Dey, D., Truong, A., Sinha, S., Shah, S., Kapoor, A., Hanrahan, P., and Joshi, N. (2017). Submodular trajectory optimization for aerial 3d scanning. In *IEEE International Conference on Computer Vision (ICCV)*.

[107] Roggeman, H., Marzat, J., Bernard-Brunei, A., and Besnerais, G. L. (2016). Prediction of the scene quality for stereo vision-based autonomous navigation. *IFAC-PapersOnLine*, 49(15).

[108] Rothermel, M., Wenzel, K., Fritsch, D., and Haala, N. (2012). SURE: Photogrammetric Surface Reconstruction from Imagery. In *Proceedings LC3D Workshop*.

[109] Rumpler, M., Daftry, S., Tscharf, A., Prettenthaler, R., Hoppe, C., Mayer, G., and Bischof, H. (2014). Automated end-to-end workflow for precise and geo-accurate reconstructions using fiducial markers. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(3).

[110] Rumpler, M., Tscharf, A., Mostegel, C., Daftry, S., Hoppe, C., Prettenthaler, R., Fraundorfer, F., Mayer, G., and Bischof, H. (2017). Evaluations on multi-scale camera networks for precise and geo-accurate reconstructions from aerial and terrestrial images with user guidance. *Computer Vision and Image Understanding*, 157. Large-Scale 3D Modeling of Urban Indoor or Outdoor Scenes from Images and Range Scans.

[111] Sadat, S. A., Chutskoff, K., Jungic, D., Wawerla, J., and Vaughan, R. (2014). Feature-rich path planning for robust navigation of mavs with mono-slam. In *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*.

[112] Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nešić, N., Wang, X., and Westling, P. (2014). High-resolution stereo datasets with subpixel-accurate ground truth. In *Pattern Recognition*. Springer International Publishing, Cham.

[113] Schmid, K., Hirschmüller, H., Dömel, A., Grixa, I., Suppa, M., and Hirzinger, G. (2012). View Planning for Multi-View Stereo 3D Reconstruction Using an Autonomous Multicopter. *Journal of Intelligent and Robotic Systems*, 65.

[114] Schönberger, J. L. and Frahm, J.-M. (2016). Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[115] Schönberger, J. L., Zheng, E., Frahm, J.-M., and Pollefeys, M. (2016). Pixel-

wise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, Cham. Springer International Publishing.

[116] Schops, T., Schönberger, J. L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., and Geiger, A. (2017). A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[117] Schöps, T., Schönberger, J. L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., and Geiger, A. (2017). A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[118] Schulter, S., Leistner, C., Wohlhart, P., Roth, P. M., and Bischof, H. (2014). Accurate object detection with joint classification-regression random forests. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[119] Schulter, S., Wohlhart, P., Leistner, C., Saffari, A., Roth, P. M., and Bischof, H. (2013). Alternating decision forests. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[120] Seki, A. and Pollefeys, M. (2016). Patch based confidence prediction for dense disparity map. In *British Machine Vision Conference (BMVC)*. BMVA Press.

[121] Shen, S. (2013). Accurate multiple view 3d reconstruction using patch-based stereo for large-scale scenes. *IEEE Transactions on Image Processing*, 22(5).

[122] Shotton, J., Johnson, M., and Cipolla, R. (2008). Semantic texton forests for image categorization and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[123] Shotton, J., Winn, J., Rother, C., and Criminisi, A. (2009). Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1).

[124] Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Learning local feature descriptors using convex optimisation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(8).

[125] Spyropoulos, A., Komodakis, N., and Mordohai, P. (2014). Learning to detect ground control points for improving the accuracy of stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[126] Strecha, C., von Hansen, W., Van Gool, L., Fua, P., and Thoennessen, U. (2008). On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[127] Sweeney, C., Sattler, T., Höllerer, T., Turk, M., and Pollefeys, M. (2015). Optimizing the viewing graph for structure-from-motion. In *IEEE International Conference on Computer Vision (ICCV)*.

[128] Tang, Z., von Gioi, R. G., Monasse, P., and Morel, J. M. (2017). A precision analysis of camera distortion models. *IEEE Transactions on Image Processing*, 26(6).

[129] Tarbox, G. H. and Gottschlich, S. N. (1995). Planning for complete sensor coverage in inspection. *Computer Vision and Image Understanding*, 61:84–111.

[130] Tonioni, A., Poggi, M., Mattoccia, S., and Di Stefano, L. (2017). Unsupervised adaptation for deep stereo. In *IEEE International Conference on Computer Vision (ICCV)*.

[131] Torralba, A., Russell, B., and Yuen, J. (2010). Labelme: Online image annotation and applications. *Proceedings of the IEEE*, 98(8).

[132] Tosi, F., Poggi, M., Tonioni, A., Di Stefano, L., and Mattoccia, S. (2017). Learning confidence measures in the wild. In *British Machine Vision Conference (BMVC)*, volume 2.

[133] Trummer, M., Munkelt, C., and Denzler, J. (2010). Online Next-Best-View Planning for Accuracy Optimization Using an Extended E-Criterion. In *International Conference on Pattern Recognition*.

[134] Trzcinski, T., Christoudias, M., Lepetit, V., and Fua, P. (2012). Learning image descriptors with the boosting-trick. In *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc.

[135] Vu, H.-H., Labatut, P., Pons, J.-P., and Keriven, R. (2012). High accuracy and visibility-consistent dense multiview stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5).

[136] Weinzaepfel, P., Revaud, J., Harchaoui, Z., and Schmid, C. (2013). Deepflow: Large displacement optical flow with deep matching. In *IEEE International Conference on Computer Vision (ICCV)*.

[137] Wenzel, K., Rothermel, M., Fritsch, D., and Haala, N. (2013). Image acquisition and model selection for multi-view stereo. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-5/W1:251–258.

[138] Wu, P., Liu, Y., Ye, M., Li, J., and Du, S. (2017). Fast and adaptive 3d reconstruction with extensively high completeness. *IEEE Transactions on Multimedia*, 19(2).

[139] Yamaguchi, K., McAllester, D., and Urtasun, R. (2014). Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *European Conference on Computer Vision (ECCV)*, Cham. Springer International Publishing.

[140] Yu, J. J., Harley, A. W., and Derpanis, K. G. (2016). Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *European Conference on Computer Vision (ECCV)*, Cham. Springer International Publishing.

[141] Zabih, R. and Woodfill, J. (1994). Non-parametric local transforms for computing visual correspondence. In *European Conference on Computer Vision (ECCV)*. Springer.

[142] Zheng, E., Dunn, E., Jojic, V., and Frahm, J. M. (2014). Patchmatch based joint view selection and depthmap estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[143] Zhu, Z., Stamatopoulos, C., and Fraser, C. S. (2015). Accurate and occlusion-robust multi-view stereo. {*ISPRS*} *Journal of Photogrammetry and Remote Sensing*, 109.