Dissertation
Submitted to the Combined Faculties
for the Natural Sciences and for Mathematics
of the Ruperto Carola University of Heidelberg, Germany
for the Degree of Doctor of Natural Sciences

Put forward by
M.Sc. Katrin Honauer
born in Kehl, Germany

Date of oral examination: _____

Performance Metrics and Test Data Generation for Depth Estimation Algorithms

Advisers:
Prof. Dr. Bernd Jähne
Prof. Dr. Lena Maier-Hein

Zusammenfassung

Die vorliegende Arbeit befasst sich mit der Entwicklung von Performanzmetriken und Testdaten zur Evaluierung von Tiefenschätzungsalgorithmen.

Stereo- und Lichtfeld-Algorithmen erhalten strukturierte Kamerabilder als Eingabe und rekonstruieren daraus eine Tiefenkarte der abgebildeten Szene. Mittlerweile finden derartige Algorithmen vielfältige Anwendung in der Praxis, zum Beispiel in der industriellen Inspektion und in der Filmindustrie. Darüber hinaus werden sie zunehmend auch in sicherheitskritischen Bereichen wie Fahrerassistenzsystemen und computerassistierter Chirurgie eingesetzt. Trotz dieser steigenden praktischen Relevanz werden Tiefenschätzungsalgorithmen noch immer mit simplen Fehlermaßen und auf kleinen akademischen Datensätzen evaluiert. Für die Auswahl und Weiterentwicklung geeigneter und sicherer Algorithmen ist jedoch ein genaues Verständnis der jeweiligen Stärken und Schwächen essentiell.

In dieser Arbeit zeige ich auf, dass für eine sorgfältige und belastbare Performanzanalyse neben durchschnittlichen Pixelfehlern der Algorithmenergebnisse auch die spezifischen Anforderungen der Anwendung sowie die Eigenschaften der zur Verfügung stehenden Testdaten einbezogen werden müssen. Ich definiere Metriken zur spezifischen Quantifizierung von Tiefenschätzungen an kontinuierlichen Oberflächen, Tiefenkanten und feinen Strukturen. Diese Geometrien sind besonders relevant für viele Anwendungen und herausfordernd für Algorithmen. Im Gegensatz zu gängigen Metriken berücksichtigen die vorgeschlagenen Metriken, dass Pixel innerhalb eines Bildes weder räumlich voneinander unabhängig, noch einheitlich anspruchsvoll, oder gleichermaßen relevant sind.

Neben Performanzmetriken spielen Testdaten eine große Rolle bei der Evaluierung. Diese sind in der Regel nur in begrenzter Menge, Qualität, und Diversität verfügbar. Ich zeige Strategien auf, wie Defizite der zur Verfügung stehenden Testdaten durch spezifische Metriken, zusätzliche Annotation sowie durch stratifizierte Testdaten kompensiert werden können.

Anhand von systematischen Testfällen, einer Anwenderstudie sowie einer ausführlichen Fallstudie weise ich nach, dass die vorgestellten Metriken, Testdaten und Visualisierungen eine aussagekräftige, quantitative Analyse der Stärken und Schwächen verschiedener Algorithmen ermöglichen. Im Gegensatz zu existierenden Auswertungsverfahren können anwendungsspezifische Prioritäten berücksichtigt und die jeweils besten Algorithmen identifiziert werden.

Abstract

This thesis investigates performance metrics and test datasets used for the evaluation of depth estimation algorithms.

Stereo and light field algorithms take structured camera images as input to reconstruct a depth map of the depicted scene. Such depth estimation algorithms are employed in a multitude of practical applications such as industrial inspection and the movie industry. Recently, they have also been used for safety-relevant applications such as driver assistance and computer assisted surgery. Despite this increasing practical relevance, depth estimation algorithms are still evaluated with simple error measures and on small academic datasets. To develop and select suitable and safe algorithms, it is essential to gain a thorough understanding of their respective strengths and weaknesses.

In this thesis, I demonstrate that computing average pixel errors of depth estimation algorithms is not sufficient for a thorough and reliable performance analysis. The analysis must also take into account the specific requirements of the given applications as well as the characteristics of the available test data.

I propose metrics to explicitly quantify depth estimation results at continuous surfaces, depth discontinuities, and fine structures. These geometric entities are particularly relevant for many applications and challenging for algorithms. In contrast to prevalent metrics, the proposed metrics take into account that pixels are neither spatially independent within an image nor uniformly challenging nor equally relevant.

Apart from performance metrics, test datasets play an important role for evaluation. Their availability is typically limited in quantity, quality, and diversity. I show how test data deficiencies can be overcome by using specific metrics, additional annotations, and stratified test data.

Using systematic test cases, a user study, and a comprehensive case study, I demonstrate that the proposed metrics, test datasets, and visualizations allow for a meaningful quantitative analysis of the strengths and weaknesses of different algorithms. In contrast to existing evaluation methodologies, application-specific priorities can be taken into account to identify the most suitable algorithms.

Acknowledgements

First and foremost, I want to thank my supervisor Prof. Dr. Bernd Jähne for the opportunity to complete my PhD at the Faculty for Mathematics and Computer Science at Heidelberg University. He provided a stimulating and collaborative research environment at the Heidelberg Collaboratory for Image Processing (HCI). I am grateful for his academic support, the scientific freedom to pursue my research, and his great interest and expertise in real-world applications.

Special thanks go to Dr. Daniel Kondermann for being an inspiring and supportive mentor who encouraged me to pursue research off the mainstream. I am thankful for his genuine guidance into the world of academia, the critical questions, and the countless discussions about algorithms, research, and beyond.

I thank the AIT Austrian Institute of Technology for funding my PhD and for challenging me with highly relevant research questions. In particular, I want to thank Dr. Wolfgang Herzner, Oliver Zendel, and Markus Murschitz for the close and fruitful collaboration during my PhD and beyond.

It was a great pleasure to work with Ole Johannsen and Prof. Dr. Bastian Goldlücke from the Computer Vision and Image Analysis (CVIA) group in Konstanz. What started with a general discussion about light field research at ICCV in Chile soon developed into a close collaboration that was not only successful and productive, but also fun, insightful, adventurous, and inspiring. I am proud and thankful about our joint commitment and perseverance to scope, design, implement, publish, deploy, and maintain an academic benchmark that has since been used (and still is) by hundreds of researchers around the world.

I want to thank Dr. Wolfgang Niehsen and his team at Bosch Hildesheim as well as Prof. Dr. Lena Maier-Hein and her team at DKFZ Heidelberg. They supported this work with real-world input data and with valuable insights about the challenges and requirements of stereo applications in the automotive and medical domain.

My gratitude also goes to the Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences for co-funding many of my travel expenses. Special thanks go to Dr. Michael Winckler and Maria Rupprecht for their efforts to make this graduate school a great place to meet fellow researchers.

I had the pleasure to work with many great colleagues and labmates at the HCI, especially Dr. Rahul Nair, Burkhard Güssefeld, Karsten Krispin, Hendrik Schilling, Dr. Marcel Gutsche, Alexander Brock, Dr. Max Diebold, Hamza Aziz

Ahmad, Dr. Sven Wanner, Dr. Christoph Garbe, and Dr. Florian Becker. Thanks for all the in-depth discussions, academic advice, highly scientific kicker matches, and the countless pieces of shared raspberry pie.

Special thanks go to Barbara Werner, Karin Kruljac, and Dominic Spangenberger for their considerable contributions to creating a great lab atmosphere and for supporting me with all administrative matters.

Last but not least, I want to thank everyone who proofread parts of this thesis: Dr. Rahul Nair, Alexander Schulze, Dr. Daniel Kondermann, Karsten Krispin, Dr. Stephan Meister, Alexander Brock, Ole Johannsen, Sophie Berckhan, and Prof. Dr. Bernd Jähne.

Contents

1 Introduction

1.1. Motivation

Meaningful performance characterization of computer vision algorithms is essential for scientific progress and commercial applications. Both, researchers and practitioners, are frequently facing questions such as how to optimize the parameters of a given algorithm for a specific task, how to identify the most suitable algorithm for a given application, or how to create representative performance profiles of different algorithms.

Academia. In academia, performance evaluation affects algorithm development itself as well as benchmark design and scientific review processes, hence, every aspect of computer vision research. Nonetheless, decisions on evaluation methodologies are rarely discussed in scientific papers. As of 2018, top-tier computer vision conferences do not provide subject areas dedicated to datasets or performance evaluation. Publications on algorithmic solutions for a given computer vision problem are often evaluated based on different metrics and test images, making an objective comparison difficult [44].

As an improvement of this situation, benchmarks with public leaderboards and periodic challenges gained great popularity in recent years. These commonly accepted and widely used benchmarks consolidate existing research and boost scientific progress by structuring the field and providing incentives to work on specific challenges, as e.g. the KITTI Vision Benchmark for real-world automotive stereo [88] or the Sintel Optical Flow Benchmark for large displacement optical flow [10]. However, with their great benefit and impact also comes great responsibility as flawed evaluation methodologies may hamper and misrepresent scientific progress. Benchmarks are rarely scrutinized by the community. They are regarded as fair and impartial platforms for objective comparison. Yet, characteristics such as non-representative scene content, inaccurate reference data, averaging error metrics, or scalar based rankings may skew algorithm comparisons or support overfitting [28, 17, 44]. Fur-

thermore, as pointed out by Clark and Clark [19], academic competitions for "the best algorithm" may defeat the purpose of fostering the development of better algorithms as second or third best methods may be stifled. The research community should be aware of the significant impact of their evaluation methodologies. A solid understanding and powerful tools are needed to appropriately measure and incentivize scientific progress.

Commercial Applications. In recent years, computer vision algorithms have evolved from being applied to academic pet problems to valuable tools for many real-world applications, e.g. in entertainment, driver assistance, or medical applications.

Safety-critical applications such as autonomous driving are complex and require rigorous performance assessment as algorithm failure may lead to life threatening situations. Many questions are yet to be answered on how to perform adequate performance evaluation for such complex tasks. At this scale and complexity, qualitative human performance assessment is unfeasible. Hence, powerful and reliable quantitative evaluation tools are indispensable.

Several evaluation methods exist which apply general performance measures to rather small academic test datasets [112, 88]. However, for real-life applications, it is often prohibitively expensive to capture reference data with the desired volume, quality, and content distribution. As a consequence, synthetic data generation, systematic sampling procedures, and data-aware evaluation metrics gain in importance.

1.2. Overview and Contributions

Thorough performance evaluation of computer vision algorithms is hard and research on this topic used to be little acknowledged. Christensen and Förstner as well as Haralick deplore a "lack of theory and methodology for testing" [18, 34], a "lack of acknowledgment" [18], and the "non-acceptance of empirical or theoretical comparisons of vision algorithms as original research" [18].

Indeed, many aspects make performance evaluation difficult, laborious, and cumbersome: Computer vision is complex, its models are often heavily simplified, and most algorithms have many tuning parameters but lack a sound theory [25]. Moreover, creating reference data is expensive [25, 85], sampling representative scene content is difficult [154, 36, 129], and it is not clear to what extent synthetic data may complement or replace real data [25, 85, 31]. Because of

these intricate challenges, prevalent performance characterization of vision algorithms is often limited to single scalar measures such as the mean squared error of disparity maps, making it difficult to truly grasp and compare the performance of a given algorithm.

In this dissertation, I review existing performance measures for depth estimation algorithms and propose a more comprehensive and easily applicable evaluation methodology. I argue that a thorough performance analysis must not only take into account the actual output of an algorithm but it should also be aware of input and reference data characteristics, requirements and constraints of the application domain, the impact of different error metrics, ranking schemes, and visualizations as well as their mutual influence on each other.

More specifically, my contributions can be summarized as follows. First, I provide a thorough literature review and derive a taxonomy of existing yet little noticed performance evaluation methods. I show that most methods are either theoretically sophisticated but difficult to use or easy to use but too simplistic to provide meaningful and reliable insights for practitioners. Prevalent methods tend to be easy to use but focus on averaging error metrics without taking local input data or reference data peculiarities into account.

Second, I address these issues and present an evaluation methodology which is both more easily applicable than the theoretical approaches and more comprehensive than prevalent approaches. I propose eight geometry-aware and semantically intuitive stereo metrics [46]. In contrast to other metrics, the proposed metrics incorporate spatial pixel dependency, local geometry differences, and application-specific priorities, while still being easy to implement. I show that visualizations and rankings based on the proposed metrics reveal specific algorithm strengths and weaknesses which are not reflected by existing metrics. The metrics can be applied to existing stereo data and support practitioners in selecting the most suitable algorithm for their application.

Third, I show how to overcome reference data deficiencies to obtain meaningful performance evaluations. I demonstrate how to use the proposed metrics when only limited annotation is available [66]. Together with Ole Johannsen, I introduce the concept of stratified scenes [44, 56]. These puristic but insightful scenes closely interweave test data and error metrics. They are designed to evaluate algorithm performance at specific algorithmic challenges.

In order to raise awareness, spark research interest, and push

the adoption of thorough performance evaluation methods, I co-organized four workshops, co-hosted two accompanying challenges, and released three online benchmarks (see Appendix B and C for details).

The "authorial we" is used for the remainder of this dissertation. Where applicable, substantial contributions by collaborating partners, joint research results, and shared publications are indicated at the beginning of each chapter.

1.3. Outline

The remainder of this thesis is structured as follows. In Chapter 2, we specify aspects of performance analysis and related terminology. We explain the basic concepts and challenges of stereo algorithms and briefly review stereo applications and their requirements. In Chapter 3, we review related work and derive a taxonomy of performance analysis approaches for stereo algorithms.

In Chapter 4, we present our geometry-aware disparity metrics to quantify performance at continuous surfaces, discontinuities, and fine structures. In Chapter 5, we demonstrate how to apply these metrics to overcome dataset deficiencies. We further introduce the concept of stratified scenes.

In Chapter 6, we use systematic test examples and human rankings to assess the correctness and expressiveness of our proposed metrics and stratified scenes. For the case study in Chapter 7, all performance evaluation aspects are incorporated into the creation of a public academic light field benchmark. We discuss our design choices and demonstrate how our metrics, datasets, and visualizations support a holistic and comprehensive performance analysis of light field algorithms.

2

Background

In this chapter, we first specify the numerous aspects that affect performance evaluation of computer vision algorithms. For each aspect, we describe how it is related to the scope of this thesis.

Second, we briefly explain basic concepts and typical challenges of stereo and light field algorithms. We further describe the diverse application domains for such algorithms and their vastly different requirements.

2.1. Aspects of Performance Evaluation

The prevalent evaluation method for evaluating stereo algorithms consists of the following: Algorithm results are computed for an academic dataset of 10-200 image pairs. Per algorithm and image, the percentage of pixels with errors beyond a certain threshold is computed. The average percentage per algorithm across all images is then compared to identify the superior algorithm [112, 88]. While this procedure may serve as a good performance indicator, many additional aspects strongly affect the evaluation result and should therefore explicitly be taken into account. This includes for example: the difficulty of the input data, the parameterization of the algorithm, and the error metrics for identifying the best algorithm.

In this section, we provide an overview of the various aspects that affect performance evaluation (see Figure 2.1), namely the application domain and evaluation objective, input and reference data, algorithm details, error metrics and visualizations as well as ranking schemes. We put these aspects into context with the scope of this thesis. We demonstrate that there are many open questions with no definite, universal strategy on how to incorporate these performance aspects into the evaluation. The strengths and shortcomings of existing approaches are reviewed in Chapter 3. Our proposed improvements and contributions are presented in Chapter 4 and Chapter 5.

2.1.1. Application Domain and Evaluation Objectives

Computer vision algorithms are used for very different applications and their performance is evaluated with respect to a wide range of different objectives. These objectives affect many aspects of the evaluation procedure and must therefore be carefully defined [26, 63].

For instance, researchers in the academic world often aim at a general, one-size-fits-all algorithm to solve a certain task, e.g. "dense two frame stereo reconstruction", imposing rather weak constraints on aspects such as runtime or memory usage. Their goal is to achieve an accurate reconstruction at all image regions and to perform well according to given performance measures of public academic benchmarks. By contrast, engineers in the industrial world often aim at solving a specific problem with strict resource constraints but also with handy assumptions on the application domain, e.g. "quality control of a specific car component on a conveyor belt". Their goal is to reconstruct the component sufficiently fast and accurate in order to decide whether its shape complies with given standards.

As illustrated by these examples, performance evaluation objectives and priorities are strongly affected by specific requirements of the task, viable assumptions on the application domain, and necessary constraints on the algorithm. Beyond the mere average accuracy, performance evaluation might take into account aspects such as speed, resource efficiency, testability, predictability, explainability, graceful degradation, accessibility, robustness, source code availability, licenses, or adaptability of the parameterization [132, 128, 64].

Goals of the performance evaluation might be: selecting the most accurate of n available algorithms for a given application, optimizing a single algorithm to perform as good as possible on a given task, obtaining a fair comparison of the strengths and weaknesses of different academic algorithms, assessing the impact of a specific algorithm component, or finding the most robust all-round algorithm with a guaranteed lower performance limit [13, 121].

For this thesis, we focus on accuracy and robustness evaluation of stereo algorithms in the automotive, medical, and academic domain. With our evaluation methodology, it is possible to obtain comprehensive performance profiles of the tested algorithms. They can be used to thoroughly characterize individual algorithms, to compare advantages and drawbacks of multiple algorithms, or to select the most suitable algorithm for a set of application-specific priorities.

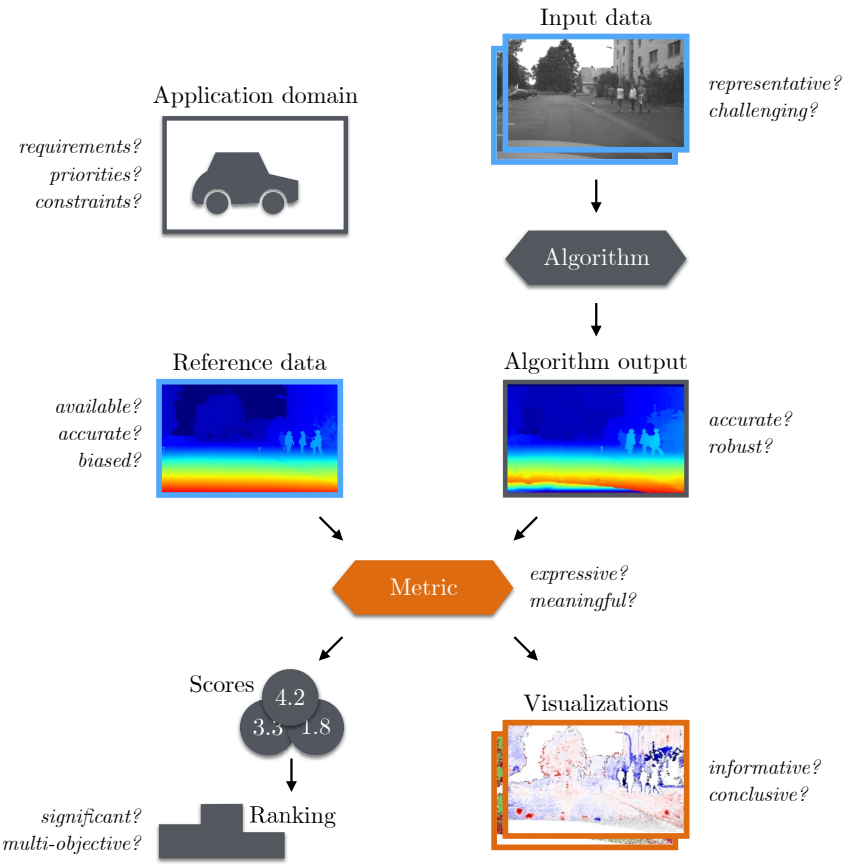


Figure 2.1.: **Aspects of Performance Evaluation.** Beyond the actual algorithm results, thorough performance evaluation should take into account application-specific requirements, characteristics of the input and reference data, the impact of metrics, ranking schemes, and visualizations as well as their mutual influence on each other (underlying images of the street scene: HCI dataset [66]).

2.1.2. Input Data and Reference Data

When creating an evaluation dataset, many questions arise concerning scene content, dataset size, acquisition modalities, and reference data creation. There is no definite set of instructions on how to create a representative and unbiased dataset with the appropriate level of diversity and difficulty. Beyond semantic scene content, this also affects more technical aspects such as distributions of geometric shapes, radiometric properties, camera characteristics, or image compression artifacts [36, 156]. Taking into account all these different aspects quickly leads to a combinatorial explosion of possible images. Yet, for evaluation purposes, a bigger dataset may not necessarily be a better dataset [156, 85]. A huge dataset is more likely to cover many situations but it may also be prohibitively expensive and laborious to create high quality reference data [28]. A smaller dataset allows for a considerate and systematic dataset design and high quality reference data [111]. However, it may lack crucial aspects and mislead dataset users to make unjustified generalizations.

When creating reference data for a dataset, there is always a trade-off between quantity, quality, time, and cost [63]. It strongly depends

on the application, what kind and quality of reference data is most useful and reasonable and which sensors and algorithms are most suitable for creating the data [87, 95]. For instance, there is no universal answer on how to combine the pros and cons of real and synthetic data with respect to input data diversity or reference data quality. Real data tends to be more representative for the actual application data. However, it is challenging and expensive to capture an adequate variety of situations and to create high quality reference data [4, 63, 87]. For synthetic data, it is easier to systematically vary scene parameters such as material properties and to generate high quality reference data [82, 44]. However, rendering realistic input data is intricate and laborious [85]. Even though synthetic data is often referred to as a way to create perfect reference data [4], limited resolution and rendering artifacts can impair reference data quality [44]. Most importantly, it remains an open question, how well evaluation insights gained on synthetic data can be transferred to performance on real data [86, 85, 31].

In some situations, reference data cannot be obtained at all. Additional human annotations [22, 76] or canny evaluation methods [121, 109, 91, 24] may compensate for missing reference data and still produce a reasonable performance characterization.

In this thesis, we use both, real and synthetic data. We demonstrate how to use synthetic data with systematic variations to perform degradation analysis. Furthermore, we show how to use minimal human annotations to overcome reference data deficiencies of real data.

2.1.3. Algorithm Details

Algorithms differ considerably with respect to their accessibility, parameterization, and modularity. Availability ranges from full access to well documented source code, over access to binaries with adjustable parameters, to limited access of the final output only. For many algorithms on academic benchmarks, only the final result but no source code is available. Parameters are tuned by the respective algorithm authors and results obtained from the best global parameter settings are submitted to the benchmark. However, many algorithms perform very differently depending on their parameterization. It is an open research question how to find a good set of parameters which maximizes algorithm performance and is robust for a wide range of different input scenarios. Apart from parameterization, pre- and post-processing steps often have a huge impact on

algorithm performance. For instance, outlier filtering and smoothing considerably improve scores of averaging error metrics. Hence, access to source code with a modular algorithm implementation is valuable for performance evaluation as the effects of different components can be evaluated individually [109, 41, 42, 98].

To maximize applicability and accessibility, performance evaluation in this thesis is focused on the output files and does not require access to algorithm source code.

2.1.4. Error Metrics and Visualizations

On the most general level, an error metric can be regarded as a distance function between the expected and the actual algorithm result. However, for a specific computer vision task, it is often not clear how to define the metric exactly or how to incorporate multiple, potentially contrasting, performance aspects like surface smoothness and sensitivity to fine details. According to related literature, error metrics should ideally be expressive, unbiased, reliable, automatic, and consistent as well as semantically intuitive, easy to use, commonly accepted, and efficient to compute [20, 15, 46]. However, typically, no methods are provided to quantify how well a given metric fulfills these desired characteristics.

Beyond individual error scores, appropriate visualizations like error maps or 3D point clouds can be powerful tools to reveal specific strengths and weaknesses of different algorithms. When dealing with multiple scenes, algorithms, and metrics, it is often difficult to get a good grasp of relative performance. Many research papers provide tables with average scores only. While being easy to understand, a lot of information is lost when simply averaging across scenes or metrics, potentially leading to misinterpretation. In computer vision research, visualizations like histograms, scatter plots, or radar charts are rarely used to display richer error statistics, [16, 46].

Furthermore, there is little discussion or research on how to deal with mutual dependencies between different metrics or between data and metrics. When averaging across multiple scenes of a dataset, the results are potentially skewed by the distribution of scenes in the dataset or of image regions in the scenes.

For this thesis, we propose error metrics which are sensitive to local scene geometry and thereby more robust to dataset bias. We propose multiple specific and semantically intuitive metrics to accommodate for different application priorities and to allow for a thorough algorithm assessment. We define a specific error map visualization for

each metric and use radar and scatter charts for multi-dimensional performance analysis. For our light field benchmark, we take advantage of interactive web possibilities to display high dimensional data and to let the user decide how to marginalize and condition in the space of algorithms, scenes, and metrics [45].

2.1.5. Rankings

When dealing with multiple scenes, algorithms, and metrics, it is not obvious how to define an overall rank and whether it is actually desirable at all to define such a ranking.

On most benchmarks, overall ranks are defined based on the average error on some error metric and all scenes [88] or based on their average ranks for the individual scenes [112]. In both cases, top ranking benchmark participants are tempted to claim superior performance even for marginal differences between the individual results. More sophisticated ranking methods were proposed (e.g. based on statistical significance [98], Pareto dominance [11, 13], or practical difference [70]) but have so far not been widely put into practice for computer vision benchmarks. Ranking differences should be meaningful for the respective algorithms, relevant for the application requirements, and appropriate for the available reference data accuracy.

For this thesis, we do not propose novel ranking methods but carefully decide if and how algorithm should be called superior to other algorithms. Instead of a single overall ranking, we argue for multi-dimensional radar chart visualizations which are closely related to Pareto dominance.

2.2. Binocular Stereo and Light Field Algorithms

Stereo algorithms use images of two cameras with known relative positions as input to infer the distance to the depicted scene. In this section, we briefly explain the underlying principles and describe commonly used assumptions and stereo challenges. We further describe common applications of depth estimation techniques and explain the relation to light field algorithms.

2.2.1. Triangulation

In this section, we briefly explain the concept of triangulation for inferring depth from stereo images. We refer to Jähne [53] and Hartley and Zisserman [37] for detailed explanations of epipolar geometry.

Depth from triangulation estimates the difference in projection when taking images from two different viewpoints. This difference can easily be observed by alternately closing the left and right eye. The horizontal position of close-by objects changes considerably while the position of more distant objects changes only slightly.

Figure 2.2 depicts a prototypical stereo setup with two pinhole cameras C_L and C_R. The cameras have identical *focal lengths* f and parallel optical axes. The right camera C_R is shifted horizontally with respect to the left camera C_L by the *baseline* distance b. Within this setup, the 3D world point $P(X, Y, Z)$ is projected to different locations $P_L(x_L, y)$ and $P_R(x_R, y)$ on the image planes of the cameras C_L and C_R. This difference in projection is called the *disparity* d, with $d = x_R - x_L$. Based on similar triangles (compare Figure 2.2), the following relations hold true:

$$x_L = f \cdot \frac{X}{Z}, \qquad x_R = f \cdot \frac{X + b}{Z} \tag{2.1}$$

which leads to:

$$d = x_R - x_L = f \cdot \frac{X + b}{Z} - f \cdot \frac{X}{Z} = \frac{f \cdot b}{Z}, \tag{2.2}$$

i.e. the distance Z of a 3D point to the camera plane is inversely proportional to the distance d between the two projections of this point. The main challenge for stereo algorithms consists of identifying corresponding projections in the two images.

2.2.2. Correspondence Search

Thanks to epipolar geometry, the search space for a corresponding projection in a binocular camera system with known calibration is

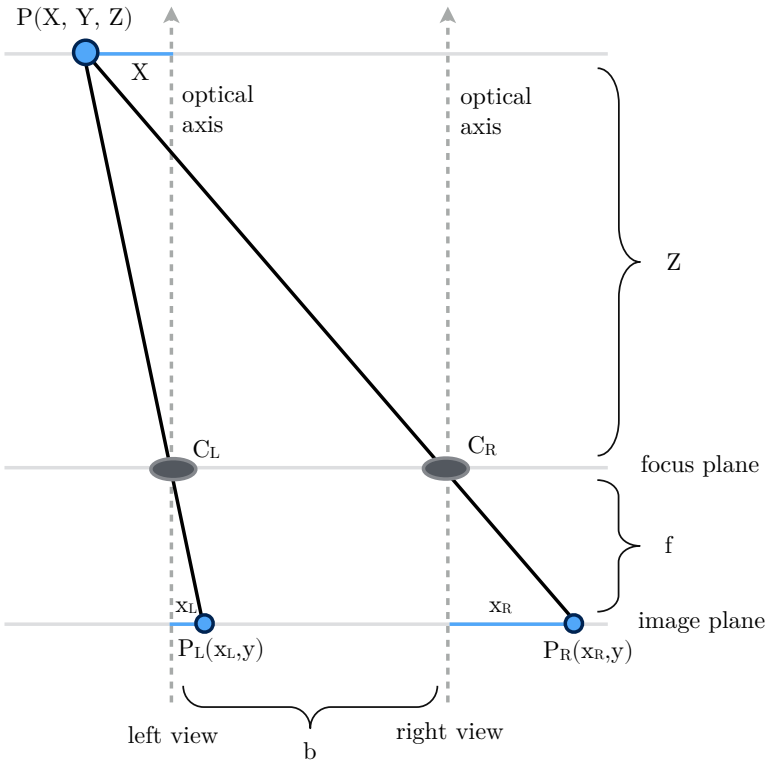


Figure 2.2.: **Depth From Triangulation.** The prototypical stereo setup consists of two pinhole cameras C_L and C_R with identical focal lengths f and parallel optical axes. C_R is translated horizontally by the baseline b. The 3D world point $P(X, Y, Z)$ is projected to different positions $P_L(x_L, y)$ and $P_R(x_R, y)$ on the camera sensors and hence the resulting images. The disparity $d = x_R - x_L$ between these projections is inversely proportional to the distance Z of $P(X, Y, Z)$ to the camera plane: $d = f \cdot b/Z$.

limited to the epipolar line [37]. In practice, input images are rectified to simulate perfectly aligned cameras. This simplifies the stereo matching since it limits the correspondence search to the same image row (see Figure 2.3). For each pixel on the left image, stereo algorithms try to identify the corresponding projection on the same row of the right image to compute the disparity d. Despite this limitation of the search space, stereo matching is a non-trivial task due to matching ambiguities and non-matchable pixels. Textureless areas and occlusion at object boundaries (see Figure 2.3) are just two of various stereo matching challenges.

Numerous stereo algorithms were proposed to create disparity maps from stereo input images. In this thesis, we focus on the basic algorithmic principles and common challenges. We refer to Scharstein and Szeliski [109] for a comprehensive taxonomy and comparison of stereo algorithms. More recent algorithms can be found on the leaderboards of the Middlebury [112] and Kitti [88] benchmark.

Stereo algorithms can roughly be grouped into local and global methods. Local methods use a sliding window approach to find the most similar image patch for each pixel. In its most simple version, the algorithm tries to derive a disparity map $\hat{D}$ such that:

$$\hat{D} = \underset{D}{\operatorname{argmin}} \left(I_L(x, y) - I_R(x + d, y)\right)^2, \tag{2.3}$$

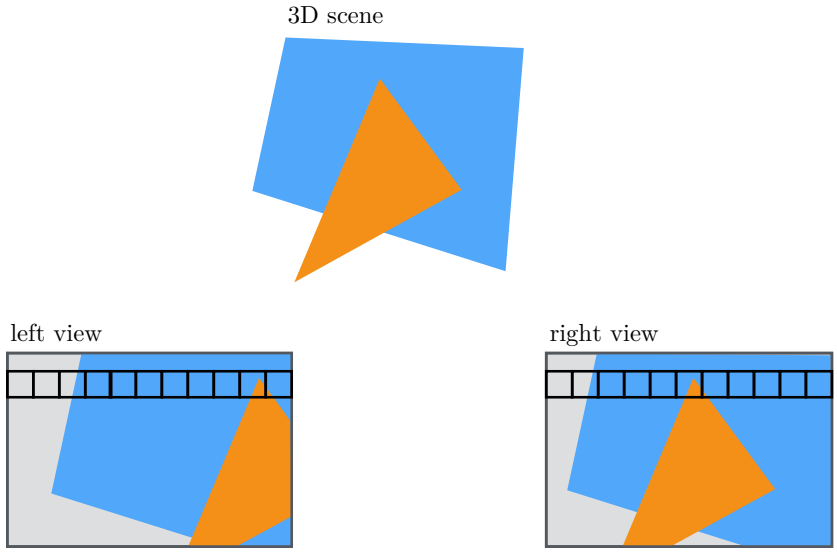


Figure 2.3.: **Correspondence Search.** On rectified images, matching pixels are located on the same image row. For each pixel on the left view, stereo algorithms try to match the corresponding pixel location on the right view. The peak of the orange triangle can easily be matched. Correspondence matching is harder for the textureless blue polygon and the partially occluded gray background.

where $\hat{D}$ contains a disparity estimate d for each pixel of the left input image I_L such that the color difference is minimized between the image patch around the left image pixel $I_L(x, y)$ and its corresponding patch around the pixel $I_R(x + d, y)$ on the right image. Various functions were proposed to quantify the color or gradient similarity between two patches [109, 41, 42]. As an advantage, local methods are simple and fast to compute. However, choosing an appropriate patch size is non-trivial. Small patches may lead to incorrect matches at local minima. Big patches may lead to object fattening at discontinuities. Furthermore, matching textureless areas such as the blue polygon in Figure 2.3 is highly ambiguous and cannot be solved without knowledge about the global scene structure.

Global methods typically solve an energy minimization problem based on a data and a smoothness term:

$$\hat{D} = \underset{D}{\operatorname{argmin}}\ E_{data}(I_L, I_R, D) + \lambda \cdot E_{smooth}(D). \tag{2.4}$$

The data term $E_{data}(I_L, I_R, D)$ penalizes appearance differences between the left image and the corresponding pixels on the right image. The smoothness term $E_{smooth}(D)$ penalizes strong gradients on the disparity map D. Global methods are more robust at low texture areas but they are often computationally expensive. Determining an appropriate local weighting λ between the data and the smoothness term is crucial to produce a smooth disparity map while preserving fine structures and discontinuities at object boundaries.

2.2.3. Common Assumptions and Stereo Challenges

As indicated in the previous section, stereo matching is a non-trivial task as pixel patches may be ambiguous or non-matchable. Stereo algorithms explicitly and implicitly apply various assumptions to compensate for missing information and to make the optimization strategy tractable. These assumptions simplify and improve stereo matching on most image areas [131]. Assumption violations typically affect only a small portion of the image. Therefore, their impact on averaging error metrics seems negligible. However, the affected image regions are particularly relevant for many applications and evaluation of algorithm performance at these regions is particularly insightful. In the following, we describe common stereo assumptions and provide examples for assumption violations.

Figure 2.4 depicts a stereo image pair and the corresponding disparity map of the HD1K dataset [66]. The boxes in Figure 2.4c indicate violations of common stereo assumptions.

Photoconsistency Assumption. Many algorithms assume that objects have the same appearance in the left and right image [42]. This *photoconsistency assumption* is violated by objects with non-Lambertian surface properties, by non-equal camera settings, or when stereo images are captured over time with changing lighting conditions. Finding the correct matches is much harder when the patches do not look alike on both images.

The windshields and puddles on Figure 2.4c (orange boxes) are examples of non-Lambertian surfaces. The appearance of such transparent or specular surfaces depends on the viewing angle and may thus be different on the two stereo images.

Unique Appearance Matching Assumption. Some algorithms assume that there is a single best matching patch on the right image for each left image patch. This assumption is violated by low texture areas and repetitive textures. The sky and parts of the right building on Figure 2.4c are saturated (green boxes). All pixel values are identically set to white, making it difficult to identify the correct match among many matches with identical matching cost. The repeated windows on the left building also lead to multiple local optima.

Continuity Assumption. Marr and Poggio [81] note that the world consists of mostly coherent objects. Most algorithms explicitly or implicitly encode this *smoothness constraint* [132].

(a) Left input image

(b) Right input image

(c) Assumption violations on the right input image

(d) Reference disparities for the left input image (red indicates high values)

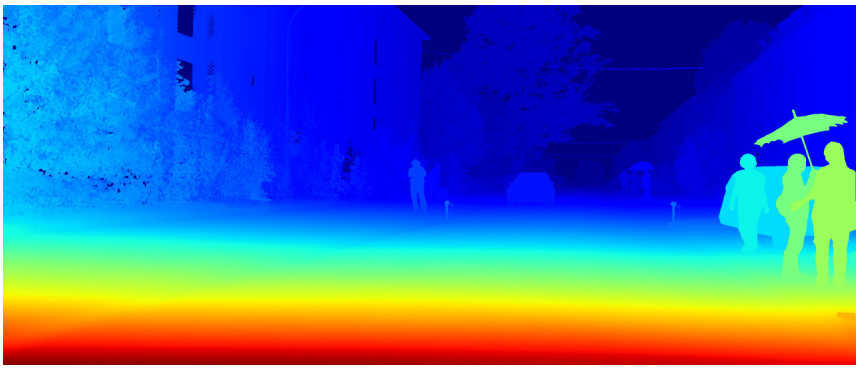


Figure 2.4.: **Examples for Violated Stereo Algorithm Assumptions.** Most image regions comply with the common stereo assumptions. The colored boxes in Figure 2.4c indicate violations of these assumptions.
Orange: The appearance of non-Lambertian surfaces such as puddles changes with the viewing angle, violating the *photoconsistency assumption*.
Green: Textureless image regions and repeated structures violate the *unique appearance matching assumption*.
Yellow: Depth discontinuities at object boundaries and high frequency geometries violate the *smoothness assumption*.
Blue: Semi-occluded objects violate the *unique geometry matching assumption*.
Purple: Thin, close-by structures like the stick of the umbrella violate the *ordering assumption*.
Red: Color gradients which do not coincide with disparity gradients violate the *figural continuity assumption*.

They rely on the assumption that disparities change smoothly between most image areas except for rare discontinuities at object boundaries and high frequency object surfaces.

Indeed, most image areas on Figure 2.4d have small disparity gradients. The vegetation and the object boundaries of the depicted people are examples for discontinuities (yellow boxes).

Unique Geometry Matching Assumption. Stereo algorithms assume that each point of the 3D world is projected exactly once into each stereo camera and that, vice versa, the projection at each pixel belongs to exactly one 3D point. This assumption is violated by occlusion regions, reflections, and semi-transparent objects.

The blue boxes on Figure 2.4c illustrate occlusion phenomena. The left-most part on the right image is not visible on the left image and hence, cannot be matched. The rear wheel of the parked car on the right is visible on the left image but occluded by a pedestrian on the right image. Many algorithms perform left-right consistency checks to detect occlusion regions. They try to apply additional reasoning to hypothesize about the correct matches at these regions.

Ordering Assumption. Most global algorithms implicitly encode an ordering assumption. As described by Baker [3] and generalized by Yuille and Poggio [152] scene objects tend to keep their relative order when projected into the left and right image. This monotonicity is also preserved when some parts are missing due to occlusion. Small, close-by objects (so-called "flies" [67]) violate this assumption.

In Figure 2.4c, the stick of the umbrella demonstrates such an effect (purple box). On the left image, the umbrella is located between the second and third window from the right. On the right image, the umbrella and the third window change their relative ordering.

Figural Continuity Assumption. According to Mayhew and Frisby [83], the human visual system assumes that object edges coincide with texture edges. In a similar way, many stereo algorithms assume that intensity differences in the input images correspond to disparity gradients and vice versa. For instance, high color gradients are often used as a prior for occlusion regions. As a consequence, occlusions at low color contrast regions are particularly prone to being missed.

In Figure 2.4c, the upper right part of the left building (red box) yields almost no color contrast to the saturated sky despite a considerable depth discontinuity. By contrast, the parking lot number on the left (red box) features strong color but no disparity discontinuity.

(a) Input image

(b) Estimated disparity map

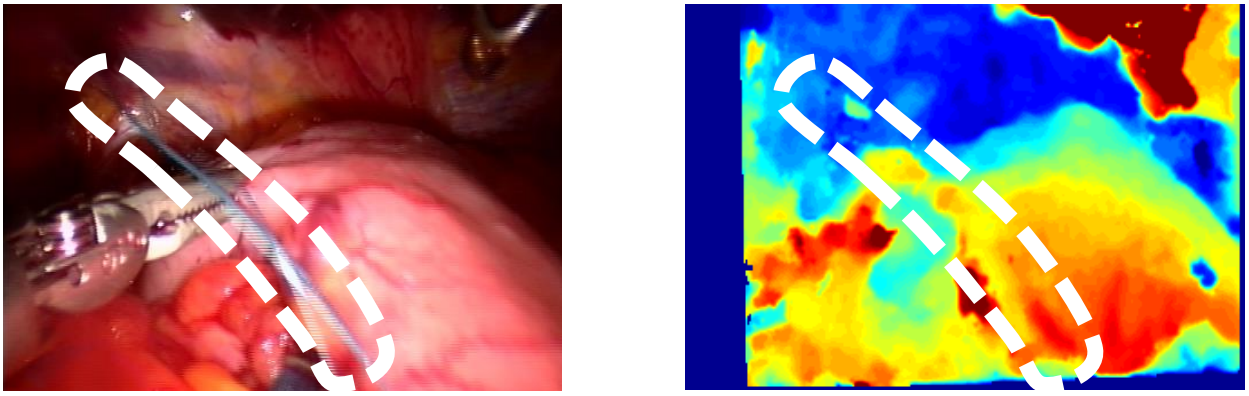


Figure 2.5.: **Medical Application.** Stereo reconstruction in the medical domain is challenging due to motion blur, compression artifacts, low texture, and specular reflections. Despite these adverse conditions, the reconstruction must be sensitive to thin instruments and sutures (image courtesy: Maier-Hein et al. [46]).

2.2.4. Applications and Requirements

Stereo algorithms are employed for numerous applications such as robotics [128], cartography [49], agriculture [100], image-based rendering [134, 24, 121], driver assistance [132, 30], or computer assisted surgery [78, 139]. The input images of these application domains are subject to very different constraints and impose diverse challenges. Furthermore, the application-specific requirements and priorities differ greatly with respect to accuracy, robustness, and runtime.

We demonstrate these differences by briefly describing application-specific challenges and requirements when using stereo algorithms for the media industry, medical imaging, and driver assistance systems.

Media Industry. In the media industry, stereo algorithms are used to support video editing tasks such as view interpolation or image matting. View interpolation depends on accurate reconstructions of the scene geometry to create synthetic views between two captured frames [24, 121]. For this application, the perceived quality is more important than metric errors. Hence, errors at homogeneously colored regions are less severe than temporal inconsistencies or errors at high texture regions [134, 90]. In contrast to most other application domains, constraints on computing resources or runtime are not very strict and expert manual user input is readily available to support the stereo reconstruction [120].

Medical Imaging. In the medical domain, stereo algorithms are used for computer assisted surgery [139, 78]. Figure 2.5a depicts an image of laparoscopic surgery. Common challenges for medical stereo applications include motion blur, homogeneous colors, specular reflections, smoke, and image compression artifacts [75]. Stereo

reconstruction for medical imaging must be robust towards these adversities. Furthermore, it must be fast and sensitive to fine structures such as thin instruments and threads.

Driver Assistance. In driver assistance systems, stereo reconstructions are used as input for higher level scene understanding tasks such as pedestrian detection [59, 58] or general obstacle avoidance [133, 33]. These safety-critical tasks impose strict requirements on the stereo reconstruction. Distance estimates must be computed in real time and with limited computing resources [128, 132]. Furthermore, the reconstruction must be accurate, robust to varying lighting conditions [23], and sensitive to fine structures such as traffic signs.

While sparse results may be sufficient for basic collision avoidance [134], dense reconstructions allow for a better image segmentation with accurate object boundaries [132, 101]. Underestimated object boundaries may lead to collisions while overestimated object boundaries may lead to unnecessary, potentially dangerous, collision avoidance maneuvers.

2.2.5. Relation to Light Field Algorithms

The metrics and dataset concepts proposed in this thesis are also applicable to evaluate light field algorithms. Light field algorithms can be regarded as another type of depth estimation algorithms. In this section, we provide a short introduction to the structure of light fields and briefly describe two common strategies for inferring depth from light fields.

Light Field Setup. Compared to the binocular stereo setup, typical light field setups allow for a much denser sampling of the 3D scene thanks to additional views and smaller baselines. Figure 2.6 depicts a common light field setup. The 3D scene in Figure 2.6b is captured by a regular grid of 5×5 cameras resulting in 25 so-called subaperture views (see Figure 2.6a). Analogously to common stereo setups, the cameras have parallel optical axes and equal focal lengths. They are shifted horizontally and vertically by a baseline distance b. The result of the depth reconstruction is typically provided as a disparity map of the so called *center view* (see cyan frame at the center of Figure 2.6a) to the right neighboring view in the grid. We refer to Vianello [138] for additional details on how light fields are acquired using camera arrays, translation stages, plenoptic cameras, and synthetic rendering.

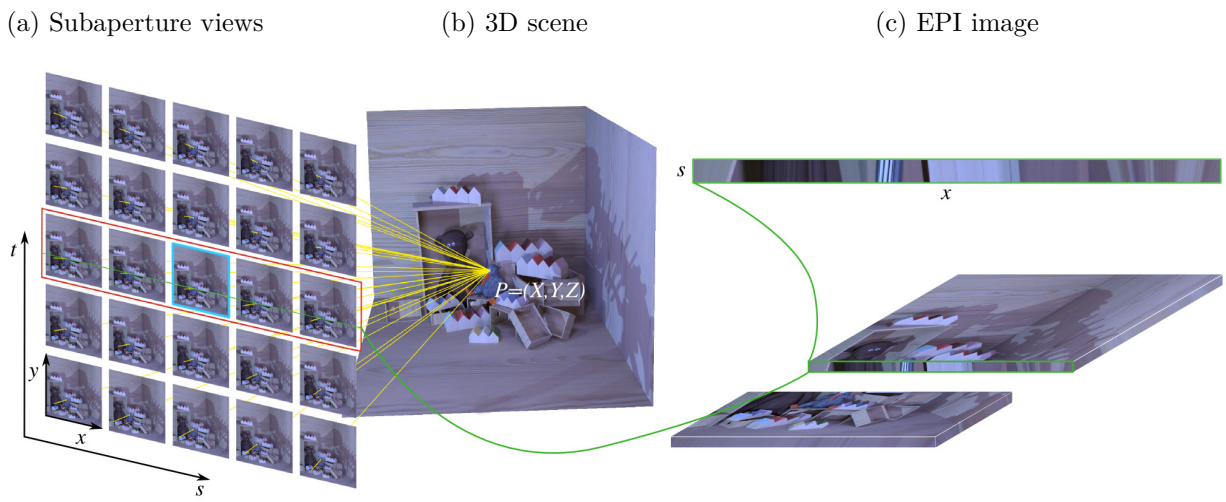


Figure 2.6.: **Light Field Representations.** The 3D scene in (b) is captured by a regular grid of 5 × 5 cameras, resulting in 25 subaperture views of the scene (a). The EPI image in (c) depicts the same image row of all subaperture views of the red grid row stacked on top of each other. The slope of the resulting lines is directly related to the distance of the respective scene objects and can thus be estimated to infer depth (image courtesy: Johannsen et al. [56]).

Depth from Light Fields. As described in the previous sections, finding the right correspondences is the main challenge for stereo algorithms. Occlusions, fine structures, and view-dependent appearance changes make this search difficult. The denser sampling and the additional views of light fields allow for a more robust and accurate reconstruction of the captured scene [141, 147, 39, 54]. The different strategies of light field algorithms are described in great detail by Ole Johannsen and Bastian Goldlücke in our light field taxonomy [56]. For this thesis, we briefly describe two approaches: subaperture views and epipolar plane images.

Strategies for depth estimation from subaperture views are very similar to stereo and multi-view algorithms [55, 99, 54]. Corresponding image patches are matched based on similar appearance of the depicted scene objects on all views (see subaperture grid on Figure 2.6a). Besides subaperture views, epipolar plane images are a commonly used representation for depth estimation from light fields. Figure 2.6c depicts the stacked subaperture views of the center grid row (see red frame in Figure 2.6a). The same image row of these subaperture views is used to create an EPI image as depicted on the top of Figure 2.6c. The resulting patterns of this EPI image feature useful properties for depth estimation [7]. Scene points with Lambertian surfaces form lines whose slopes are directly related to the distance of the respective scene object. Close-by objects move by greater distances between the subaperture views, leading to flatter lines in the EPI pattern. Instead of searching for correspondences, EPI based light field algorithms estimate the slopes of these lines to infer depth [57, 160, 145, 146, 113].

3
Related Work

There is a variety of noteworthy but rarely applied methods related to general evaluation concepts [36, 26, 21, 19, 126] as well as stereo-specific evaluation aspects [121, 114, 68]. In numerous publications, authors have claimed a lack of stereo evaluation techniques and proposed their own [72, 122, 109, 67, 13, 80, 115, 134, 127]. Despite this diversity, the vast majority of published stereo algorithms is evaluated based on the datasets, per-pixel error metrics, and public leaderboard rankings of the Middlebury [112] and Kitti [88] benchmarks.

In this chapter, we first review related work on general concepts of performance evaluation. Second, we compile and discuss stereo performance metrics. We define a unified notation to consolidate these metrics and derive a metric taxonomy based on data availability. Finally, we briefly review literature on stereo evaluation datasets.

3.1. General Concepts of Performance Evaluation

The most universal and mathematically sound approaches to evaluate computer vision algorithms were published in the 90s [36, 26, 21, 19, 79]. Back then, computing resources were much more limited and evaluation focused on rather comprehensible vision tasks such as corner detection. Compared to most contemporary evaluation methodologies, thorough mathematical modeling played a more important role for both, algorithm design and algorithm evaluation.

Many of these early evaluation approaches seem to have passed into oblivion. In this section, we highlight approaches which we consider as valuable inspiration for prevalent methodologies. At the end of this section, we discuss shortcomings and challenges when applying these early approaches to todays complex vision tasks.

Haralick [35, 36] proposes the establishment of a well-defined and statistically sound performance characterization protocol to allow

for well-founded comparisons between different computer vision algorithms. For research results to be used in engineering, one should define and know precisely i) the task to be done, ii) an appropriate error criterion, and iii) the performance of the algorithm under various degradations of the input data. For statistically sound performance characterization, a model of the ideal world with perfect data and a perturbation model are required to systematically sample a random, independent, and representative set of images. Based on statistical hypothesis testing, the number of samples needed for a desired uncertainty can be derived to test the hypothesis that an algorithm meets the application-specific requirements. Image "equivalence classes" are suggested to reduce the number of variable combinations [36]. Yet, no explicit procedure is provided to derive these classes.

Courtney et al. [21] focus on modeling the algorithm rather than the input data. The authors assume that "a simple algorithm with predictable performance may be better than a complex algorithm with better mean but less predictable performance". They propose to identify the statistical data distributions that affect algorithm results. Samples drawn from these distributions are then propagated through the modeled algorithm. In order to make the modeling feasible, the authors suggest a rigorous approach of intertwined algorithm implementation and evaluation. Algorithm complexity is only added when the effects of the change are adequately modeled.

Along these lines, Thacker et al. [126] define performance characterization as the process of "obtaining a sufficiently quantitative understanding of performance that the output data from an algorithm can be interpreted correctly". Ideally, algorithms should be designed and regarded as an estimation process to allow for more formal statistical analyses. Thacker and Courtney [125] demonstrate such an analysis on a simplistic corner matching algorithm.

Förstner [25] lists pros and cons concerning performance characterization of computer vision algorithms. While recognizing that performance characterization is challenging and often cumbersome, he emphasizes that a commonly accepted evaluation methodology is an essential prerequisite for scientific progress. Förstner acknowledges that obtaining ground truth is expensive, simulations are not reality, algorithms often lack well-founded theory, and performance measures are often poorly comparable. To address these adversities, Förstner proposes to share costs and efforts for creating test beds and to harness the benefits of simulation. He further advocates to only accept algorithm parameters with a well-defined meaning and

to rely on statistically motivated performance measures.

Clark and Clark [19] point out that academic competitions for "the best algorithm" may defeat the purpose of fostering the development of better algorithms as second or third best methods may be stifled. Instead, Clark and Clark propose the concept of a publicly available "straw man algorithm". Every approach outperforming this algorithm should be considered worth publishing and more focus should be put on explaining why the approach outperforms the straw man.

Despite their mathematical universality, most of these evaluation methods are hardly feasible for stereo evaluation in current research or real-world scenarios. Most analytical approaches require very good models of the algorithm [21, 126] or the input data [36, 26]. Both may be feasible for a simple corner detection method but much less so for complex real-world scenarios such as robust pedestrian detection in diverse geographic, lighting, and seasonal conditions. For such scenarios, these rigorous and comprehensive evaluation approaches quickly become intractable unless strong assumptions and compromises are applied. Nonetheless, certain aspects of the theoretical frameworks by Haralick, Förstner, or Courtney et al. are indeed used by prevalent evaluation methods though they are rarely put into that context. In the following sections, we provide details on stereo evaluation aspects which are related to Courtney et al.'s understanding of algorithm concepts [109, 41, 42], as well as Haralick's systematic data sampling [51, 155, 63, 62] and perturbations of the input data [93, 72, 132].

3.2. Stereo Performance Metrics

In this section, we provide a comprehensive survey of stereo evaluation metrics. We refine and extend the survey by Vargas et al. [136] and propose a more comprehensive taxonomy which is based on data availability (see Figure 3.1). Typically, performance evaluation is constrained by limited quantity, density, or accuracy of the available input, reference, and algorithm data. Furthermore, different applications impose different requirements on the stereo algorithms. Hence, there is no single best evaluation metric and diverse metric characteristics are desirable.

To highlight differences and to demonstrate the equivalence of certain metrics, we define a simple, unified notation for all metric definitions: let a be an estimated algorithm disparity map, r the reference

disparity map, and $\mathcal{M}$ an evaluation-specific set of pixels x_i of an image $\mathcal{I}$, e.g. all pixels at low texture regions.

The remainder of this section is structured according to our taxonomy as depicted in Figure 3.1. We first review stereo evaluation metrics for dense algorithm results with dense reference data. We then discuss methods for evaluation with limited data availability, namely methods for sparse algorithm results, as well as methods for sparse, weak, or missing reference data. Methods for evaluating particular stereo challenges are mostly based on specific datasets. They are discussed in Section 3.3.

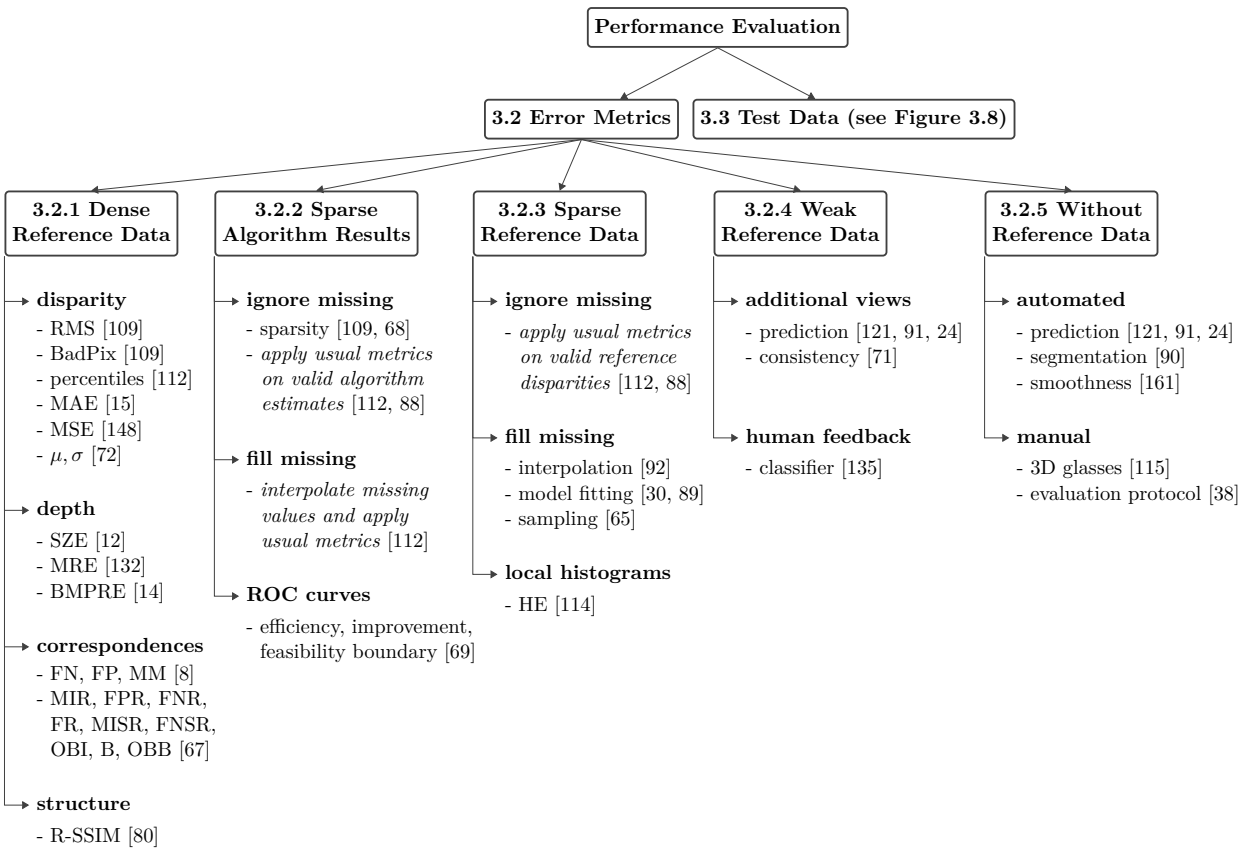


Figure 3.1.: **Taxonomy of Stereo Evaluation Metrics.** Various metrics were proposed for evaluation with full, sparse, weak, or missing reference data. They are described and discussed in the following sections.

3.2.1. Dense Reference Data

For dense reference data, performance metrics were proposed for four different perspectives: disparities, depth values, correspondences, and 3D structure.

Disparity Errors. The two prevalent metrics in the stereo community for evaluating disparity maps are *RMS* and *BadPix*. They were introduced by Scharstein and Szeliski [109] and are used by the two most popular stereo benchmarks, Middlebury [112] and Kitti [88].

The *Root Mean Squared Error* (*RMS*) quantifies the per-pixel differences between the reference disparity map r and the algorithm disparity map a. It is defined as:

$$\mathrm{RMS}_{\mathcal{M}}(a,r) = \left(\frac{1}{|\mathcal{M}|} \sum_{x \in \mathcal{M}} \big(a(x) - r(x)\big)^2 \right)^{\frac{1}{2}}, \tag{3.1}$$

based on all pixels x of the evaluation area $\mathcal{M}$, which usually comprises the full disparity map except for a narrow image boundary.

The general *BadPix* metric denotes the fraction of valid disparity estimates $\mathcal{M}_v = \{x \in \mathcal{M} \colon a(x) \in [d_{min}, d_{max}]\}$ whose error exceeds a certain threshold t. It is defined as:

$$\mathrm{BadPix}_{\mathcal{M}_v}(a,r,t) = \frac{100}{|\mathcal{M}_v|} \big|\{x \in \mathcal{M}_v \colon |a(x) - r(x)| > t\}\big|. \tag{3.2}$$

The valid disparity range $[d_{min}, d_{max}]$ is often set conservatively to $[0, \text{image width}]$. Common thresholds in the stereo community are $t = 1.0$ and $t = 3.0$ pixels [109, 28]. Sometimes, missing or invalid algorithm disparity estimates are also counted as "bad pixels":

$$\mathrm{BadPix}_{\mathcal{M}}(a,r,t) = 100 - \frac{100}{|\mathcal{M}|} \big|\{x \in \mathcal{M}_v \colon |a(x) - r(x)| \leq t\}\big|. \tag{3.3}$$

RMS and *BadPix* scores may also be computed on pixel subsets which provide additional insights for typical problem areas. The Middlebury and Kitti benchmark distinguish between non-occluded regions and the full reference data. Scharstein and Szeliski [109] propose to automatically identify regions with low texture, occlusions, or depth discontinuities. They compute scores on these image regions and their complements.

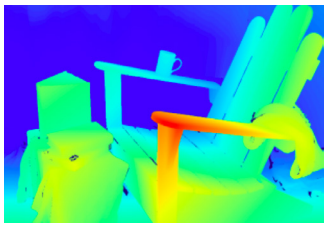
The Middlebury benchmark further reports the *Error Percentiles* P50, P90, P95, and P99, denoting the maximum absolute disparity difference on the best n percent of pixels. The $50th$ percentile is equal to the median error.

(a) Spatial dependency

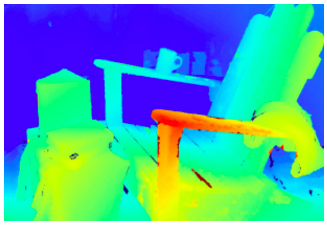
(b) Importance & difficulty

Figure 3.2.: **Relevant Pixel Characteristics.** Image pixels are neither spatially independent (a), nor equally important or equally challenging (b). For instance, pedestrians (red) are more important for automotive applications than background vegetation. Reflections and low texture areas (blue) are more challenging than regular street areas (underlying image: HCI dataset [66]).

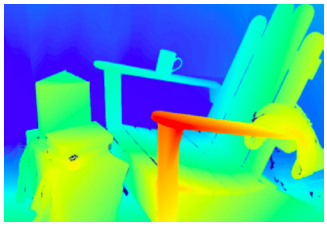
(a) Reference disparity map



(b) Typical algorithm result



(c) Single pixel outlier



(d) Missing cup

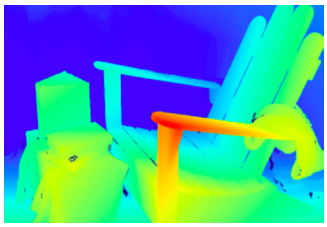


Figure 3.3.: **RMS Flaws.** All three disparity maps (b)-(d) feature the same *RMS* of 5.2 with respect to the reference data (a) of the *Adirondack* scene [111]. However, their error distributions are very different: (b) depicts a typical algorithm result with some inaccuracies at discontinuities. (c) is almost perfect with a single pixel outlier. (d) is mostly correct but the cup on the armrest is missing.

In rare cases, other statistics of the disparity error distribution are used. Cabezas et al. [15] mention the *Mean Absolute Error (MAE)*:

$$\mathrm{MAE}_{\mathcal{M}}(a,r) = \frac{1}{|\mathcal{M}|}\sum_{x\in\mathcal{M}} |a(x)-r(x)|. \tag{3.4}$$

The light field community mostly uses the *Mean Squared Error*, which is often multiplied by 100 [148, 44]. It is defined as:

$$\mathrm{MSE}_{\mathcal{M}}(a,r) = \frac{1}{|\mathcal{M}|}\sum_{x\in\mathcal{M}} \big(a(x)-r(x)\big)^2. \tag{3.5}$$

Leclercq and Morris [72] propose using the mean μ and standard deviation σ of the disparity error distribution instead of the *RMS* in order to reveal algorithm bias:

$$\mu_{\mathcal{M}}(a,r) = \frac{1}{|\mathcal{M}|}\sum_{x\in\mathcal{M}} \big(a(x)-r(x)\big), \tag{3.6}$$

$$\sigma_{\mathcal{M}}(a,r) = \sqrt{\frac{1}{|\mathcal{M}|}\sum_{x\in\mathcal{M}} \big(a(x)-r(x)-\mu_{\mathcal{M}}\big)^2}. \tag{3.7}$$

Advantages of the presented averaging error metrics are that they are generally applicable, parameter-free, easy to understand, and fast to compute. As a major drawback, all disparity estimates are treated equally. As depicted in Figure 3.2, image pixels are not spatially independent, nor are they equally relevant for applications or uniformly difficult for algorithms. As demonstrated in Figure 3.3, metric scores may be identical for very different algorithm results. Apart from being location-agnostic, this kind of metrics is also prone to outliers which may heavily distort the evaluation results.

By contrast, outliers are handled naturally by the *BadPix* metric. The threshold can be customized to the error range which is relevant for the application. However, setting the threshold appropriately is non-trivial and algorithm rankings may differ considerably depending on the threshold (compare Figure 3.4a and 3.4b). When choosing a threshold, the disparity range of the scene, ground truth accuracy limits, and application requirements should be taken into account.

Depth Errors. Some researchers claim that depth rather than disparity errors should be compared, especially for 3D reconstruction applications where a specific metric error tolerance is important [12, 132]. Depth z and disparity d are inversely related as $z = f \cdot b/d$, with focal length f and baseline b. A disparity error of 1px may

correspond to a small metric error at a foreground object with big disparities or to a bigger metric error at a background object.

Cabezas et al. [12] propose the *Sigma-Z-Error (SZE)*, which is in fact the sum of the absolute depth errors:

$$\mathrm{SZE}_{\mathcal{M}}(a, r) = \sum_{x \in \mathcal{M}} \left| \frac{f \cdot b}{a(x)} - \frac{f \cdot b}{r(x)} \right|. \tag{3.8}$$

As a more general approach, van der Mark and Gavrila [132] propose the *Mean Relative Error (MRE)*:

$$\mathrm{MRE}_{\mathcal{M}}(a, r) = \frac{1}{|\mathcal{M}|} \sum_{x \in \mathcal{M}} \frac{|a(x) - r(x)|}{r(x)}. \tag{3.9}$$

They address the fact that the same disparity error corresponds to a larger depth error at more distant objects while still being independent of specific camera characteristics.

As a combination of the *MRE* and the *BadPix*, Cabezas et al. [14] propose the *Bad Matched Pixels Relative Error (BMPRE)*:

$$\mathrm{BMPRE}_{\mathcal{M}_t}(a, r, t) = \sum_{x \in \mathcal{M}_t} \frac{|a(x) - r(x)|}{r(x)}. \tag{3.10}$$

The *BMPRE* metric quantifies the relative error among all estimates $\mathcal{M}_t = \{x \in \mathcal{M} \colon |a(x) - r(x)| > t\}$ that exceed a certain disparity error threshold t.

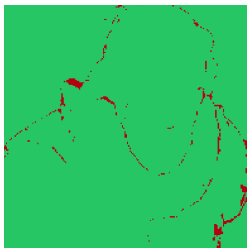
For a general, automated evaluation, we argue that disparity based error metrics should be preferred over depth based metrics. Disparity metrics represent a more universal, camera-independent measurement which can be compared more easily across different camera setups. However, when evaluating for an application with a relevant and meaningful metric interpretation, all of the presented disparity metrics may indeed be computed for the respective depth values.

Correspondence Errors. A third type of error metrics is focused on the perspective of correspondence matching between the left and right image pixels.

Bolles et al. [8] propose metrics for three types of correspondence errors: *Mismatches (MM)* for correspondence estimates at non-occluded areas with an estimation error greater than one, *False Negatives (FN)* for missing correspondences at non-occluded areas, and *False Positives (FP)* for correspondences at half-occluded areas. *MM* corresponds to $BadPix_{\mathcal{M}}(1)$ of Equation 3.2 with an evaluation mask $\mathcal{M}$ for non-occluded regions. *FN* is related to the *Sparsity*

(a) Tolerant threshold, $t = 0.07$

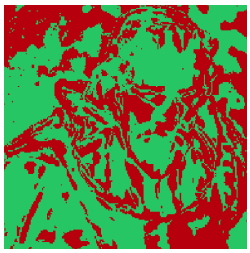
$a_1 = 1.8$



$a_2 = 3.0$

(b) Strict threshold, $t = 0.01$

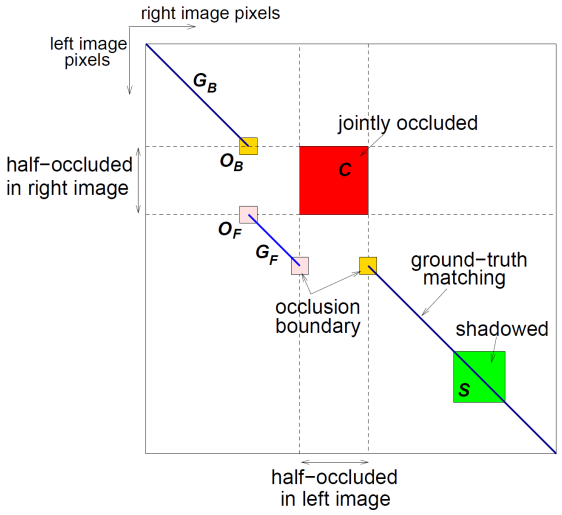
$a_1 = 43.9$



$a_2 = 26.7$

Figure 3.4.: **BadPix Threshold Sensitivity.** (a) When assessed with a tolerant threshold of 0.07, the light field algorithm a_1 achieves a better *BadPix* score than a_2. (b) When assessed with a stricter threshold of 0.01, a_1 performs much worse than a_2.

(a) Region definitions



(b) Algorithm result

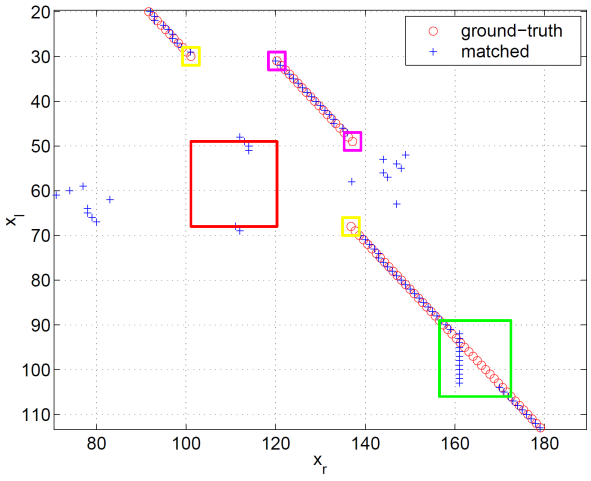


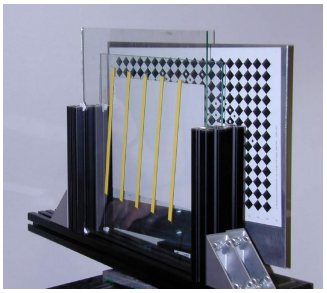
Figure 3.5.: **Matching Table by Kostková et al.** Algorithms are evaluated based on subsets of the full scanline disparity matching space between the left and right image row (image courtesy: Kostková et al. [67]).

of Equation 3.13. The *FP* metric assumes that algorithms return an explicit value for half-occluded regions. However, since most of todays algorithms perform scene filling of varying complexity at half-occluded regions [135], it is often more meaningful to explicitly quantify estimation errors at these regions.

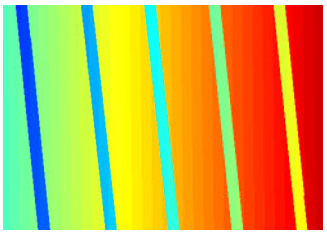
Kostková et al. [67] extend the metrics of Bolles et al. They define nine error metrics based on different correspondence subsets in the context of *matching tables* (see Figure 3.5a). For each row in the stereo image pair, a matching table depicts all possible matches P between the left and right image row. In the literature, this table is closely related to the *disparity space image (DSI)* [130, 6] and similarity matrices [131]. Typically, half of the matches are ruled out due to the assumption of parallel optical axes, limiting the disparity space to positive values. In the following, we briefly describe the proposed metrics and show that most metrics are closely related to the Middlebury evaluation, including special treatment of low-texture and occlusion regions and their complements as well as the explicit handling of invalid versus inaccurate matches. The *Mismatch Rate (MIR)* corresponds to *BadPix(1)* at textured, non-occluded regions (see Equation 3.2). In contrast to *BadPix*, *MIR* is normalized per scanline before the overall average is computed. The *False Positive Rate (FPR)* counts the number of algorithm estimates at jointly occluded regions C. The *False Negative Rate (FNR)* quantifies the sparsity by counting the missing disparity estimates at non-occluded, textured regions. The *Failure Rate (FR)* corresponds to the stricter version of the *BadPix* which also penalizes missing values (see Equation 3.3). The *Mismatch Rate at Textureless Re-*

gions (MISR) corresponds to *BadPix(1)* at textureless regions S. In contrast to Scharstein and Szeliski [109], occluded regions are not ignored. The *False Negative Rate at Textureless Regions (FNSR)* counts the sparsity at textureless regions, again not excluding semi-occluded regions. The *Occlusion Boundary Inaccuracy (OBI)* corresponds to $BadPix_{\mathcal{M}}(1)$ with $\mathcal{M} = O$ (Equation 3.3). *Bias (B)* quantifies the difference between *BadPix* scores at foreground objects R_f and background objects R_b. This metric seems to be tailored to the test data as depicted in Figure 3.6. In this scene, textureless areas S only occur in the background and there is a clear notion of foreground and background objects. However, for arbitrary scenes, Kostková et al.'s completeness principle is violated. For instance, it is not clear how R_f and R_b should be assigned to a scene with three layered objects. As a special version of B, the *Occlusion Boundary Bias (OBB)* quantifies bias at occlusion regions.

(a) Scene setup



(b) Reference disparity map



(c) Evaluation masks

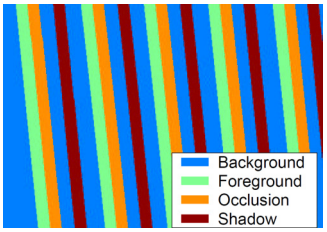


Figure 3.6.: **Test Scene by Kostková et al.** This striped scene supports focused performance evaluation at occlusion boundaries. It features a clear notion of foreground stripes and background scenery (image courtesy: Kostková et al. [67]).

Structure. A fourth type of error metrics is focused on the relative structure of disparity estimates rather than their individual values.

Malpica and Bovik [80] propose the multi-scale *Range Structural Similarity (R-SSIM)* as an adaptation of the multi-scale *SSIM* by Wang et al. [143]. The *SSIM* is used for image similarity assessment and reportedly corresponds better to human perception than the *MSE* [144]. Instead of pure per-pixel comparisons, the *SSIM* quantifies the image similarity at each pixel based on the similarity of the pixel neighborhood with respect to luminance, contrast, and structure [142]. Malpica and Bovik apply the same metrics to range images with an interpretation of depth, surface roughness, and 3D structure. The average *R-SSIM* is defined as:

$$\text{R-SSIM}_{\mathcal{M}}(a,r) = \frac{1}{|\mathcal{M}|}\sum_{\boldsymbol{x}\in\mathcal{M}} d(\boldsymbol{x}_a,\boldsymbol{x}_r)^{\alpha} r(\boldsymbol{x}_a,\boldsymbol{x}_r)^{\beta} s(\boldsymbol{x}_a,\boldsymbol{x}_r)^{\gamma}, \quad (3.11)$$

where $\boldsymbol{x}_a, \boldsymbol{x}_r$ denote a pixel neighborhood of the algorithm and reference disparity map with typical sizes of around 9×9 and weights usually set to $\alpha = \beta = \gamma = 1$. More specifically, the similarity in depth d, roughness r, and structure s between two neighborhoods $\boldsymbol{x}$ and $\boldsymbol{y}$ is quantified as:

$$d(\boldsymbol{x},\boldsymbol{y}) = \frac{2\mu_x\mu_y}{\mu_x^2+\mu_y^2}, \qquad r(\boldsymbol{x},\boldsymbol{y}) = \frac{2\sigma_x\sigma_y}{\sigma_x^2+\sigma_y^2}, \qquad s(\boldsymbol{x},\boldsymbol{y}) = \frac{\sigma_{xy}}{\sigma_x\sigma_y}. \quad (3.12)$$

where μ_x denotes the Gaussian weighted average disparity of all pixels $i \in 1,2,...,N$ of the neighborhood $\boldsymbol{x}$ with $\mu_x = \sum_{i=1}^{N} w_i x_i$. Sim-

ilarly, σ_x^2 and σ_{xy} denote the weighted disparity variance and covariance. For the multi-scale *R-SSIM*, the roughness and structure components are computed for multiple scales and all results are multiplied with the full resolution depth similarity. Malpica and Bovik report a high statistical correlation between *R-SSIM* and *BadPix* scores on the Middlebury benchmark [80]. They report a single qualitative example with two algorithm results where the *R-SSIM* assigns a better rank to the visually more appealing result as compared to the *BadPix* ranking. Since the *R-SSIM* is considerably more expensive to compute than the *BadPix*, one should consider carefully when to use it.

3.2.2. Sparse Algorithm Results

For applications such as obstacle avoidance in robotics [134] sparse disparity maps often provide sufficient information. Their computation is faster and more resource efficient. They also tend to be more accurate since only high confidence correspondences are used.

The most simple and common approach for evaluating sparse algorithm results is to ignore missing values and to compute the error metrics on valid disparity estimates only [112]. An additional sparsity score may be reported:

$$S_{\mathcal{M}}(a) = \frac{100}{|\mathcal{M}|}|\mathcal{M} \setminus \mathcal{M}_v| \tag{3.13}$$

i.e. the percentage of invalid pixels where $\mathcal{M}_v = \{x \in \mathcal{M} \colon a(x) \in [d_{min}, d_{max}]\}$ denotes all pixels x with a valid disparity estimate $a(x)$. This approach works fine for mostly dense algorithm results. However, with increasing sparsity, the local distribution of the correspondences becomes more important as entire objects may be missing and go unnoticed.

As a second approach, algorithm results are made dense prior to evaluation. The Middlebury benchmark applies a simple scanline-based interpolation to compare dense and sparse methods [112].

Kostlivá et al. [68] propose an evaluation method based on Receiver Operating Characteristics (ROC). They assess how estimation errors and sparsity change with different algorithm parameter settings. The *Sparsity Rate (SR)* denotes the percentage of missing correspondences based on all matchable pixels (as in Equation 3.13). The *Error Rate (ER)* corresponds to the *BadPix(1)* definition in Equation 3.2 which does not penalize missing values. To create the ROC curve, different parameter settings are sampled and their *ER*

and *SR* scores are computed. Those points are used for the ROC curve for which no point is found with both, better accuracy and better density (see Figure 3.7). The *Feasibility Boundary* is defined as the ROC curve based on all algorithms combined. It provides a notion of the best possible performance.

The proposed metrics and visualizations by Kostlivá et al. are valuable complements to the prevalent Middlebury evaluation. They provide insights on the influence of parameter settings and support algorithm selection for a specific application with clear priorities for density or accuracy. However, defining the right sampling of the parameter space for the proposed experiments remains a non-trivial task. Inappropriate sampling may heavily distort the results and computing results for many parameter combinations is expensive.

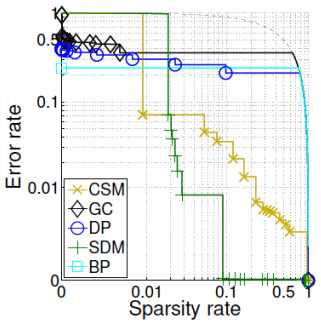


Figure 3.7.: **ROC Based Evaluation by Kostlivá et al.** Algorithm *SDM* features the lowest error rate at sparse results while *BP* and *DP* perform better at dense results. To quantify sparsity versus accuracy across different parameter settings, the area of the resulting ROC curves is computed. (image courtesy: Kostlivá et al. [68])

3.2.3. Sparse Reference Data

Many depth acquisition methods such as LIDAR scanners or structured light approaches lead to sparse reference data [28] or at least occasional invalid depth measures [111].

The most common approach for performance evaluation with sparse reference data is to simply ignore pixels with missing reference disparities as done on the Middlebury [112] and Kitti benchmark [88]. However, if the algorithm and reference disparity map are sparse, per-pixel metrics may find only few corresponding estimates.

As a second approach, sparse reference data is made dense, e.g. for static scenes via specific sampling methods [65] or via algorithmic augmentation based on additional 3D object information such as cars [30, 89].

Morales and Klette [92] combine *BadPix* and confidence scores to evaluate stereo performance on sparse reference data. *BadPix* scores are computed where reference data is available. For each remaining disparity estimate, three closeby 3D depth reference points are selected and the geometric properties of the point sets in the reference and algorithm depth map are compared.

Sellent and Wingbermühle [114] propose a histogram based evaluation method that can deal with both, sparse reference data and sparse algorithm results. It does not depend on interpolation or warping methods to densify the disparity maps. Instead of per-pixel comparisons, the *Earth Mover's Distance (EMD)* [107] is used to quantify the difference between the disparity histograms of the algorithm and reference disparity map. The *Histogram Error (HE)* is

defined as the average histogram distance across 2^n image tiles:

$$\mathrm{HE}^n = \frac{1}{2^n} \sum_{i=1}^{2^n} \mathrm{EMD}(h_a^i, h_r^i), \tag{3.14}$$

where h_a^i and h_r^i denote the normalized disparity histograms of the algorithm and reference disparity maps a and r at the image tile $i \in \{1, 2, ..., 2^n\}$. As advantages, the *Histogram Error* is robust to disparity density differences and forgiving to alignment errors. It is more sensitive to big missing objects as the simple sparsity score. However, reference correspondences do need to feature a minimum sampling of the image. Furthermore, the selected tile size and histogram binning strongly influence the resulting score. The authors note that dividing the image into four tiles usually suffices to reliably quantify the reconstruction of scene objects. It should be considered carefully that appropriate choices heavily depend on the local scene structure, disparity range, and application requirements. Large tiles at image regions with diverse disparities can easily be quantified to be similar to very noisy algorithm results.

3.2.4. Weak Reference Data

In some situations, no disparity reference data is available at all. Quantitative evaluation may still be performed using other types of data, e.g. additional views of the same scene or human annotations.

Szeliski [121] proposes the *Prediction Error (PE)* as a quality metric for stereo results. It does not require disparity reference data. Instead, calibrated multi-image stereo data is required such as the 5×5 grid of the "Head and Lamp" scene of the University of Tsukuba [97]. The stereo result from two input images is used to predict a third, inter- or extrapolated view which is then compared to the actual third image of the scene.

Morales and Klette [91] apply a similar trinocular approach on long stereo sequences in the context of driver assistance systems. *RMS* and *Normalized Cross Correlation (NCC)* are used to quantify the difference between the predicted and the actual views. Führ et al. [24] use *SSIM* and *PSNR* scores between the predicted and actual views to quantify stereo performance for view interpolation.

These view prediction evaluations are well suited for perception related applications such as view interpolation in video editing. For instance, view interpolation errors at low texture areas are less visible and therefore less relevant than at high texture areas. This is implicitly factored in the *PE* but not in evaluation metrics which are

purely based on disparity value comparisons. However, as a significant drawback, view prediction performance does not only depend on the disparity map of the algorithm but also heavily on the warping and interpolation schemes as well as on the applied image similarity measure. One should further be aware that the *PE* is tolerant to errors in uniform regions and sensitive to isolated pixel errors. Errors at repetitive textures or other systematic errors may also be missed.

Leclerc et al. [71] propose running the algorithm under test on several different image pairs of a multi-view dataset. The consistency between the different 3D reconstruction results is used to quantify algorithm performance. Such a self-consistency metric may serve as a good indicator to detect gross errors. However, consistently wrong estimates will go unnoticed.

As a third approach, Varekamp et al. [135] propose a learning based method to detect and correct disparity errors. Human graders visually inspect disparity maps and annotate areas which appear erroneous. These labels are then fed into a binary classifier which uses basic image features. Research has shown that human annotators are capable of accurately annotating individual correspondences given the right tools [77]. However, it is questionable how manual annotation of erroneous disparities can be performed accurately and cost-efficiently. Furthermore, the classifier is at risk of learning how to identify challenging image regions rather than disparity errors.

3.2.5. No Reference Data

In some situations, no additional data is available other than the input images and the algorithm result but performance evaluation is still desired. Some techniques aim at an automated quantitative evaluation by harnessing additional information from the input images or disparity maps. Other techniques propose evaluation protocols for human graders to judge the algorithm results. Given the difficult setup, most of these evaluation strategies understandably suffer from considerable limitations. The automated techniques are prone to bias as many of the exploited assumptions are also used by stereo algorithms. With manual graders, repeatability as well as intra- and inter-grader consistency are challenging. However, the following techniques still provide valuable insights when applied carefully for applications where perceived quality or relative consistency are more relevant than metric depth values.

Among the automated techniques, the previously presented view prediction methods [121, 91, 24] may also be applied to predict the

target stereo view instead of a third, independent perspective.

Milani et al. [90] separately perform k-means based segmentation on the color image and the disparity map. For both results, the no-reference quality metric F_{RC} for segmentation by Rosenberger and Chehdi [106] is computed. The consistency between the segmentations is used as an indicator for disparity map quality. Such a comparison is an interesting approach for automated evaluation without any reference data or manual interaction. However, the underlying assumption that color edges are closely related to depth edges is often violated by strong intensity gradients at smooth surfaces and vice versa. Since many algorithms use this *figural continuity assumption* (see Section 2.2.3) as an occlusion prior, one should be particularly careful with biased results in favor of such algorithms.

Zhang et al. [161] quantify the smoothness of estimated disparity maps under the influence of increasing levels of noise based on local first and second order disparity gradients. The authors assume smooth disparity maps and apply *BadPix*-like thresholds on the gradient maps to quantify performance. It is not stated how erroneous gradients caused by noise artifacts are distinguished from correct gradients due to object boundaries or steep smooth surfaces. Without this distinction, evaluation results are likely to depend mostly on scene geometry rather than specific algorithm results.

Among the manual techniques, Shen et al. [115] ask 20 non-expert reviewers to judge the depth map quality of stereo results for different noise levels using red/green glasses. Reportedly, the mean scores for the perceived quality and the quantitative measure *R-SSIM* (see Equation 3.11) both decrease monotonically with higher noise levels.

For the *HCI Robust Vision Challenge* [38], eight computer vision experts were asked to judge submitted algorithm results from challenging input scenes for which no reference data was available. The experts were given specifically designed visualization tools and guidelines for the evaluation in order to facilitate a fair evaluation.

3.3. Stereo Evaluation Data

Apart from the choice of performance measures, the selection of test data strongly affects the evaluation results. Algorithm performance may vary heavily within an image or across different images. Hence, mindful creation, selection, or perturbation of datasets is crucial for meaningful performance evaluation. In this section, we review theoretical principles, viable techniques, and existing stereo datasets with notable features for evaluation purposes (see Figure 3.8).

The prevalent test datasets for depth estimation evaluation are sampled from the real-world [28], acquired in experimental lab setups [111], generated synthetically [82, 105, 98], or derived from requirements analysis and captured in a controlled environment [66]. We refer to our dataset analysis paper [154] for an extensive survey of 28 stereo datasets and an in-depth analysis of the Middlebury [111], Kitti [28], Sintel [9], Freiburg [82], and HCI [66] stereo datasets.

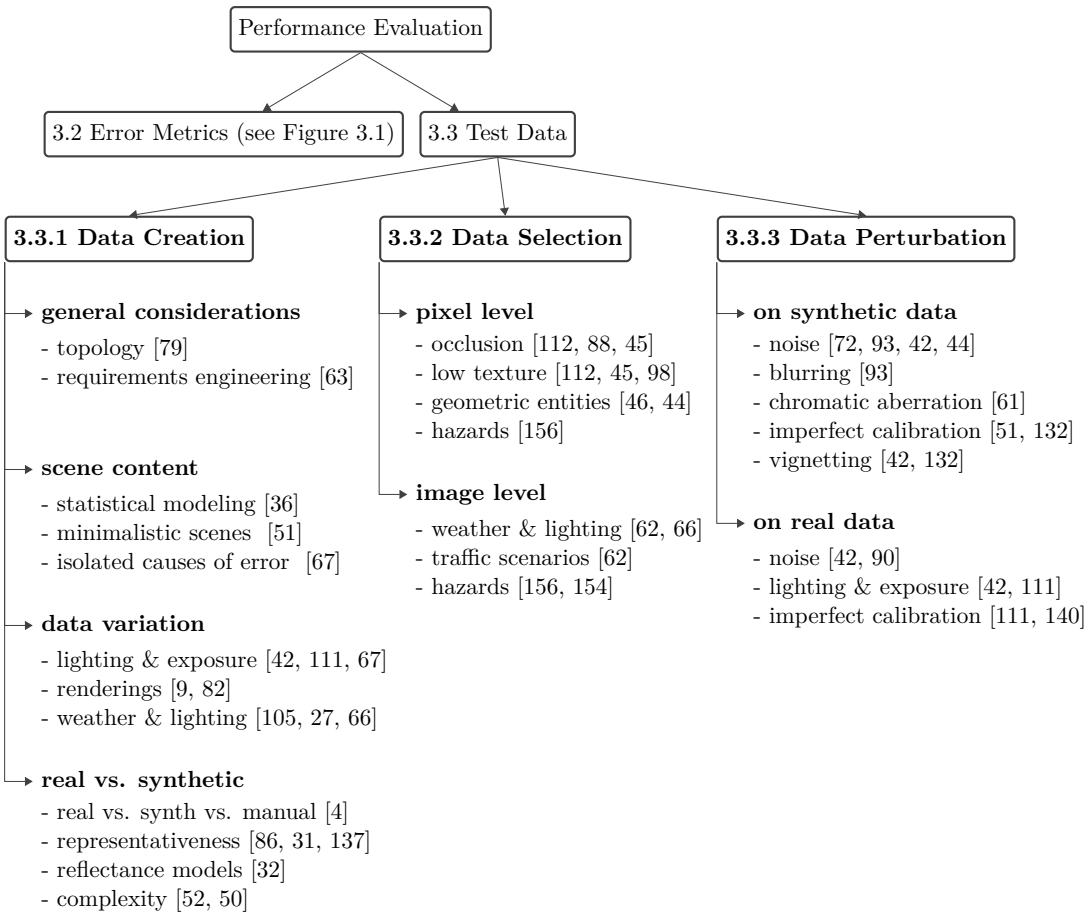


Figure 3.8.: **Taxonomy of Stereo Evaluation Data.** Apart from error metrics, suitable evaluation data is required for meaningful algorithm evaluation. It may be newly created by acquiring or rendering new data, selected as specific subsets of existing data, or created by perturbing existing data.

3.3.1. Data Creation

When creating new test datasets, questions arise concerning the distribution of the scene content and the technical data acquisition. In this section, we briefly review work related to general dataset considerations, scene content sampling, systematic data variation, as well as differences between real and synthetic datasets.

General Considerations. Maimone and Shafer [79] propose a topology of test data for stereo computer vision experiments. They analyze which performance characteristics can be assessed on test setups ranging from empirical testing on uncontrolled and controlled real data environments, over simulations, to pure mathematical analysis.

Kondermann [63] derives ground truth design principles for creating datasets with an appropriate trade-off between quantity, accuracy, time, and cost. He emphasizes the importance of requirements engineering and discusses advantages and drawbacks of performance evaluation with accurate, weak, or no ground truth.

(a) Short exposure

(b) Long exposure

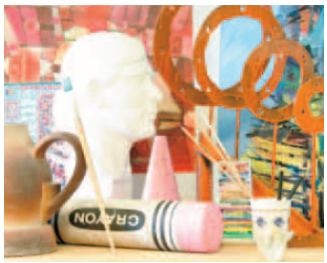


Figure 3.9.: **Test Data with Radiometric Challenges by Hirschmüller and Scharstein.** The *Art* scene was captured with varying exposures and light sources (image courtesy: Hirschmüller and Scharstein [42]).

Scene Content. Haralick [36] argues that a model of the ideal world with perfect data and a random perturbation model are required to systematically sample a representative set of images. Such an idealized, mathematically sound, and comprehensive model of the relevant data is highly desirable but also hardly feasible for complex computer vision tasks.

Häusler and Kondermann [51] propose to synthetically render isolated core problems of stereo methods without requiring photorealism. They argue that synthetic data must look realistic to the algorithm under test but not necessarily to humans.

In a similar way, Kostková et al. [67] argue that test data should be designed such that specific causes of errors can be evaluated individually. As an example, they capture a two-layer test scene with thin stripes on a piece of glass in the foreground which cast shadows on a background plane (see Figure 3.6).

Data Variation. Hirschmüller and Scharstein [42] systematically create real and synthetic data to assess the performance degradation of stereo algorithms under radiometric changes. They provide real data with three different exposures and light sources (see Figure 3.9) as well as synthetically augmented data with linear and non-linear brightness differences, vignetting effects, and Gaussian noise.

Many popular stereo datasets feature rarely used data variants

which are of particular interest for performance evaluation. The Middlebury datasets [110, 111] come with systematic lighting and exposure variations. The synthetic datasets Sintel [9] and Freiburg [82] provide a clean and a final rendering pass. The final rendering features motion blur, defocus, and atmospheric effects. The HCI Stereo Dataset [66] features variation of lighting, weather, and traffic scenarios for real data captured on a single street. The synthetic datasets Synthia [105] and VirtualKitti [27] provide systematic variations of lighting, weather, and seasons for various scene geometries.

Real versus Synthetic Data. Barnard and Fischler [4] discuss the pros and cons of synthetic data, real data, and human graders for performance evaluation. For synthetic data, it is easier to control generation parameters and to derive reference data. However, it is unclear how well evaluation results can be transferred to real-world performance. For real data, capturing reference data is often difficult and expensive, hence limiting data accuracy and quantity. According to Barnard and Fischler, human graders are reasonably inexpensive but susceptible to errors and of limited accuracy.

Meister and Kondermann [86], Güssefeld et al. [31], and Vaudrey et al. [137] investigate the differences in algorithm behavior when applied on synthetic or real data and their consequences on performance evaluation. They find that algorithm performance is comparable when highly sophisticated reflectance models are used for rendering. In Güssefeld et al. [32], we show that simple reflectance models are sufficient for certain materials for the goal of obtaining comparable optical flow error distributions.

Häusler et al. [52] propose a method based on SIFT feature matching [74] to compare the difficulty of stereo datasets. The matching is performed between the left and right input image without the epipolar constraint. The total number of matches and the number of incorrect matches are used as complexity measures. According to these measures, synthetic and engineered images such as the Middlebury data [41] feature lower complexity than real-world scenes.

3.3.2. Data Selection

On existing datasets, relevant evaluation data may be selected on a pixel or image level. Many benchmarks define pixel masks for image areas such as occlusions [112, 88, 45] or low texture [112, 45]. Images may further be tagged, e.g. based on weather and lighting conditions [66] or traffic scenarios [62].

(a) Underexposure

(b) Specular reflection

(c) Motion blur

(d) Fog

Figure 3.10.: **Hazards in Popular Stereo Datasets by Zendel et al.** Hazards like underexposure (a), reflecting puddles (b), motion blur (c), or fog (d) are particularly challenging for many stereo algorithms. The depicted images are from the Kitti [28], HCI [66], Freiburg [82], and Sintel [9] datasets.

Zendel et al. [156] introduce *CV-HAZOP*, the application of an established risk analysis procedure to the computer vision domain. The authors derive a generic computer vision model of the image generation process and the algorithm under test. Based on this model, a checklist with several hundred entries is created. Each entry describes a *hazard*, a difficult situation which may decrease algorithm performance, e.g. underexposure, occlusion, or sensor noise. With this checklist, hazards can be identified systematically on a pixel or image level. Based on the obtained labels, algorithm performance and robustness are evaluated with respect to specific hazards. In Zendel et al. [154], we use the CV-HAZOP checklist to systematically assess the hazard coverage of popular stereo datasets (see Figure 3.10). We show that algorithms perform worse on hazard frames as compared to regular frames and provide a list of missing hazards which are not covered by existing datasets.

3.3.3. Data Perturbation

Various perturbations are applied to existing stereo data in order to perform systematic performance evaluation.

Leclercq and Morris [72] evaluate robustness to noise by using synthetic data with varying amounts of additive Gaussian noise. Morales et al. [93] apply various levels of Gaussian blurring, constant intensity changes, and white Gaussian noise to long synthetic stereo sequences of traffic scenes. Milani et al. [90] add different levels of quantization noise resembling image compression artifacts. Klette et al. [61] add different levels of blooming and chromatic aberration to synthetic traffic scenes. Calibration errors are added by Scharstein et al. [111], Häusler and Kondermann [51], van der Mark and Gavrila [132], as well as Wang et al. [140].

Such artificial approximations of image degradations may provide valuable insights into algorithm robustness. However, they should be used carefully and data with actual image degradations may be required for validation.

3.4. Conclusion and Outlook

As demonstrated in the previous sections, a multitude of metric and data related evaluation concepts exists. However, only very few metrics and datasets are actively used by the stereo community. Error metrics tend to be either widely established but too general or more specific but too complicated to use. In the following chapter, we in-

troduce geometry-aware stereo metrics. We show that these metrics feature a reasonable trade-off between providing specific insights and being widely applicable on existing datasets.

Furthermore, most existing approaches focus on either metric or data aspects. Except for commonly computed *BadPix* scores on occlusion masks, pixels tend to be treated equally, independent of their spatial relationship, algorithmic difficulty, or semantic importance. Our metrics address this issue and explicitly take local data characteristics into account.

4

Performance Metrics

In this chapter, we introduce theoretical principles for the quantitative evaluation of depth estimation algorithms at continuous surfaces, disparity discontinuities, and fine structures.

First, we show that prevalent metrics struggle with quantifying performance differences at image regions which are highly relevant for stereo applications. Second, we derive metric requirements from these observations and present our metric design principles. Third, we introduce eight novel stereo metrics which quantify algorithm performance at continuous surfaces, discontinuities, and fine structures. We conclude with briefly describing how to incorporate our metrics and visualizations into multi-dimensional performance analysis.

We published the initial version of these performance metrics with a focus on stereo algorithms [46]. Together with Ole Johannsen, we defined extensions of these metrics to evaluate light field algorithms for the *4D Light Field Benchmark* [44, 56]. Details of our benchmark evaluation methodology are described in the case study in Section 7.

Introduction

As described in Section 2.2.4, stereo results are often used as crucial input for higher-level vision tasks such as object detection or image based rendering. For safety-relevant applications such as driver assistance [103] or computer aided surgery [78], stereo algorithms must be thoroughly evaluated to identify the most suitable algorithm and to warrant minimum performance requirements. Depending on their theoretical approach, implementation details, and parameter settings, algorithms exhibit different advantages and drawbacks which are prioritized differently depending on the application.

In academia, the Middlebury [112] and Kitti [88] stereo benchmarks defined the established evaluation methodology to quantitatively assess and compare algorithm performance. Most top ranking methods on these benchmarks rely on initial correspondences based on `MC-CNN` matching costs which are obtained from a convolutional neural network by Zbontar and LeCun [153]. Stereo methods

then apply different refinement steps which go beyond purely pixel based reasoning and try to explicitly incorporate local scene geometry. Approaches address aspects such as surface normal estimation (`SNP-RSM` [159]), segment-wise plane fitting with slanted support windows (`LocalExp` [123], `3DMST` [73]), occlusion handling at depth discontinuities (`FDR` [151]), and attention to fine details (`LW-CNN` [102]). Despite such higher-level reasoning, performance evaluation on the Middlebury and Kitti benchmark is limited to pixel-wise comparison of disparities. *BadPix* and *RMS* scores are computed separately on the full image and on non-occluded image areas.

We argue that more specific performance metrics are required to reflect the performance differences of state-of-the-art stereo methods. As highlighted on three close-ups of Middlebury scenes on Figure 4.1, performance between algorithms differs considerably depending on the local scene geometry. Algorithm a_0 features the lowest performance at depth discontinuities but the best performance at fine structures. By contrast, a_1 produces very crisp discontinuities but staircased planar surfaces and almost no fine structures. Quantitatively assessing these differences is relevant and valuable for both, researchers and practitioners.

We propose eight semantically intuitive metrics which characterize algorithm performance at continuous surfaces, depth discontinuities, and fine structures. For each of these geometric entities, we first identify relevant image regions from existing reference data. We then apply our evaluation functions to quantify phenomena such as edge fattening or fragmentation of fine structures.

4.1. Metric Requirements

As discussed in Section 2.1.1, an evaluation methodology should support specific objectives of performance evaluation.

First, the evaluation methodology should support a thorough performance characterization, revealing advantages and drawbacks of different algorithms [121, 122, 68]. Therefore, it should allow for a detailed, quantitative assessment of specific, diverse, and relevant performance aspects. Second, the evaluation methodology should support decision making for researchers and practitioners when looking for the most suitable algorithm [12, 46]. Therefore, it should be generally applicable, semantically intuitive, easy to use, and customizable to prioritize different requirements. In order to meet these goals, our metrics should fulfill the following five requirements.

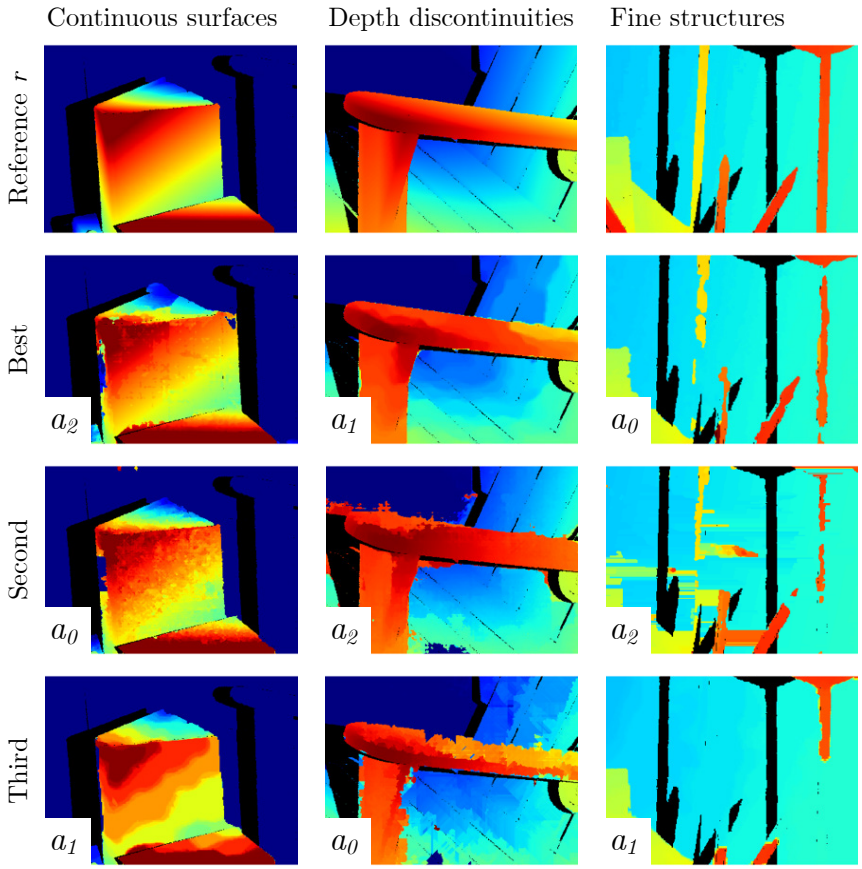


Figure 4.1.: **Varying Algorithm Performance at Different Geometric Entities.** The top row shows reference disparity maps of Middlebury scene close-ups [111] at continuous surfaces, depth discontinuities, and fine structures. The second row depicts the best performing algorithm for each geometric entity. Black denotes occluded regions.
The algorithms a_0 - a_2 perform very differently on the three geometric entities. a_0 performs well at fine structures but poor at discontinuities while a_1 has opposite strengths and weaknesses. Our proposed metrics are capable of quantifying these differences. They allow for an expressive and semantically intuitive assessment of stereo performance (figure based on Honauer et al. [46]).

R1. Provide a detailed performance profile. As described in Section 2.2.4, stereo applications often impose a set of diverse requirements, such as smooth surface reconstruction and crisp object boundaries. We define a diverse set of metrics in order to quantify how well different algorithms satisfy such requirements.

Our metrics quantify performance at continuous surfaces, depth discontinuities, and fine structures. For each of these geometric entities, we derive performance aspects which are relevant for practitioners and challenging for algorithms, such as edge fattening at discontinuities or fragmentation at fine structures. Separately quantifying these performance aspects allows for a thorough understanding of different algorithm strengths and weaknesses.

We emphasize the importance of both, quantitative evaluation and qualitative inspection, to obtain a comprehensive understanding of algorithm performance. Therefore, we pay attention to expressive visualizations of individual performance aspects as well as overall performance comparisons.

R2. Complement the expressiveness of prevalent metrics. As described in Section 3.2.1, the prevalent *RMS* and *BadPix* metrics are widely used to quantify general performance of stereo algorithms. As depicted on Figures 3.2 and 3.3, input pixels for stereo applications are neither spatially independent nor equally relevant or equally challenging. The prevalent metrics are not capable of reflecting these differences since all pixels are treated equally.

We design our metrics to explicitly take these differences into account. Our metrics are applied specifically on image regions of certain geometric entities. Furthermore, our metric functions explicitly take spatial pixel dependencies within these regions into account.

R3. Be semantically meaningful. In order to support the decision making of practitioners, each metric should have a clear purpose and a concise interpretation of what it quantifies. For instance, the quantification of edge fattening should match expert assessment obtained from visual inspection.

The design of our metrics intentionally prioritizes intuitive meanings such as smoothness and orientation of surfaces over perfect orthogonality. This makes it easier for practitioners to select the most relevant metrics for their application. However, implicit dependencies between metrics must be taken into account when comparing performance profiles of multiple algorithms.

R4. Be widely applicable. Despite being more specific than the *RMS* and *BadPix* metrics, our metrics should still be generally applicable to any stereo data, independent of semantic scene content.

To satisfy this requirement, we define image regions which are more specific than evaluating on the entire image but still general enough to be ubiquitous on stereo datasets. The relevant regions for our metrics can be identified algorithmically [46], extracted automatically from synthetic data [44], or annotated manually [66].

R5. Be easy to use and customizable. As concluded from Chapter 3, metrics should be easy to use and reasonably fast to compute. To allow for reproducible and fair comparisons, metric computation should be fully automated and require as few parameters as possible.

Instead of deriving metrics with maximum accuracy but intractable computational complexity or manual effort, we aim at finding a good trade-off between accuracy and usability. Where applicable, we propose two versions of our metrics. The simple version can be computed easily and provides a solid quantification of algorithm per-

formance. The more sophisticated version provides better accuracy and robustness at the cost of higher complexity. Most of our metric functions do not depend on parameters. For some metrics we provide thresholds with clear semantic meaning to adjust to the specific needs of different applications.

4.2. Geometry-Aware Stereo Metrics

In this section, we introduce theoretical principles for the quantitative evaluation of stereo performance at continuous surfaces, disparity discontinuities, and fine structures. For each of these geometric entities, we provide a definition, motivate their relevance for stereo applications, and explain the related algorithmic challenges when estimating disparities at these image regions. From these applications and challenges, we then derive relevant and meaningful phenomena at the geometric entities and propose specific metrics to formally quantify algorithm performance.

As in Section 3.2, a and r denote the algorithm and reference disparity maps. $\mathcal{M}$ denotes an evaluation specific set of pixels x_i of an image $\mathcal{I}$. For each proposed metric, zero denotes a perfect result. Higher scores indicate lower performance.

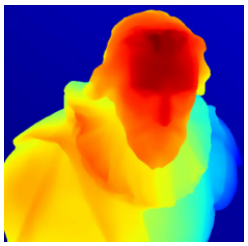
4.2.1. Continuous Surfaces

Definition. Continuous surfaces are defined as image regions where the disparity differences between adjacent pixels are low and change only smoothly, i.e. there are no big disparity jumps. Planar surfaces are treated as a special case of continuous surfaces with constant disparity change (see visualizations of the *Cotton* scene [44] on Figure 4.2c). Many objects in everyday scenes feature continuous surfaces or can be approximated by piecewise planar surfaces.

Relevance. As described in Section 2.2.4, reconstructed object surfaces are used with very different requirements among stereo applications. For image based rendering, accurate surface orientations play an important role to apply appropriate shading and illumination. For visually appealing results on view interpolation, smooth results with few artifacts are more important than general per-pixel depth accuracy [24]. For industrial inspection, reconstruction algorithms must be sensitive to local surface deviations of the inspected objects. Overly smooth results may lead to undetected defects and overly bumpy results may lead to mistakenly rejected objects.

(a) Input image I

(b) Disparity map r



(c) Regions M_p, M_n

(d) Normal map $\vec{n}_r$

(e) Max curvature c_r

Figure 4.2.: **Evaluation at Continuous Surfaces.** For each disparity map r (b), we extract pixel sets (c) with planar $\mathcal{M}_p$ (blue) and non-planar continuous surfaces $\mathcal{M}_n$ (orange). We further compute normal maps $\vec{n}_r$ (d) and curvature maps c_r (e).

Algorithmic Challenges. Reconstructing continuous surfaces such that they are smooth as well as correctly oriented is challenging. Some algorithms frame depth estimation as a labeling problem with a discrete set of disparities [54, 160]. This strategy may cause staircasing effects which are only partly alleviated by subsequent refinement procedures. Strong regularization produces smooth results but high frequency details are prone to get lost. A common strategy among many algorithms is to fit local planes or splines to some sort of superpixels [47, 158, 117, 118, 150, 116]. Their parametrization often is a trade-off between locally accurate fits with jumps between the superpixels or smoother yet less accurate results.

(a) *Angular Error*

(b) *Bumpiness*

(c) *Smoothing*

Figure 4.3.: **Phenomena at Continuous Surfaces.** The gray lines represent simplified reference surfaces r. The red lines represent the corresponding surface estimates of an algorithm a.
We propose metrics to quantify that the algorithm surface a in (a) is perfectly smooth but rotated, whereas the other algorithm surfaces are bumpier (b) and smoother (c) than the reference surface r.

Phenomena. From the described application scenarios and algorithmic challenges, we derive three metrics to quantify algorithm performance at continuous surfaces: *Angular Error*, *Bumpiness*, and *Smoothing* (see Figure 4.3).

The *Angular Error* quantifies how well the estimated local surface orientation matches the reference surface orientation. The *Bumpiness* metric indicates if estimated surfaces are less smooth than the reference data (see Figure 4.3b), e.g. when discontinuities occur between local plane patches. The *Smoothing* metric measures the opposite effect. It quantifies how much smoother the algorithm result is compared to the reference data, i.e. how much detail of the surface structure is lost.

Region Definition. To quantify the described characteristics, we define $\mathcal{M}_c$ as the set of pixels at continuous disparity areas on the reference disparity map r. For a more specific evaluation, this set may be split into planar and non-planar continuous surface areas $\mathcal{M}_p$ and $\mathcal{M}_n$ (see blue and orange regions in Figure 4.2c).

These areas may be obtained in various ways. They can be derived automatically from synthetic data (as for our light field benchmark [44]), be annotated manually, or be created algorithmically from any dense reference data (see our experiments on the Middlebury dataset [46]). To identify continuous surfaces, second order derivatives are computed on the reference disparity map r. The derivatives are low on non-planar continuous surfaces and zero on planar continuous surfaces.

Continuous Surface Metric: Angular Error. To quantify the misorientation of the estimated surfaces, we compute the angular error between the local depth map surface normals of the reference data

and the algorithm under test (see normal maps in Figure 4.2d and Figure 4.4). With $\vec{n}_a(x)$ and $\vec{n}_r(x)$ denoting the estimated surface normals of the algorithm and reference depth map at pixel x, the average *Angular Error* is defined as:

$$\mathrm{AE}_{\mathcal{M}_c}(a, r) = \frac{1}{|\mathcal{M}_c|} \sum_{x \in \mathcal{M}_c} \sphericalangle(\vec{n}_a(x), \vec{n}_r(x)). \tag{4.1}$$

The *Angular Error* may be computed separately on planar and non-planar continuous surfaces M_p and $\mathcal{M}_n$.

Depending on the application, other statistics like the median angular error or a thresholded proportion similar to the *BadPix* metric (see Equation 3.2) may be applied. Examples for average *Angular Error* scores and visualizations of the respective normal maps and error maps are depicted on the third row of Figure 4.4.

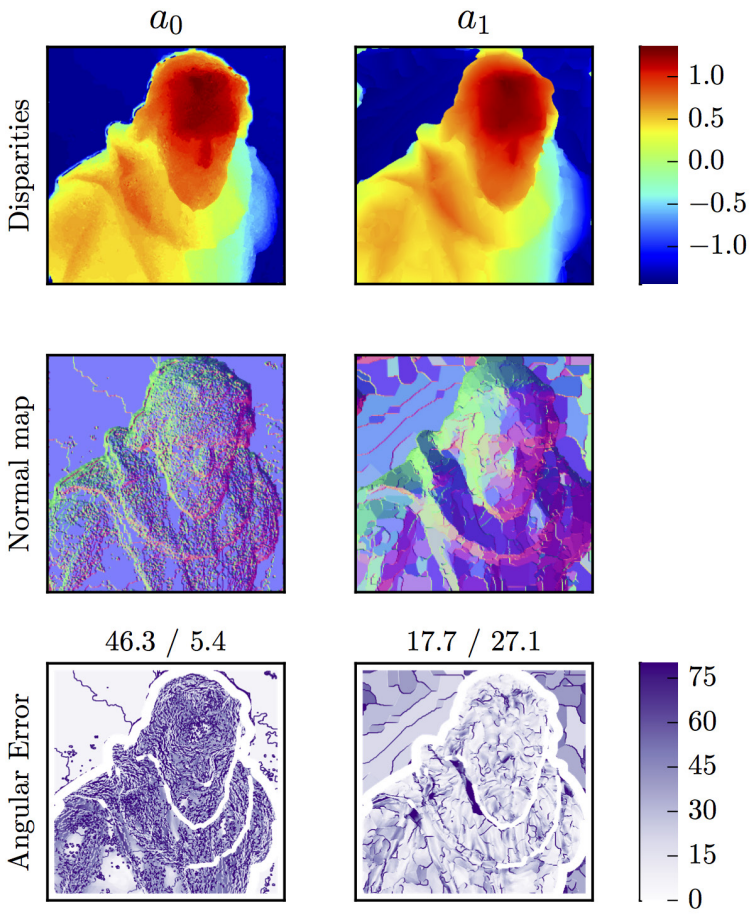


Figure 4.4.: ***Angular Error.*** Our metric scores and visualizations reveal differences between algorithms a_0 and a_1 which are hardly discernible on the disparity maps on the top row. The reference disparity map is depicted on Figure 4.2b.
The second row shows algorithm surface normals. The reference normal map is depicted on Figure 4.2d. The third row depicts per-pixel differences in degrees between the algorithm and reference normals. We separately compute *Angular Error* scores on non-planar/planar surfaces. Algorithm a_0 performs well on planes but produces noisy results on non-planar surfaces. a_1 estimates piecewise planar surface patches whose orientation is mostly similar to the reference orientation but offsets are present between the patches.

Continuous Surface Metric: Bumpiness. To quantify bumpiness, we compare the local curvature between the algorithm surface and the reference surface. For this purpose, we compute the Hessian matrix H for both, the algorithm and the reference disparity map, to quantify the second order disparity variations around each pixel:

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}. \tag{4.2}$$

The eigenvalues $\lambda_1(x)$ and $\lambda_2(x)$ of the Hessian matrix at pixel x denote the maximum and minimum value of the local curvature. Figure 4.2e depicts the signed maximum curvature per pixel of the disparity surface where darker blue and red areas indicate stronger positive and negative curvature. The relative values and signs of the two eigenvalues λ_1 and λ_2 classify local surface points into convex, concave, saddle like, or planar surface areas.

To quantify bumpiness, we focus on the magnitude of the curvature and omit its direction and sign. Let $c_a(x)$ and $c_r(x)$ denote the maximum absolute curvature at pixel x of the algorithm and reference disparity maps:

$$c(x) = max(|\lambda_1(x)|, |\lambda_2(x)|). \tag{4.3}$$

The algorithm reconstruction is then defined as being *bumpier* than the reference at pixel x if $c_a(x) > c_r(x)$. The average *Bumpiness* is quantified as:

$$\mathrm{B}_{\mathcal{M}_c}(a, r) = \frac{100}{|\mathcal{M}_c|} \sum_{x \in \mathcal{M}_c} max(0, c_a(x) - c_r(x)). \tag{4.4}$$

In accordance with Wanner et al. [148] and Scharstein and Szeliski [109], we apply a factor of 100 to the metric score. The *Bumpiness* metric solely focuses on the smoothness of an estimation. Misorientation or offset are quantified by other metrics.

Bumpiness scores and visualizations are illustrated on the second row of Figure 4.5. They reveal that algorithm a_2 yields considerable bumpiness while algorithm a_3 features a much smoother result.

Continuous Surface Metric: Smoothing. The *Smoothing* metric is defined analogously to the *Bumpiness* metric. It quantifies how much detail of the reference surface structure is lost by overly smooth algorithm results.

The algorithm reconstruction is defined to be *smoother* than the

reference data at pixel x if $c_a(x) < c_r(x)$. Hence, the average *Smoothing* is defined as:

$$\mathrm{S}_{\mathcal{M}_c}(a, r) = \frac{100}{|\mathcal{M}_c|} \sum_{x \in \mathcal{M}_c} max(0, c_r(x) - c_a(x)). \tag{4.5}$$

Smoothing scores and visualizations are illustrated on the third row of Figure 4.5. The visualization highlights that fine surface details on the ruff of the statue are lost by the surface reconstruction of a_3.

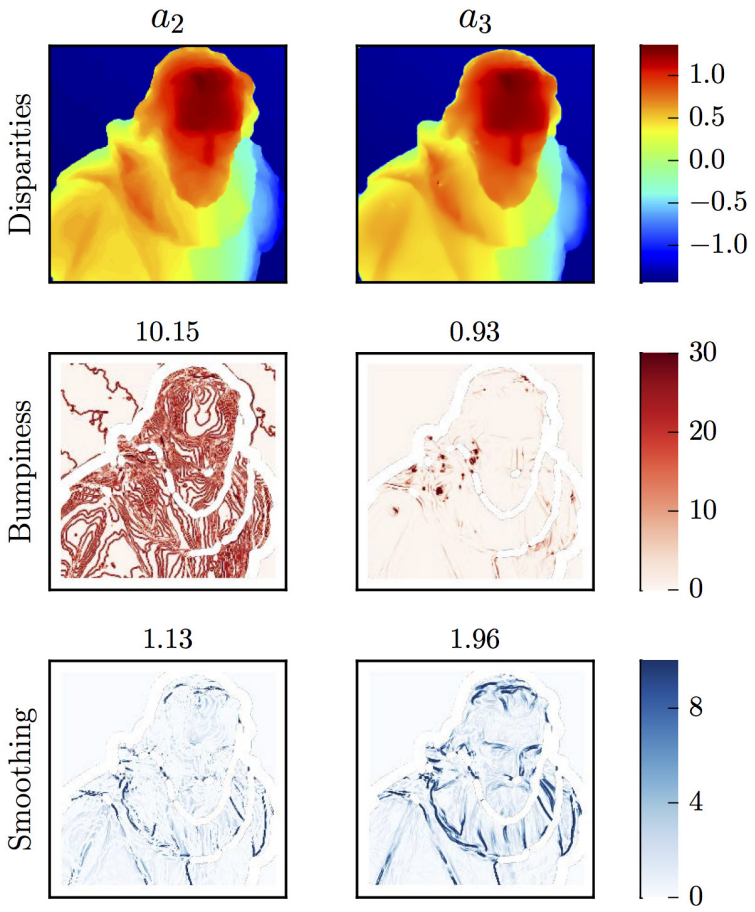
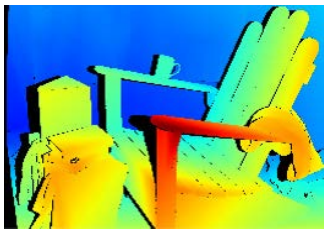


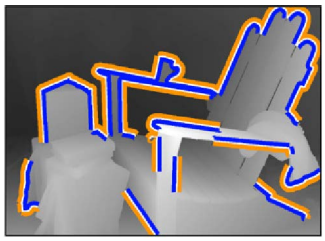
Figure 4.5.: ***Bumpiness* and *Smoothing*.** Our metrics and visualizations reveal differences between algorithms a_2 and a_3 which are hardly discernible on the disparity maps on the top row. The reference disparity map is depicted on Figure 4.2b.
The *Bumpiness* visualizations at the second row highlight that the discrete set of disparity estimates of algorithm a_2 causes severe *Bumpiness*. By contrast, the smooth result of a_3 leads to a much better *Bumpiness* score.
The *Smoothing* visualizations at the third row reveal that algorithm a_3 exhibits more smoothing than algorithm a_2. The dark blue areas indicate that many of the fine details of the ruff and hair are lost.

(a) Input image I

(b) Disparity map r



(c) Regions M_d, M_f, M_b



(d) Background map b

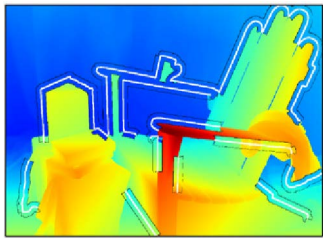


Figure 4.6.: **Evaluation at Discontinuities.** The input image (a) depicts the *Adirondack* scene of the Middlebury dataset [111].
To evaluate algorithm performance at discontinuities, we extract pixel sets (c) at depth discontinuities $\mathcal{M}_d$ (white), nearby foreground $\mathcal{M}_f$ (blue), and adjacent background $\mathcal{M}_b$ (orange). We further create extrapolated disparity maps b (d) where nearby background disparities are propagated into the foreground and vice versa.

4.2.2. Discontinuities

Definition. Discontinuities are defined as image regions where the disparity differences between adjacent pixels exceed a certain threshold. These *disparity edges* typically occur at object boundaries between foreground and background objects (see Figure 4.6).

Relevance. As described in Section 2.2.4, sharp and accurate disparity edges are important for applications such as object detection and tracking [43]. For video matting in the movie industry, artifacts occur if the boundary of the foreground object is estimated too wide. For fruit picking in automated agriculture, fruits may be damaged if the object boundary is estimated too narrow. In autonomous driving, inaccurate locations of object boundaries may lead to collisions or unnecessary emergency maneuvers if the object boundaries are estimated too narrow or too wide.

Algorithmic Challenges. Many of the commonly exploited assumptions as described in Section 2.2.3 are violated at depth discontinuity areas. Therefore, accurately estimating disparities at these areas is particularly challenging and error-prone.

Per definition, discontinuities violate the *continuity assumption*. The smoothness terms of global algorithms and the support windows of local algorithms work well on smooth disparity areas but are prone to mismatches and to overestimating the foreground at discontinuities. Furthermore, occluded image areas at discontinuities violate the *unique matching assumption*. Many algorithms perform left-right consistency checks to identify such areas. In order to produce dense disparity maps, algorithms apply various strategies to guess the disparities at semi-occluded regions by propagating surrounding non-occluded background disparities. Third, the *figural continuity assumption* is commonly used to define priors for occlusion boundaries [54]. This assumption is violated at depth discontinuities with low intensity contrast.

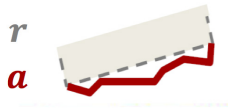
Phenomena. From the described application scenarios and algorithmic challenges, we derive metrics to quantify two phenomena at discontinuities: *Foreground Fattening* and *Foreground Thinning*.

Foreground Fattening occurs at object boundaries when the disparity estimates of the foreground object exceed its boundaries. Background pixels next to the objects are assigned the higher disparities of the foreground object rather than the lower disparities of the back-

ground, i.e. the object appears *fatter* than it actually is (see Figure 4.7a). This phenomenon is also coined edge fattening, surface overextension, or depth bleeding.

Analogously, *Foreground Thinning* occurs when disparity estimates near the object boundary are assigned background disparities rather than foreground disparities, i.e. the object appears *thinner* than it actually is (see Figure 4.7b).

(a) *Fattening*



(b) *Thinning*

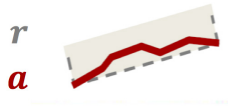


Figure 4.7.: **Phenomena at Discontinuities.** The dashed lines represent the reference boundaries r of the gray foreground object. The red lines represent the estimated object boundaries of an algorithm a.
We propose metrics to quantify that the estimated discontinuity a of the foreground object is fatter (a) or thinner (b) than the reference discontinuity r.

Region Definition. To quantify the described characteristics, we define $\mathcal{M}_d$ as the set of pixels at high gradients on the reference disparity map r (see white lines on Figure 4.6c). Furthermore, $\mathcal{M}_f$ and $\mathcal{M}_b$ denote the foreground and background areas on either side of the discontinuity (see blue and orange areas in Figure 4.6c).

These areas can be defined automatically from synthetic data (as for our light field benchmark [44]), derived from human annotations (as for our HCI benchmark [66]), or created algorithmically from any dense reference data (as for our experiments on the Middlebury dataset [46]).

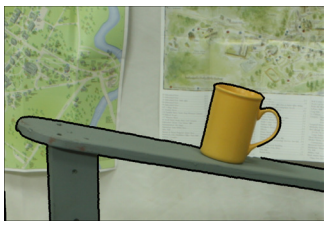
Discontinuity Metric: Foreground Fattening. We propose two ways to quantify how much the disparity estimates of foreground objects exceed their boundaries. The simple version *Foreground Fattening Simple (FFS)* is defined as a special variant of the *BadPix* metric which is applied to the set of background pixels $\mathcal{M}_b$:

$$\mathrm{FFS}_{\mathcal{M}_b}(a, r, t) = \frac{100}{|\mathcal{M}_b|} |\{x \in \mathcal{M}_b \colon (a(x) - r(x)) > t\}|. \tag{4.6}$$

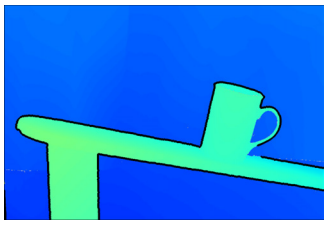
In contrast to the general *BadPix* (see Equation 3.2), only those disparity errors are taken into account where the background is mistakenly estimated to be closer than the reference. The second and third row of Figure 4.9a illustrate this difference. Algorithm a_0 produces a noisy disparity result with fattening and additional artifacts around the cup. The *BadPix* score penalizes both types of errors while our *FFS* score is focused on the fattening errors. $FFS_{\mathcal{M}_b}$ scores are equal to $BadPix_{\mathcal{M}_b}$ scores for algorithm results such as a_1 whose only disparity errors are caused by foreground fattening.

The simple metric variant can be computed directly from the algorithm and reference disparity maps but it depends on a threshold parameter t. We use $t = 6.0$ to quantify fattening on scenes from the Middlebury and comparable datasets. Setting an appropriate threshold can be challenging for more diverse datasets with high variation in disparity ranges.

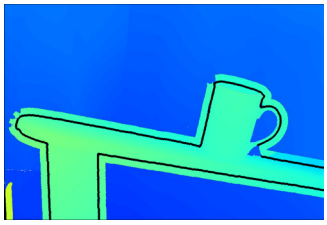
(a) Input image I



(b) Reference disparities r



(c) Foreground map f



(d) Background map b

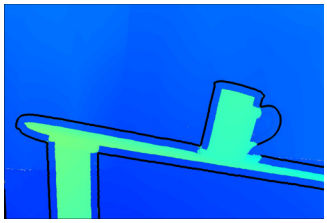


Figure 4.8.: **Disparity Extrapolation at Discontinuities.** Extrapolation is illustrated on a close-up of the *Adirondack* scene as depicted in Figure 4.6b. Black lines on the input image (a) and disparity map (b) indicate the location of the reference discontinuities.
To quantify fattening, we create the foreground map f by extrapolating the reference foreground disparities into the background (c). To quantify thinning, we create the background map b by extrapolating the reference background disparities into the foreground (d).

Our second variant of the *Foreground Fattening* metric *(FF)* does not depend on parameters. It uses extrapolated foreground maps f, as depicted in Figure 4.8c, to implicitly take local disparity range variation into account. The foreground map f is computed by linearly following local gradient directions on both sides of the discontinuity and by propagating disparities of $\mathcal{M}_f$ close to the discontinuity into $\mathcal{M}_b$. Additional details and examples are provided in our stereo metrics paper [46]. Depending on the scene content and data creation, f may also be created explicitly, as e.g. in the *Backgammon* scene of our light field dataset [44].

Based on the foreground map f, we quantify fattening by calculating the fraction of disparity estimates $a(x)$ at discontinuity background pixels $\mathcal{M}_b$ that are closer to the extrapolated foreground $f(x)$ than to the actual background $r(x)$:

$$\mathrm{FF}_{\mathcal{M}_b}(a, r) = \frac{100}{|\mathcal{M}_b|}\big|\{x \in \mathcal{M}_b \colon a(x) > h(x)\}\big|, \tag{4.7}$$

where $h(x) = 0.5 \cdot (f(x) + r(x))$ denotes the disparity halfway between the extrapolated foreground and the reference background. Metric scores $\mathrm{FF} \in [0, 100]$ are zero if no disparity estimate in $\mathcal{M}_b$ is closer to the foreground than to the background and 100 if all considered pixels mistakenly belong to the foreground.

For applications such as autonomous navigation, blocky fattening artifacts which reach far outside the true object boundary are more detrimental than moderate fattening artifacts which are evenly distributed around the object boundary. To incorporate this spatial dependency, all disparity estimates $x \in \mathcal{M}_b$ which are classified as fattening may be weighted by their distance to the closest discontinuity pixel in $\mathcal{M}_d$: $d(x, \mathcal{M}_d) = \min_{y \in \mathcal{M}_d} \|x - y\|$.

Discontinuity Metric: Foreground Thinning. Thinning is quantified analogously to *Foreground Fattening*. The simple version *Foreground Thinning Simple (FTS)* is defined as:

$$\mathrm{FTS}_{\mathcal{M}_f}(a, r, t) = \frac{100}{|\mathcal{M}_f|}\big|\{x \in \mathcal{M}_f \colon (r(x) - a(x)) > t\}\big|, \tag{4.8}$$

where $\mathcal{M}_f$ is the set of foreground pixels next to disparity discontinuities (see blue area in Figure 4.6c).

For the second variant, we use the extrapolated background map b as depicted in Figure 4.8d. We quantify *Foreground Thinning FT* by calculating the fraction of disparity estimates $a(x)$ at discontinuity foreground pixels $\mathcal{M}_f$ that are closer to the extrapolated back-

ground $b(x)$ than to the actual foreground $r(x)$:

$$\mathrm{FT}_{\mathcal{M}_f}(a, r) = \frac{100}{|\mathcal{M}_f|}\big|\{x \in \mathcal{M}_f \colon a(x) < h(x)\}\big|, \tag{4.9}$$

where $h(x) = 0.5 \cdot \big(b(x) + r(x)\big)$ denotes the disparity halfway between the extrapolated background and the reference foreground. The third and fourth row of Figure 4.9b show that the *FT* metric tends to be more tolerant to moderate disparity errors at discontinuities. Algorithm a_2 mistakenly estimates the lower part of the armrest to be slightly further away than it actually is. The error exceeds the chosen *BadPix* and *FTS* threshold of $t = 6$. Yet, the armrest is still clearly distinguished from the background. Depending on the application, $t = 6$ may not be an appropriate *FTS* threshold at this image region. Our more general metric variant *FT* penalizes only those disparity estimates which are closer to the background than to the foreground, independent of their actual distance. The same characteristics apply to our fattening metrics *FFS* and *FF*.

(a) *Foreground Fattening* (b) *Foreground Thinning*

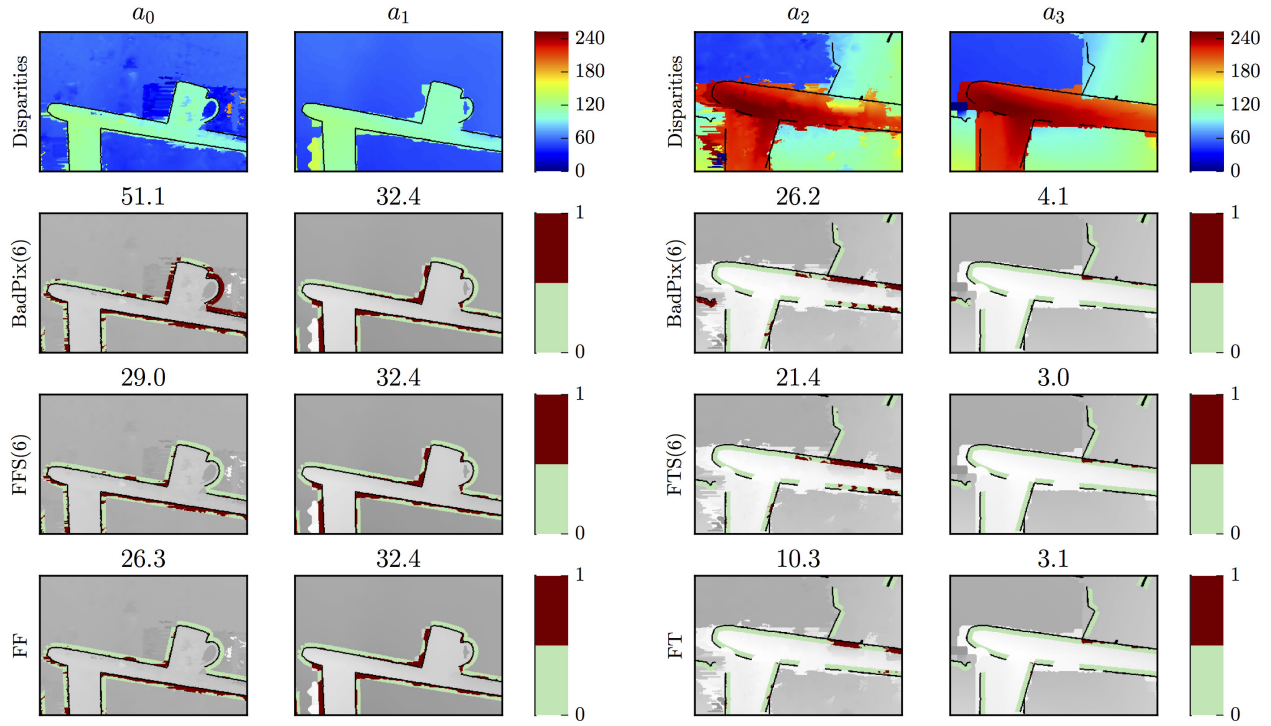
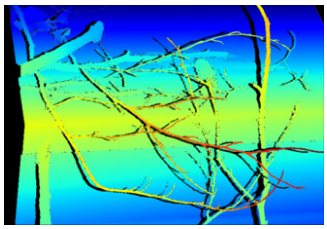


Figure 4.9.: **Metric Scores and Visualizations for *Foreground Fattening* and *Foreground Thinning*.** (a) The crops depict disparity and error maps of the rear armrest of the chair in Figure 4.6. The *FFS* score in the third row reveals that algorithm a_0 features less fattening than a_1 even though it has a much higher *BadPix* score. (b) The crops depict disparity errors and error maps of the closer armrest of the chair in Figure 4.6. The moderate disparity errors of algorithm a_2 at the lower part of the armrest are penalized by the *FTS* metric but not by *FT*.

(a) Input image I

(b) Disparity map r



(c) Regions M_r, M_s

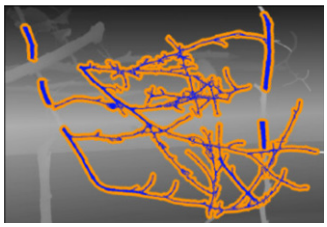


Figure 4.10.: **Evaluation at Fine Structures.** The input image (a) depicts the *Sticks* scene of the Middlebury dataset [111]. To evaluate algorithm performance at fine structures, we extract pixel sets (c) at fine structures $\mathcal{M}_r$ (blue) and adjacent background regions $\mathcal{M}_s$ (orange).

4.2.3. Fine Structures

Definition. Fine structures are defined as image regions where the geometry of foreground objects is comprised of thin, elongated structures of any orientation which are only a few pixels wide (see Figure 4.10). These structures often occur as part of grids.

Relevance. Metrics such as *BadPix* and *RMS* compute average scores of the entire image. They are forgiving with respect to errors at fine structures as these structures typically make up just a small fraction of the image. However, reconstructing such structures is crucial for many applications. Often, the spatial distribution of the reconstructed pixels is key whereas there is a certain tolerance about the exact distance to the structure.

In autonomous driving scenarios, thin objects like the bars of traffic signs or boom gates must be correctly detected to avoid serious accidents. For collision avoidance, the spatial distribution of detected fine structure pixels makes a big difference. A reconstruction which is severely fragmented but whose fragments are evenly distributed across the entire structure is preferable over a connected block which misses half of the structure. By contrast, such fragmented results can be a misleading input for applications like object recognition. For medical applications, the exact boundaries of thin instruments and threads must be known for accurate computer assisted surgery.

Algorithmic Challenges. Fine structures can be regarded as a particularly challenging case of general depth discontinuities. The same assumptions are violated as described for general discontinuities in Section 4.2.2. In addition, fine structures often violate the *ordering assumption*. When occurring as part of regular grids, the *unique appearance assumption* is violated in case of uniform backgrounds.

In the trade-off between minimizing artifacts and preserving fine structures, the latter are often sacrificed for smooth disparities at larger objects. Imperfect scale-space approaches and strong regularization tend to wipe out high frequency information such as fine structures.

Phenomena. From the described application scenarios and algorithmic challenges, we derive three metrics to specifically quantify algorithm performance at fine structures: *Porosity*, *Detail Fattening*, and *Fragmentation* (see Figure 4.11).

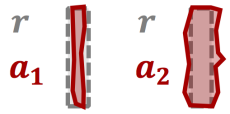
The *Porosity* metric quantifies how well fine structures are sampled. It is a measure of *fine structure sensitivity*. In Figure 4.11a, both algorithms correctly estimate the same number of pixels. Yet, the reconstruction of a_1 provides a much better sampling of the entire length of the structure. The large missing part of the structure on a_2 is penalized by the *Porosity* metric.

Complementary to *Porosity*, *Detail Fattening* is a measure of *fine structure specificity*. Analogously to *Foreground Fattening* (as introduced in Section 4.2.2), the *Detail Fattening* metric quantifies the amount of surface overextension at fine structures. *Detail Fattening* is particularly relevant for fine structures in a grid. Algorithms tend to either omit these fine structures or fill the grid holes with foreground disparities.

Fragmentation quantifies the coherence of reconstructed structures. In Figure 4.11c, both algorithms reconstruct the same number of pixels. Yet, algorithm a_1 produces a single component while algorithm a_2 produces three fragments.

(a) *Porosity*

(b) *Detail Fattening*



(c) *Fragmentation*

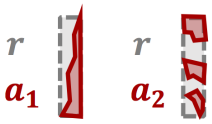


Figure 4.11.: **Phenomena at Fine Structures.** The gray areas represent the reference shapes r of the fine structures. The red areas represent the estimated fine structures of an algorithm a.
We propose metrics to quantify that the fine structure reconstructions of algorithm a_2 compared to their counterparts of a_1 feature a poorer sampling of the structure (a), yield more detail fattening (b), and are more fragmented (c).

Region Definition. To quantify the described characteristics, we define $\mathcal{M}_r$ as the set of fine structure pixels on the reference disparity map r. We further define $\mathcal{M}_s$ as the background pixels near these fine structures (see Figure 4.10c). Furthermore, $\mathcal{M}_a$ denotes all correctly estimated fine structure pixels of a given algorithm:

$$\mathcal{M}_a(t) = \{x \in \mathcal{M}_r : |a(x) - r(x)| \leq t\}. \tag{4.10}$$

Hence, the set of missing fine structure pixels for an algorithm a is $\mathcal{M}_r \setminus \mathcal{M}_a$. Fine structures can be defined automatically from synthetic data (as for our light field data [44]), derived from human annotations, or created algorithmically from any dense reference data (as for our experiments on the Middlebury dataset [46]). $\mathcal{M}_r$ can be obtained heuristically by shifting positive and negative gradients of the reference disparity map towards each other and by keeping regions with high overlap. Further details are provided in our paper and supplemental material [46].

Fine Structure Metric: Porosity. We quantify how well a fine structure is sampled by penalizing big missing parts of the structure. For each pixel $x \in \mathcal{M}_r$, we compute the logarithmic distance to the closest correctly estimated fine structure pixel $x \in \mathcal{M}_a$:

$$\mathrm{P}_{\mathcal{M}_a,\mathcal{M}_r}(a, r) = \frac{100}{|\mathcal{M}_r|} \sum_{x \in \mathcal{M}_r} ln(1 + d(x, \mathcal{M}_a)) \tag{4.11}$$

(a) Input image I

(b) Disparity map r

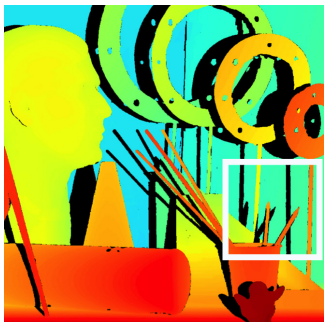


Figure 4.12.: **Scene with Fine Structures.** The input image (a) depicts the *Art* scene of the Middlebury dataset [111]. The white box on the disparity map (b) indicates the crop that is used to highlight performance differences at fine structures in Figure 4.13.

with $d(x, \mathcal{M}) = \min_{y \in \mathcal{M}} \|x - y\|$. Depending on the application, the actual pixel distance may be used instead of its logarithm.

We use a crop of the *Art* scene [42] as shown on Figure 4.12 to highlight performance differences at fine structures between different algorithms. *Porosity* scores and visualizations are depicted on the third row of Figure 4.13. Algorithm a_1 scores well on *Porosity*. Despite minor artifacts and jagged boundaries, the overall sampling of the fine structures is good. By contrast, the missing tip of the pen by a_0 and the big missing parts of the fine structures by algorithm a_2 are penalized by our metric.

Fine Structure Metric: Detail Fattening. Analogously to the foreground fattening metrics *FFS* and *FF*, we quantify the fattening at fine structures as depicted on Figure 4.11b. The simple variant *Detail Fattening Simple (DFS)* is defined as:

$$\mathrm{DFS}_{\mathcal{M}_s}(a, r, t) = \frac{100}{|\mathcal{M}_s|} \big|\{x \in \mathcal{M}_s \colon (a(x) - r(x)) > t\}\big|. \tag{4.12}$$

The metric variant *Detail Fattening (DF)* quantifies the extent to which background pixels next to fine structures are erroneously closer to the extrapolated foreground map f than to the background disparities in r. It is defined as:

$$\mathrm{DF}_{\mathcal{M}_s}(a, r) = \frac{100}{|\mathcal{M}_s|} \big|\{x \in \mathcal{M}_s \colon a(x) > h(x)\}\big|, \tag{4.13}$$

where $h(x) = 0.5 \cdot (r(x) + f(x))$. The foreground map f denotes the extrapolated foreground disparities of the fine structures $\mathcal{M}_r$.

Detail Fattening scores and visualizations are depicted on the fourth row of Figure 4.13. The algorithms a_0 and a_1 exhibit fattening at the left bar and between the bar and the pens. These phenomena are quantified by our metric and highlighted by our metric visualization.

Fine Structure Metric: Fragmentation. We quantify the fragmentation of $\mathcal{M}_a$ by counting all distinct 8-connected components per fine structure. Normalized by the number of reference structures, fragmentation is quantified as:

$$\mathrm{F}(a, \mathcal{S}) = \frac{100}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} 1 - \frac{1}{|\mathcal{F}_s|}, \tag{4.14}$$

where $\mathcal{S}$ is the set of reference structures and $\mathcal{F}_s$ is the set of algorithm fragments for each estimated structure $s \in \mathcal{S}$. The summand

is set to 1 for each structure with $|\mathcal{F}_s| = 0$, i.e. for fine structures with no estimated fragment at all. $F_{frag} \in [0, 100]$ is zero, if the algorithm produces a single component per structure and closer to 100 with an increasing number of fragments.

Fragmentation scores and visualizations are depicted on the fifth row of Figure 4.13. Algorithm a_0 produces coherent fine structure reconstructions leading to a perfect *Fragmentation* score while a_1 and a_2 exhibit moderate and severe fragmentation.

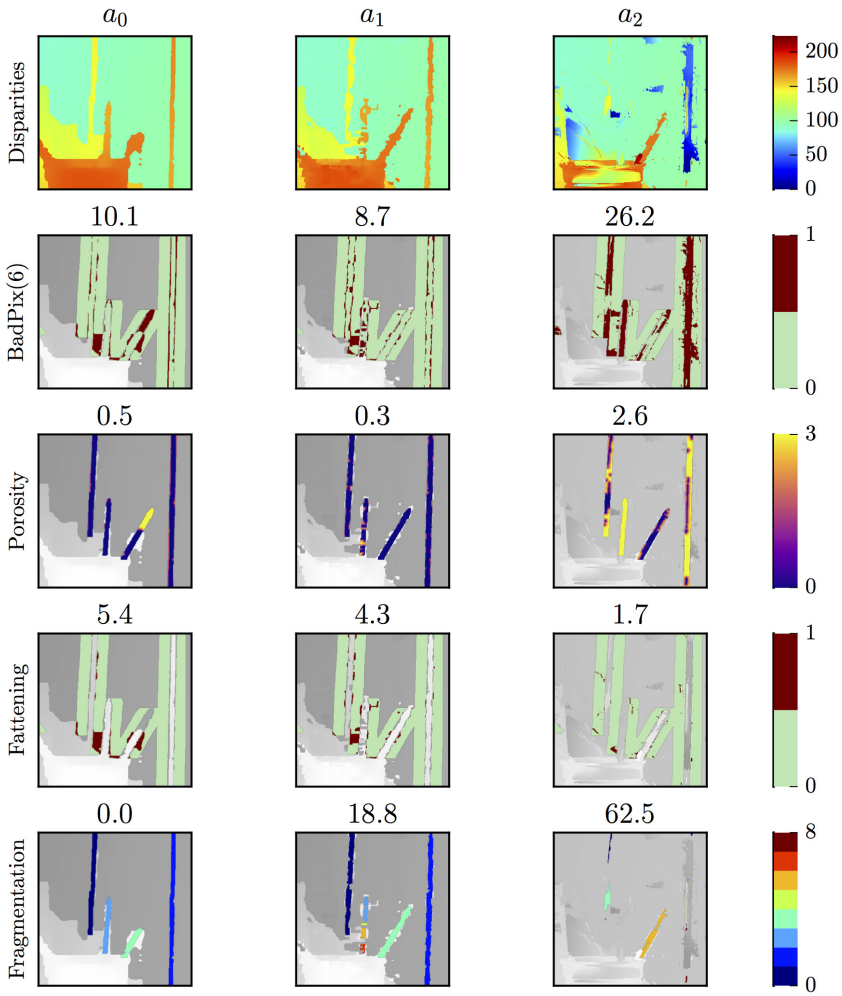


Figure 4.13.: ***Porosity*, *Detail Thinning*, and *Fragmentation*.** The top row depicts estimated disparity maps of the algorithms a_0, a_1, and a_3. The crops show pens and bars from the *Art* scene of the Middlebury dataset (see Figure 4.12).
Algorithm a_0 performs well at reconstructing the fine structures of the *Art* scene. The algorithm exhibits moderate *Detail Fattening* between the structures and some *Porosity* at the tip of the pen.
By contrast, a_2 exhibits severe *Porosity* and *Fragmentation* with almost no *Detail Fattening*.

4.3. Conclusion and Outlook

We proposed eight metrics to quantify and visualize performance at continuous surfaces, depth discontinuities, and fine structures. In Chapter 6, we thoroughly evaluate how well the proposed metrics meet the requirements as derived in Section 4.1. In our case study in Chapter 7, we demonstrate how to jointly apply our metrics to perform a comprehensive, multi-dimensional performance evaluation. In particular, we highlight the use of radar charts for visually comparing the relative performance of several algorithms with respect to multiple performance aspects simultaneously.

As discussed in Section 2.1, both, data and metric properties strongly affect the implementation and outcome of performance analyses. In the following chapter, we demonstrate how to apply our metrics in the presence of data deficiencies such as sparse algorithm results or weak reference data.

5
Test Data

The evaluation of stereo algorithms is strongly affected by both, performance metrics and evaluation data. Both aspects mutually influence each other and have considerable impact on the evaluation results. Ideally, evaluation datasets feature high quality reference data and perfectly representative scene content. Yet, reference data is often sparse, of limited accuracy and quantity, or even not available at all. Furthermore, the distribution of semantic scene content and algorithmic challenges on the input data rarely reflect the application priorities in an adequate way.

In this chapter, we propose two ways to address and overcome data deficiencies in performance evaluation. In Section 5.1, we demonstrate how to apply our metrics to sparse algorithm results and how to utilize weak reference data according to the taxonomy derived in Section 3.2. In Section 5.2, we show how considerate and puristic scene design with spatially increasing difficulty allows for specific evaluation of isolated algorithmic challenges.

5.1. Dealing with Data Deficiencies

As derived in Section 3.2, the characteristics of the available algorithm results and reference data considerably affect the applicability and expressiveness of performance metrics.

For our proposed metrics, two aspects are affected: the region extraction and the computation of the metric function. As described in Section 4.2, both can be performed automatically provided that the algorithm and reference data is dense. In the following, we briefly demonstrate how to apply our metrics in situations where such data is not available. We discuss how to evaluate sparse algorithm results and how to make use of weak reference data.

5.1.1. Sparse Algorithm Results

Many applications related to autonomous navigation rely on fast and sparse 3D reconstructions of the environment. For such appli-

cations, the spatial distribution of the depth estimates is of great importance. Entire objects may go unnoticed if the estimates are not evenly spread.

For sparse algorithm results, the same principles apply to our metrics as discussed for metrics of related work in Section 3.2.2. Missing values may be ignored or interpolated before our performance metrics are applied. For *Foreground Fattening*, *Foreground Thinning*, and *Detail Thinning*, missing values may either be ignored or penalized, just as for the general *BadPix* metric (compare *BadPix* definitions at Equations 3.2 and 3.3).

In addition to these metrics, *Sparsity* may be quantified as the percentage of missing pixels (see Equation 3.13). We propose spatially aware variants of the *Sparsity* metric and the *BadPix* metric. The proposed variants are based on a generalization of our *Porosity* metric (see Equation 4.11). Coherent areas with many missing disparity estimates are penalized more strongly as compared to areas where missing estimates are evenly distributed. The spatially aware sparsity metric *SS* does not depend on any reference data. Just like the general, location-agnostic *Sparsity* metric, our metric is applied on the full set of image pixels $\mathcal{M}$:

$$\mathrm{SS}_{\mathcal{M},\mathcal{M}_v}(a) = \frac{100}{|\mathcal{M}|} \sum_{x \in \mathcal{M}} ln(1 + d(x, \mathcal{M}_v)). \tag{5.1}$$

The set $\mathcal{M}_v = \{x \in \mathcal{M} \colon a(x) \in [d_{min}, d_{max}]\}$ denotes all pixels with valid algorithm disparity estimates, independent of their accuracy. Analogously, we propose a spatially aware *BadPix* measure $SBP_{\mathcal{M},\mathcal{M}_k}(a, r, t)$ where the set $\mathcal{M}_k = \{x \in \mathcal{M} \colon |a(x) - r(x)| < t\}$ denotes all valid algorithm disparity estimates with an absolute error below a given threshold t.

5.1.2. Weak Reference Data

Weak reference data, such as additional views or human feedback, has been used to quantitatively evaluate stereo algorithms in the absence of actual reference disparities (see Section 3.2.4 for details). We demonstrate how to apply our *Bumpiness* metric as well as the discontinuity metrics based on minimal human annotation.

Bumpiness. On Figure 5.1, we illustrate the quantification of local curvature. We adjust our *Bumpiness* metric to not require any reference disparities. Weak labels of image areas with mostly smooth geometries, denoted as $\mathcal{M}_p$, are sufficient to quantify and compare

the amount of local curvature on the algorithm results. This *Weak Bumpiness (WB)* metric is defined as:

$$\mathrm{WB}_{\mathcal{M}_p}(a) = \frac{100}{|\mathcal{M}_p|} \sum_{x \in \mathcal{M}_p} c_a(x), \tag{5.2}$$

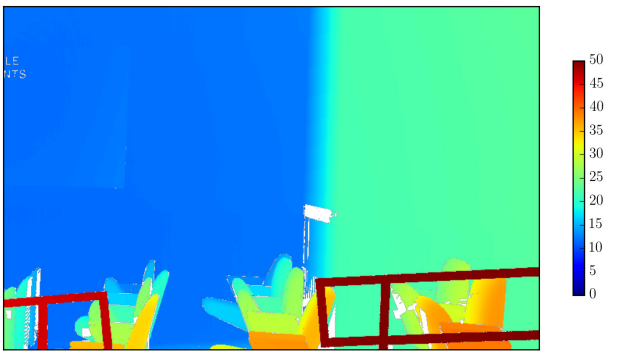
where $c_a(x)$ denotes the local curvature of the algorithm result as defined in Equation 4.3. For controlled environments such as industrial inspection, the same region labeling $\mathcal{M}_p$ may be used on multiple images.

The bumpiness scores on Figure 5.1 show that the smooth result of `ST-2` (see Figure 5.1d) and the bumpy result of `Elas` (see Figure 5.1h) are adequately quantified by our *WB* metric. Qualitatively, the bumpiness visualization on the second row of Figure 5.1 highlights that `ST-2` is affected by the shadows on the right wall. On the third row, the segmentation and plane patch fitting approach of `SPSS` is clearly visible.

Discontinuities. We further demonstrate how our discontinuity metrics are applicable to weak reference data. As depicted on the top row of Figure 5.2, the HCI dataset [66] provides input images with dynamic traffic scenarios. Reference disparity maps are limited to the static scenery because the 3D point cloud was captured separately from the images [65]. Manual annotation masks are provided for the dynamic objects to ignore these regions of missing reference data during evaluation (see Figure 5.2c). We utilize these masks to quantify algorithm performance at dynamic objects despite missing reference disparities. For each object, we create approximate cardboard disparities based on the median disparity of the respective ground region. These approximations (see Figure 5.2d) are sufficiently accurate to apply our *Fattening* and *Thinning* metrics as defined in Equations 4.7 and 4.9. As depicted on Figure 5.2f, the foreground fattening of the pedestrian is correctly identified. Our metrics implicitly adjust to local disparity ranges and penalize those disparity estimates which are closer to the extrapolated cardboard disparities than to the reference background.

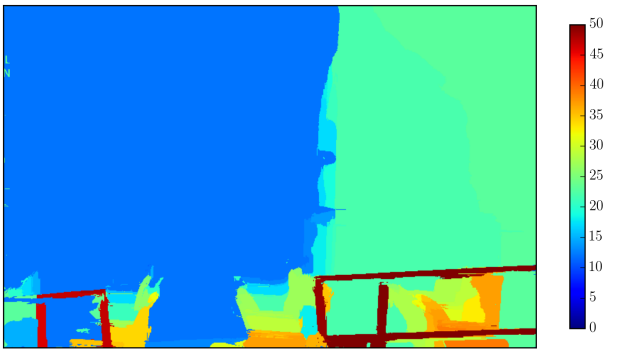
On Figure 5.3, we show additional examples of our cardboard generation. Approximating dynamic objects with cardboard disparities works well for most traffic situations. On the third row, the person lying on the ground at the bottom right of the image demonstrates a failure case. For such situations, slanted cardboards or a *Stixel* representation [2] would be required.

(a) Reference disparities

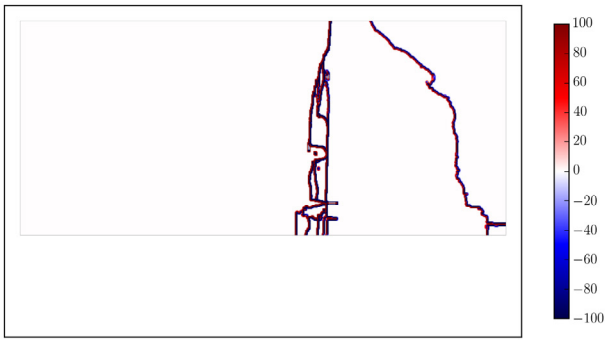


(b) Input image of the *Classroom* scene [111]

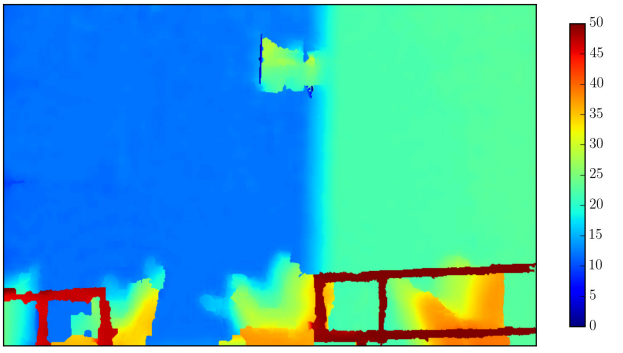
(c) Algorithm disparities `ST-2` [84]



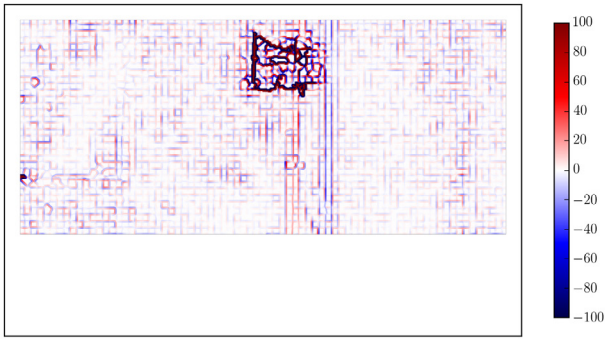
(d) Mean bumpiness `ST-2`: 7.9



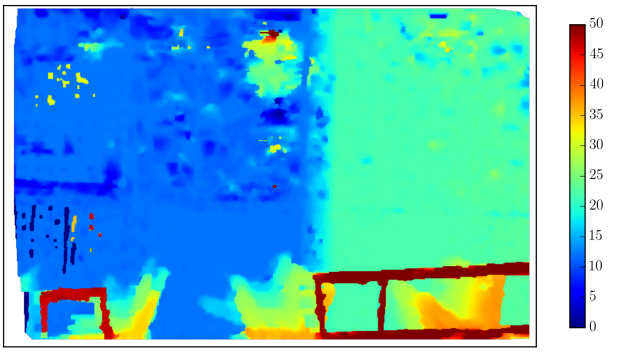
(e) Algorithm disparities `SPSS` [150]



(f) Mean bumpiness `SPSS`: 13.5



(g) Algorithm disparities `Elas` [29]



(h) Mean bumpiness `Elas`: 73.4

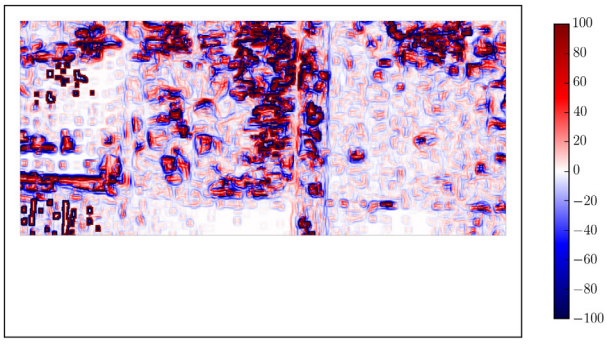


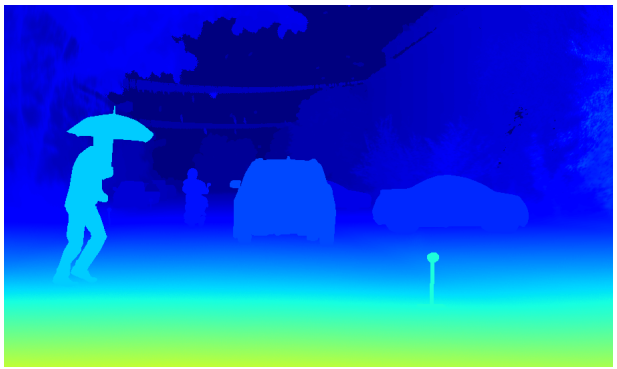
Figure 5.1.: **Algorithm Evaluation Based on Weak Region Annotations.** Our bumpiness metric *WB* as defined in Equation 5.2 does not depend on reference disparities. It quantifies the average local curvature on the algorithm disparity maps based on rough manual annotations of image regions with smooth geometry. The reference disparity map (a) is shown for comparison. It is not required for the evaluation.

(a) Input image

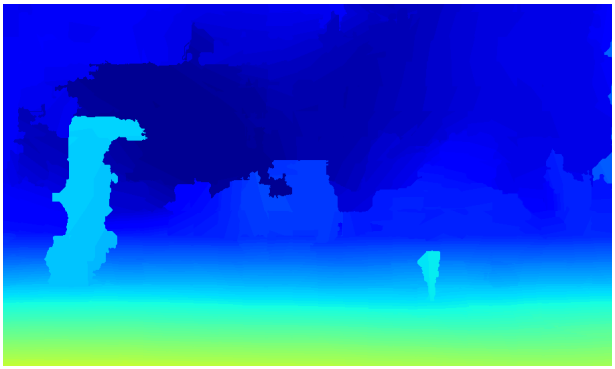
(b) Reference disparities (big disparities are green)

(c) Manual annotations

(d) Cardboard disparities



(e) Algorithm disparities



(f) Foreground fattening

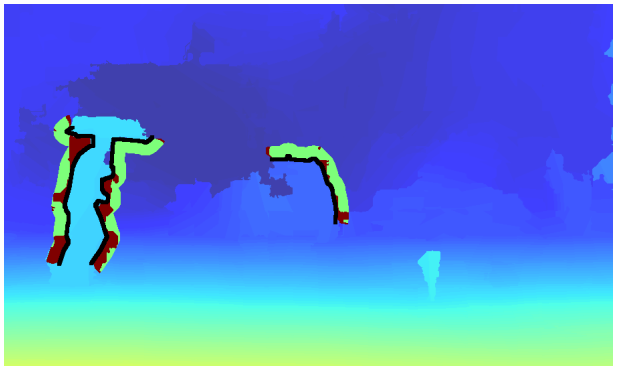


Figure 5.2.: **Algorithm Evaluation Based on Cardboard Disparities.** The HCI dataset [66] provides input images (a) and disparity maps for the static scenery (b). Manual annotations of the dynamic objects (c) are available for 3500 scenes. We utilize these annotations to create approximate cardboard disparities (d). Phenomena like *Foreground Fattening* (f) can be evaluated based on the cardboard disparities.

(a) Input images with annotations (b) Static reference disparities (c) Cardboard disparities

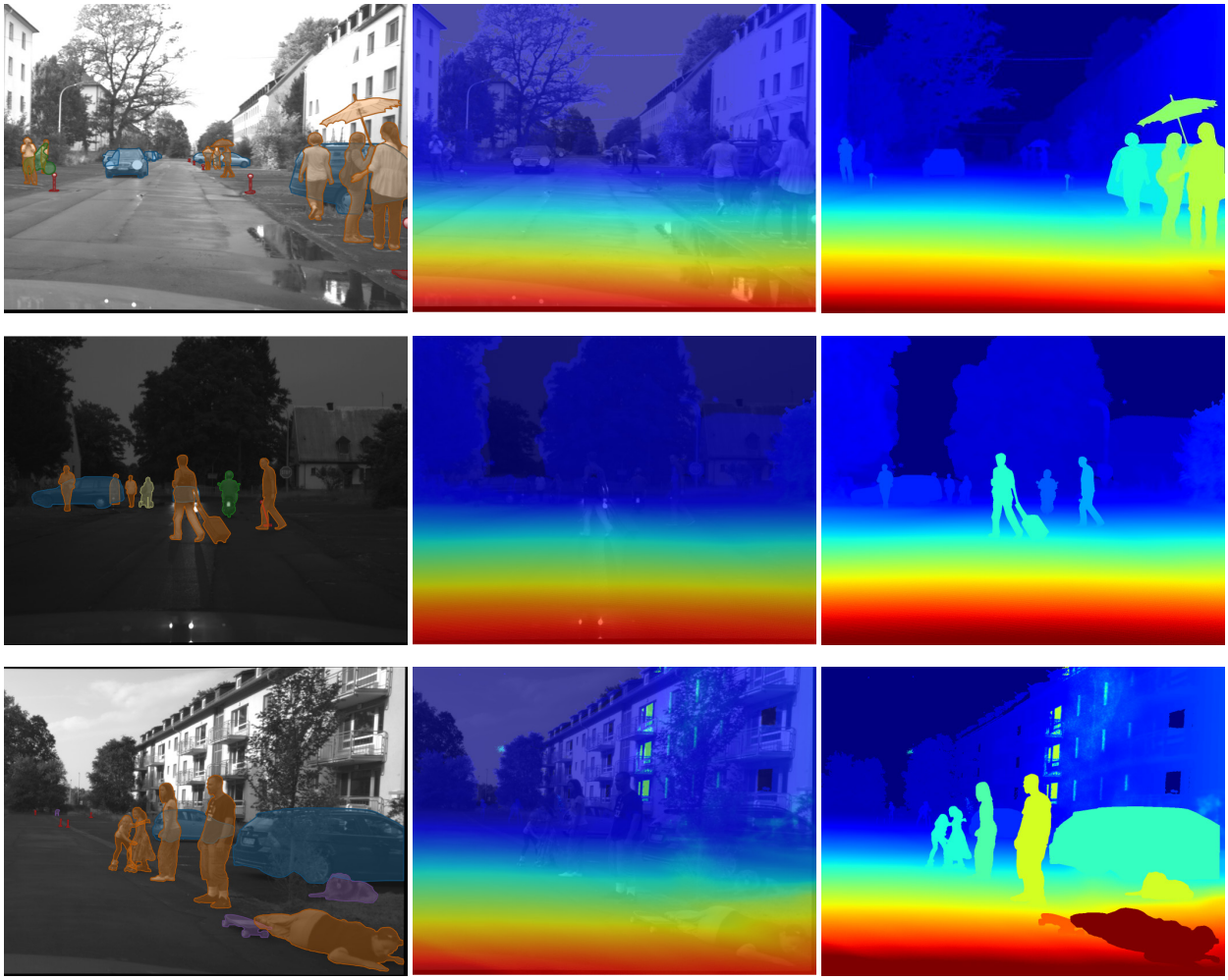


Figure 5.3.: **Further Examples of Cardboard Disparities.** Approximating dynamic objects with cardboard disparities works well for most traffic situations. For situations like the person lying on the ground (see third row), slanted cardboards or a *Stixel* representation [2] are required.

5.2. Stratified Scenes

As described in Section 2.1.2, it is difficult to draw specific and definite conclusions about algorithm performance from big datasets which have unknown distributions of algorithmic challenges. As a complementary type of evaluation data, we propose the concept of *stratified scenes* to systematically test for robustness and graceful degradation at algorithmic challenges. Analogously to our proposed metrics, these scenes are more specific than general datasets yet sufficiently broad to be relevant and applicable to various applications.

In this section, we first describe the underlying concepts of stratified scenes. Second, we derive design principles in the context of stereo and light field algorithms. Third, we propose four examples of stratified scenes which were designed jointly with Ole Johannsen as part of the *4D Light Field Benchmark* [44].

5.2.1. Concept

Stratified scenes are designed to pose specific, isolated algorithmic challenges. Scene content is varied gradually to create spatially increasing difficulty within the image. Simplistic synthetic rendering is typically sufficient for creating stratified scenes. As an example, the puristic *Backgammon* scene in Figure 5.4 poses specific algorithmic challenges related to depth discontinuities. These challenges increase with thinner peaks and narrower gaps along the vertical image axis.

The term "stratified" is freely adapted from the concept of stratified sampling that is often applied in statistical surveys [108]. It is particularly valuable if comparatively few samples can be collected. Instead of sampling from the full population, the population is first divided into more homogeneous, mutually exclusive subgroups. Random samples are then drawn proportionally from each subgroup. Incorporating prior knowledge for the division into appropriate subgroups allows for more meaningful results, reduces the risk of strong sampling skews, and ensures that all relevant subgroups are represented. In a similar way, our stratified scenes represent subgroups of algorithmic challenges, e.g. low texture or occlusions. By evaluating performance on representative examples of these groups, a solid understanding of algorithm performance can be obtained from a comparably small set of samples.

Our stratified scenes build upon the concept of synthetic scenes by Häusler and Kondermann [51]. The authors propose series of synthetic scenes related to decalibration, inconsistencies between views,

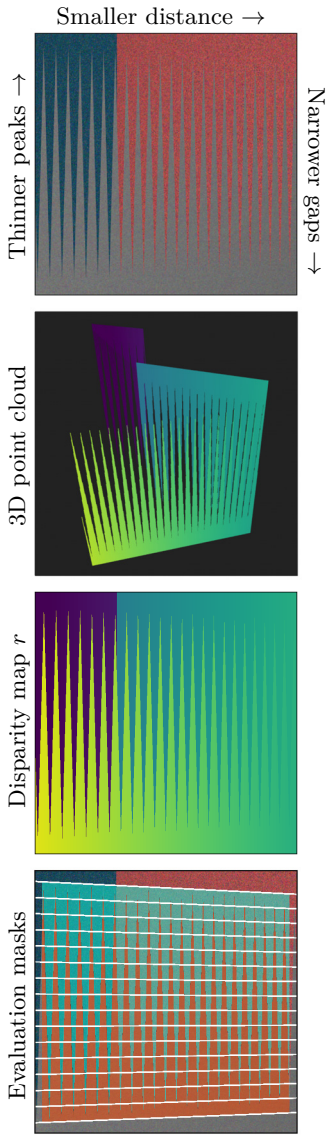


Figure 5.4.: **Evaluation at Thin Structures and Narrow Gaps.** From top to bottom, the input image, 3D point cloud visualization, disparity map, and evaluation mask of the *Backgammon* scene [44] are shown. This scene features spatially increasing challenges at depth discontinuities. On the disparity map, yellow indicates close-by geometry while violet indicates distant geometry. The evaluation masks in the bottom row allow for specific, quantitative analysis of performance degradation.

edge fattening, and signal to noise ratio. Instead of series of images, we propose scene designs with spatially increasing difficulty within the same image. Häusler and Kondermann aim at isolating core problems to "combinatorially iterate through the design space" and to create "large numbers of images at low cost" [51]. Instead of creating large numbers of scenes, we aim at minimizing the number of scenes which are required for deriving meaningful insights about algorithm accuracy and robustness.

The design of our scenes with spatially increasing difficulty allows for immediate visual inspection and easy comparison between algorithms. Furthermore, quantitative evaluation along the respective axes of difficulty reveals insights about robustness and graceful degradation.

5.2.2. Design Principles

When designing stratified scenes, the following aspects should be considered: Which challenges and which challenge combinations are relevant for the computer vision task and the evaluation objective at hand? How should the challenges be represented on the scene with varying levels of difficulty? Is the interference of any unrelated algorithmic challenges minimized?

For stereo and light field algorithms, the axes with increasingly difficult challenges are related to three categories: scene geometry, surface appearance, and the camera setup. Geometric challenges include objects with various surface orientations, occlusion configurations, or surface geometries such as low local curvature, edges, or corners. They further include different shapes and sizes such as tiny objects or holes, narrow slits, thin peaks, or various types of grid-like structures. Appearance based challenges may address the *photoconsistency assumption* or the *unique matching assumption* (see Section 2.2.3). This includes various types and levels of texture, noise, specularity, or transparency. On stereo and other multi-camera setups, inconsistencies between the views may be included such as imperfect calibration, differing sensor characteristics, or unequal focus levels.

A stratified scene may be designed to feature one or multiple challenges of the same or different challenge categories. It depends on the evaluation objectives which challenges and which combinations are most relevant. Spatially increasing difficulty may be represented by a set of discrete instances with different properties or by one instance with continuously changing properties. It is crucial for a decoupled

and meaningful evaluation that the impact of other effects is minimized. For instance, a stratified scene addressing purely geometric challenges should feature good texture and contrast.

Similar challenge categories can be identified for other computer vision tasks such as optical flow or tracking. In the following section, we present and discuss four examples which are related to geometric and appearance based challenges of stereo and light field algorithms. Additional stratified scenes may be created based on the same design principles.

5.2.3. Examples

In this section, we apply the previously introduced design principles to derive four stratified scenes. For each scene, we explain the evaluation purpose, the resulting scene design, and the evaluation axes for qualitative visual inspection and quantitative degradation analysis. We perform an experimental algorithm evaluation for each scene in Section 6.3 to test the applicability and usefulness of the proposed scenes. The scenes were created together with Ole Johannsen. They are widely used in our *4D Light Field Benchmark* [44]. Additional details are provided in the respective publication.

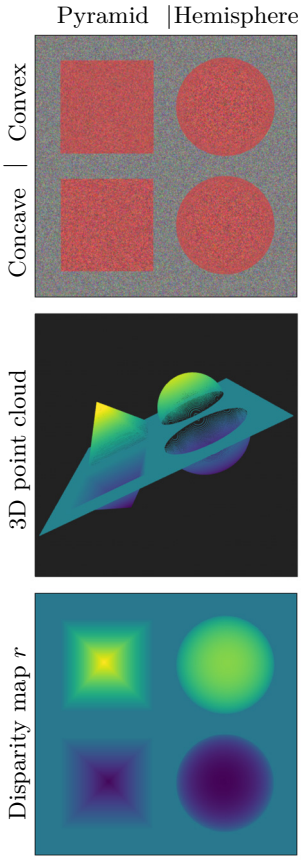


Figure 5.5.: **Evaluation at Planar and Non-Planar Surfaces.** From top to bottom, the input image, 3D point cloud visualization, and disparity map of the *Pyramids* scene [44] are shown.
The setup of the *Pyramids* scene challenges algorithms at reconstructing continuous, planar and non-planar as well as convex and concave surfaces at different orientations.

Backgammon: Thin Structures and Narrow Gaps

The *Backgammon* scene is designed to assess performance at purely geometric challenges: thin structures and narrow gaps at varying depth discontinuities. As depicted on the top row of Figure 5.4, the scene consists of two slanted background planes (red and blue) and a jagged, inversely slanted foreground plane (gray). In order to focus on the geometric challenges, all surfaces feature a regular texture and sufficient contrast between the geometric entities. From top to bottom, the gaps between the peaks get narrower. From bottom to top, the foreground plane features increasingly thinner peaks. As an additional evaluation aspect, the background planes are slanted such that the disparity differences and hence the occlusion areas increase from right to left (see disparity map visualizations on the second and third row of Figure 5.4).

Depending on their occlusion handling and their regularization techniques, algorithms tend to miss thin structures and to interpolate foreground geometry between narrow gaps. This behavior can be inspected visually when comparing algorithm results on the *Backgammon* scene.

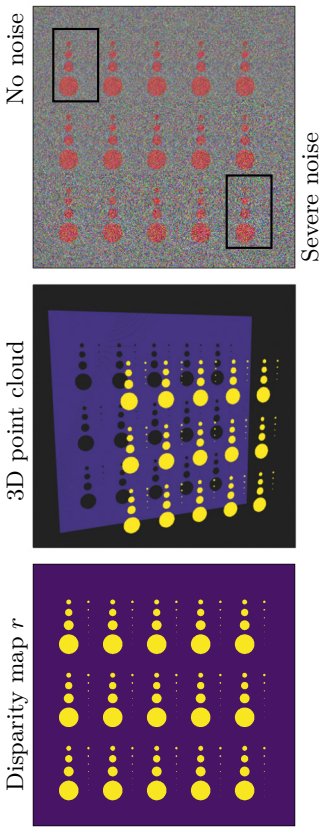


Figure 5.6.: **Evaluation of Tiny Objects under the Impact of Noise.** From top to bottom, the input image, 3D point cloud visualization, and disparity map of the *Dots* scene [44] are shown.
The *Dots* scene is based on a grid of three rows with five cells each. Each cell contains two columns with increasingly bigger dots. To support degradation analysis, the level of noise increases in row-major order from the top left to the bottom right cell (see black frames).

For a quantitative analysis, we apply evaluation masks as depicted on the fourth row of Figure 5.4. We restrict the evaluation to the cyan or orange areas to quantify performance at the gaps or peaks respectively. In addition, we quantify performance degradation along the vertical axis of the scene. Reconstructing the gaps gets more challenging from the top to the bottom of the image. We quantify *Foreground Fattening* separately on increasingly challenging horizontal image regions as separated by the white horizontal lines.

Pyramids: Planar and Non-Planar Surfaces

The *Pyramids* scene is focused on assessing performance at different geometric challenges: slanted planar and non-planar surfaces. As depicted in Figure 5.5, the scene consists of a fronto-parallel base plane, two pyramids, and two hemispheres. The upper hemisphere and pyramid are convex; they stick out of the base plane. The lower hemisphere and pyramid are concave.

Depth estimation algorithms operate in a discrete or continuous disparity space and apply different types of refinement. These algorithm interna may cause staircasing effects or a better reconstruction of planar as compared to non-planar surfaces. Such differences in algorithm performance can be inspected visually on the *Pyramids* scene. For a quantitative comparison, we apply the *Bumpiness* and *Angular Error* metrics, as introduced in Section 4.2.1, on the individual geometric entities.

Dots: Noise and Tiny Objects

The *Dots* scene is designed to assess the interplay of geometric and appearance related challenges: sensitivity to small objects and robustness to noise. As depicted in Figure 5.6, the scene consists of 5×3 cells which are placed on a regular grid. Each cell features the same geometry, a fronto-parallel background plane with a range of increasingly smaller circles in the foreground. Increasing levels of noise are present in row-major order from the top left to the bottom right cell. We approximate thermal and shot noise by applying Gaussian noise with variances between 0.0 and 0.2 [44].

As described in Section 2.2, strong regularization makes algorithms more robust to noise but also more prone to missing small objects. The performance of handling this trade-off can be inspected visually on algorithm results for the *Dots* scene.

For a more quantitative analysis, we compute the accuracy of the background and the number of missing dots. We analyze how

increasing noise levels affect algorithm performance by computing these scores for each of the 15 cells separately [44].

Stripes: Texture and Contrast at Occlusion Regions

The *Stripes* scene is designed to assess the interplay of geometric and appearance related challenges: the impact of texture and contrast at occlusion boundaries. As depicted in Figure 5.7, the scene consists of a gray fronto-parallel background plane and 17 coplanar vertical stripes in the foreground. The stripes yield alternating dark and bright intensity, resulting in high and low contrast to the background. The top row of Figure 5.7 highlights that the low contrast stripes are hard to distinguish from the background. The level of texture increases from the bottom to the top on the stripes and from left to right on the background.

As described in Section 2.2.3, algorithms rely on a minimum level of texture to confidently match correspondences. Many algorithms use the *figural continuity assumption*: texture discontinuities correspond to geometric discontinuities and vice versa. Performance in the presence of these challenges can be inspected visually when comparing algorithm results on the *Stripes* scene.

For a more quantitative analysis, we define three evaluation masks to explicitly compute performance at non-occluded low texture areas, high contrast occlusion areas (see fourth row on Figure 5.7), and low contrast occlusion areas (see fifth row on Figure 5.7). To evaluate performance degradation, we quantify the reconstruction accuracy of the stripes at horizontal bins with decreasing levels of texture, similar to the bins on *Backgammon* (compare fourth row of Figure 5.4).

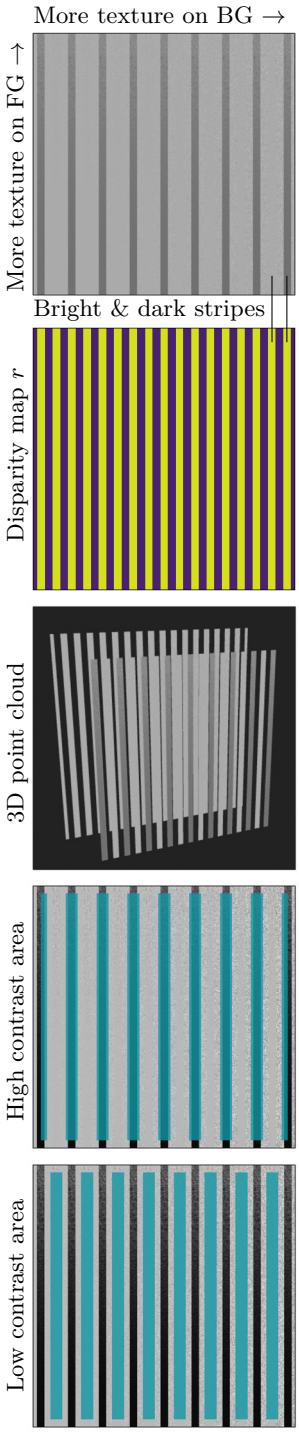


Figure 5.7.: **Evaluation at Occlusion Regions with Low Texture and Contrast.** The *Stripes* scene [44] consists of alternating dark and bright stripes on top of a bright background. Background texture is low and foreground texture is high on the top left corner. This relation is reversed towards the bottom right corner.

5.3. Conclusion and Outlook

We proposed two ways to conduct performance evaluation despite reference data deficiencies. First, we described how to apply our metrics to sparse algorithm results and weak reference data. Second, we introduced the concept of stratified scenes for the systematic evaluation of algorithm robustness. We presented four examples of stratified scenes which feature combinations of geometric and radiometric challenges.

In Section 6.3, we perform an experimental algorithm evaluation to assess the applicability and expressiveness of the proposed scenes. For each scene, we test whether it allows for insightful visual algorithm comparison as well as quantitative degradation analysis.

6

Experiments and Results

We perform a threefold evaluation of our proposed performance metrics and stratified scenes. An additional validation is presented by means of our case study in Chapter 7.

In Section 6.1, we test the specificity and complementary expressiveness of our proposed metrics. We derive systematic artificial test cases to assess how well our metric scores and visualizations meet the requirements R1 and R2 as specified in Section 4.1. In Section 6.2, we perform a user study to test whether our metrics are semantically meaningful and conform with visual performance assessment as specified by requirement R3. In Section 6.3, we evaluate the expressiveness of our stratified scenes. We refer to our case study in Chapter 7 for an evaluation of the applicability and customizability of our metrics according to requirements R4 and R5.

6.1. Evaluation with Systematic Test Cases

As discussed in Section 4.1, our novel metrics should provide an adequate profile of algorithm performance (R1) which complements the expressiveness of existing metrics (R2). We derive systematic deviations of algorithm results at continuous surfaces, discontinuities, and fine structures to evaluate the satisfaction of these requirements.

6.1.1. Experimental Setup

In order to test for requirement R1, we assess whether our metrics adequately quantify the specific aspects of algorithm performance as represented by systematic test cases. To test for requirement R2, we assess whether the geometry-awareness of our metrics allows for insights into algorithm performance which are not quantified by the prevalent *RMS* and *BadPix* metrics. Figures 6.1, 6.2, and 6.3 feature test cases for the three geometric entities. Each column represents one test case with a synthetic disparity map, metric visualizations, scores, and ranks. In case of ties, the minimum rank is assigned to the affected algorithms which is indicated by underlined ranks.

6.1.2. Continuous Surfaces

To evaluate our proposed metrics at continuous surfaces, we define a smooth, rotated plane with a small bump at the lower right corner as our reference disparity map r. As deviations, we add surface deformations, offsets, rotations, and artifacts. The first two rows on Figure 6.1 depict disparity maps and normal maps of the reference r and the synthetic algorithm results $a_0 - a_5$. The remaining rows depict scores, relative ranks, and visualizations of the proposed *Angular Error*, *Bumpiness*, and *Smoothing* metrics as well as the general *RMS* and *BadPix(1.0)* metrics.

Smoothing of surface details as in the artificial test algorithm a_0 commonly occurs when strong regularization is applied. Deviations as in $a_1 - a_3$ are typically due to piecewise planar surface fitting. a_1 represents a blocky result without additional refinement. On a_2, the left plane estimate is not correctly oriented. On a_3, the left plane estimate is correctly oriented but misplaced. a_4 and a_5 represent algorithms with artifacts: a_4 has additional bumps while a_5 produces a noisy surface.

Angular Error. The *Angular Error* scores in the third row of Figure 6.1 reflect the increasing proportion of incorrect surface orientations of the fold in a_2, the blocky surface in a_1, and the noisy artifacts in a_5. As intended, the *Angular Error* does not penalize the translated but correctly oriented planar surface on the left in a_3.

Bumpiness. The perfect *Bumpiness* score of a_0 adequately reflects that there is no additional curvature on the smoothened disparity map. As intended, the *Bumpiness* metric does not penalize the smooth but tilted surface of a_2 or the smooth but transposed surface of a_3. As illustrated by the metric visualizations, it is only the fold in a_2 and the gap in a_3 that cause moderate bumpiness scores on these algorithm results. In a_5, the heavy artifacts of the noisy result are adequately quantified as being severely bumpy.

Smoothing. Similar to *Bumpiness*, the *Smoothing* metric is tolerant towards the misorientation in a_2 and the offset in a_3. It adequately penalizes the smoothing of the bump in a_0 and a_5 as well as on the planar regions of the blocky result in a_1.

RMS* and *BadPix. *RMS* scores are identical for the synthetic algorithms even though their disparity and normal maps differ considerably (see row six on Figure 6.1). By contrast, the *Angular Error*, *Bumpiness*, and *Smoothing* scores reflect the performance differences more precisely. Thereby, they provide a more comprehensive performance profile and support decisions on algorithm se-

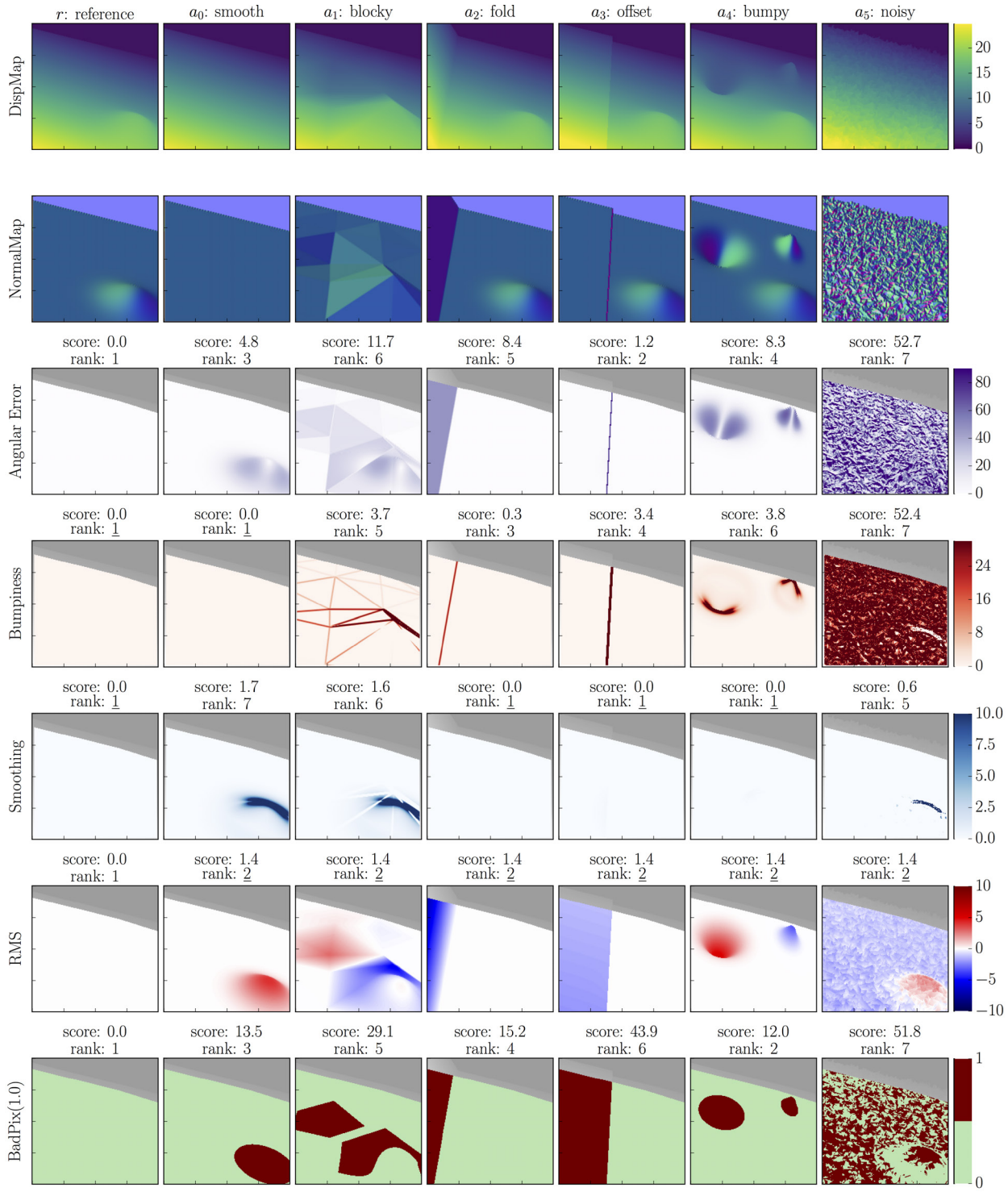


Figure 6.1.: **Systematic Test Cases for Continuous Surface Metrics.** The first two rows depict disparity maps and normal maps of the reference r and the synthetic algorithm results $a_0 - a_5$. The test algorithms are designed to represent isolated failure cases of surface reconstructions that are common in real algorithm data.
The remaining rows depict scores, relative ranks, and visualizations of the proposed surface metrics as well as the general *RMS* and *BadPix(1.0)* metrics. The scores of the *Angular Error*, *Bumpiness*, and *Smoothing* metrics accurately quantify the considerable performance differences between the algorithms $a_0 - a_5$ whereas *RMS* scores are identical for all algorithms.

lection. *BadPix* scores differ between the synthetic algorithms and define a distinct ranking. The offset in a_3 and the noise in a_5 heavily impair *BadPix* performance. However, by only taking into account *BadPix* scores without additional visualizations, it is difficult to tell which algorithm to choose for which application. For general 3D reconstruction, the relatively low *BadPix* scores of a_4 and a_0 indicate their suitability for this application. By contrast, for accurate shading, a_3 may be preferred over a_2 even though it has a much higher *BadPix* score. The scores of our semantically meaningful and more specific metrics allow for easier interpretation of the evaluation results as compared to the general *BadPix* scores.

6.1.3. Depth Discontinuities

We evaluate our proposed discontinuity metrics on a vertical depth discontinuity along the image center with a foreground object to the left (green) and background scenery to the right (blue) as illustrated by the reference disparity map r in Figure 6.2. As deviations, we add varying levels of over- and under-estimated pixels together with different line shapes (see synthetic algorithms $a_0 - a_5$ in Figure 6.2). On a_0, artifacts are present at both sides of the discontinuity. a_1 produces a crisp vertical discontinuity but the foreground object on the left is smaller than it should be. By contrast, on a_2, the foreground object is bigger than it should be. On a_3, the lower part of the discontinuity is accurate while the upper part severely overestimates the foreground. a_4 produces a fuzzy discontinuity which is approximately at the right location while a_5 is rotated.

Foreground Fattening. The algorithms a_2 and a_3 severely overestimate the foreground object which is accurately reflected by our *Fattening* scores and visualizations (see second row of Figure 6.2). Half of the evaluation region features over-estimated foreground disparities (green) instead of background disparities (blue), resulting in a *Fattening* score of 50%. The third row on Figure 6.2 depicts the distance-weighted variant of our metric where fattening is penalized more strongly if it occurs further away from the reference edge. As intended, this variant assigns a lower rank to a_3 as compared to a_2.

The fuzzy results of a_4 and the rotated result of a_5 exhibit limited fattening which is reflected by moderate *Fattening* scores. As intended, the artifacts on a_0 and the thinning of a_1 are not penalized by our *Fattening* metric. The lower artifact on a_0 denotes incorrect disparities behind the reference background. As explained in Section 4.2.2, only those disparity errors are considered for foreground

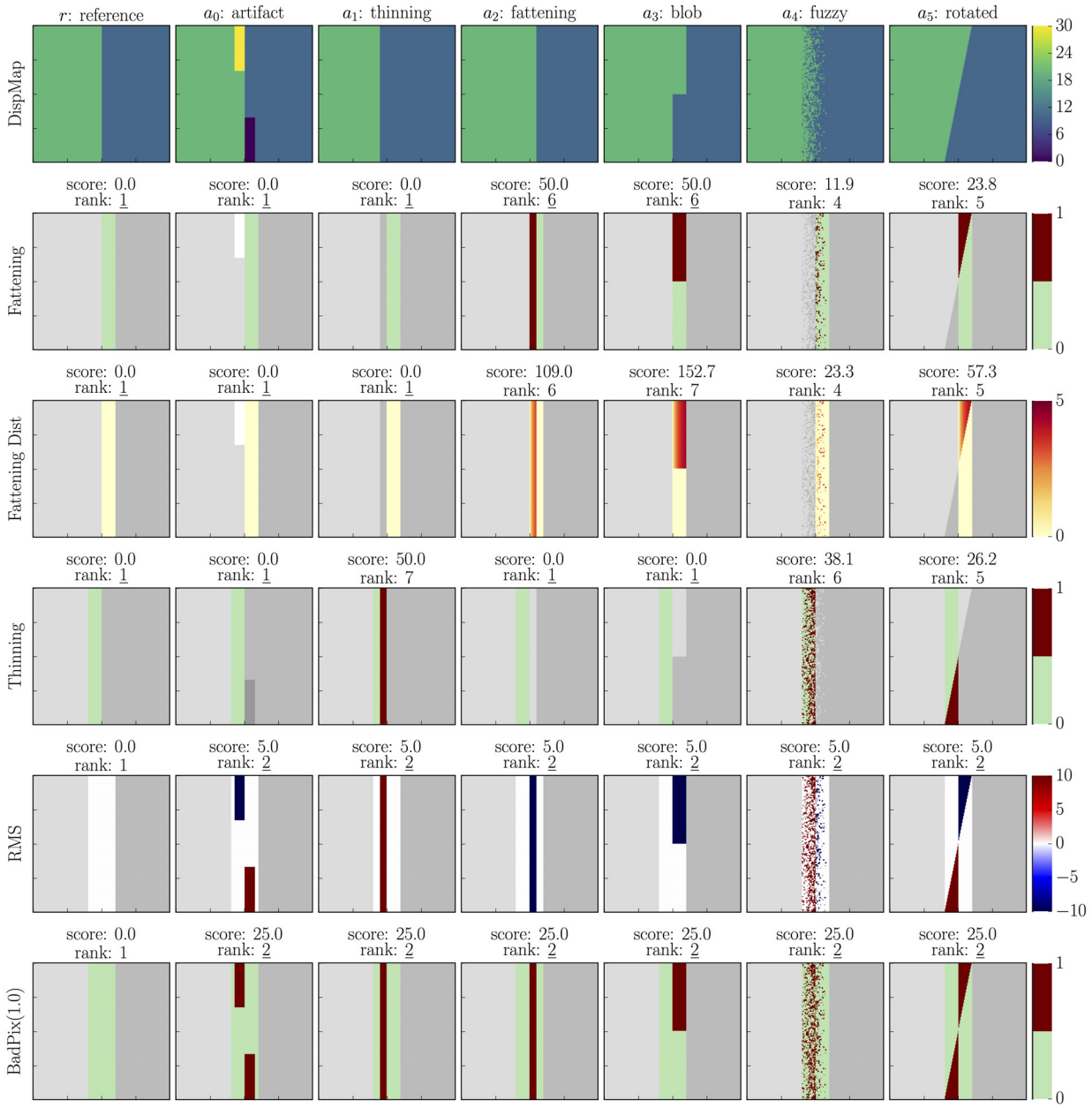


Figure 6.2.: **Systematic Test Cases for Discontinuity Metrics.** The top row depicts disparity maps of the reference r and the synthetic algorithm results $a_0 - a_5$. The green area on the left denotes a foreground object in front of the blue background area. The test algorithms are designed to represent isolated failure cases at object boundaries that are common in real algorithm data.
The test cases are designed such that *RMS* and *BadPix* scores are identical for all algorithms even though the shapes of the reconstructed discontinuities differ greatly. As depicted on rows two to four, our *Foreground Fattening* and *Foreground Thinning* metrics adequately quantify the performance differences.

fattening which are in front of the reference background.

Foreground Thinning. Analogously to a_2 on *Fattening*, a_1 scores poorly on the *Thinning* metric because it under-estimates the foreground object and produces background disparity estimates instead. The fuzzy algorithm a_4 scores worse on *Thinning* than on *Fattening* as it lacks more pixels on the green foreground than on the blue background. The upper artifact on a_0 is ignored by the *Thinning* metric since the erroneously estimated disparities are in front of the reference foreground instead of behind.

RMS* and *BadPix. All algorithms have identical *RMS* and *BadPix* scores even though their discontinuity estimates are very different. By contrast, the *Fattening* and *Thinning* metric do reflect the performance differences. For applications like obstacle detection, algorithms with severe thinning might cause dangerous situations. By contrast, severe fattening causes unwanted artifacts for applications such as image matting in visual effects. Our specific and semantically intuitive metrics support algorithm selection for such scenarios.

6.1.4. Fine Structures

We evaluate our proposed fine structure metrics on a thin, vertical bar (yellow) in front of planar background (purple) as illustrated by the reference disparity map r in Figure 6.3. As deviations, we add varying levels of over- and under-estimated pixels together with varying degrees of fragmentation (see algorithms $a_0 - a_5$ in Figure 6.3). The artificial algorithms a_0 and a_1 both reconstruct the bar as a single structure which is thinner than the reference bar. The reconstruction of a_0 is slightly too small on both sides of the structure while a_1 is missing structure on the right. a_2 and a_3 represent fragmented reconstructions of the fine structure. This is a common flaw when fine structures yield little contrast to the background texture and when strong regularization is applied. a_4 and a_5 represent noisy results with many small artifacts on a_5 and bigger artifacts on a_4.

Porosity. The *Porosity* metric is designed to quantify how well a structure is sampled. On a_0 and a_1, the same amount and shape of the fine structure is reconstructed. Yet, the distance to the actual boundary of the fine structure is much bigger on a_1 than on a_0, resulting in a higher *Porosity* score for a_1 (see second row on Figure 6.3). The rather tiny holes on a_4 and a_5 lead to moderate *Porosity* scores since the sampling of the structure is still good. By contrast, substantial parts of the structure are missing in a_2 and a_3 which is adequately reflected by high *Porosity* scores. Overall,

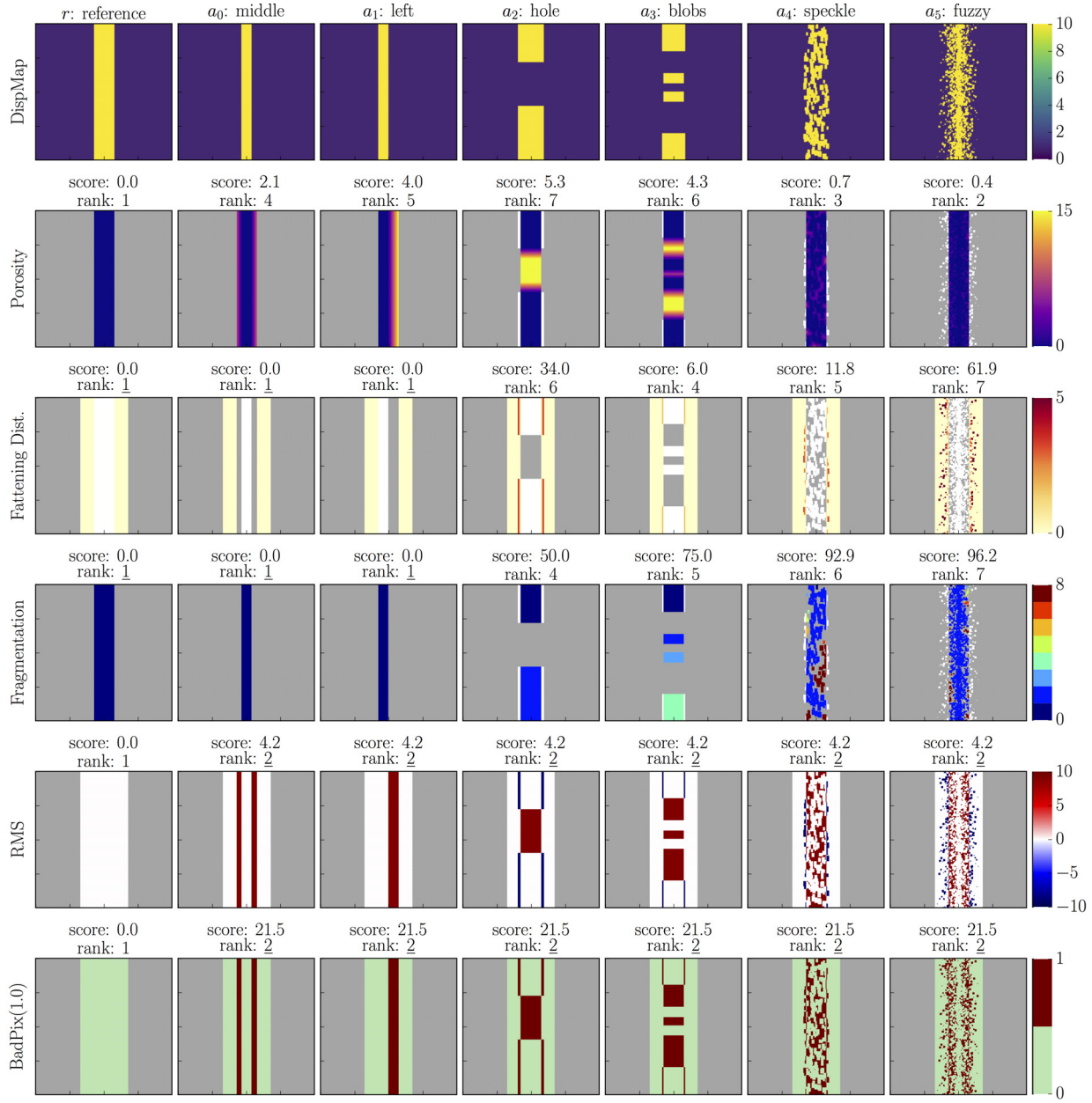


Figure 6.3.: **Systematic Test Cases for Fine Structure Metrics.** The top row depicts disparity maps of the reference r and the synthetic algorithm results $a_0 - a_5$. The yellow area on the reference disparity map represents a thin vertical bar in front of a planar background area. The test algorithms are designed to represent isolated failure cases at fine structures that are common in real algorithm data.
The test cases are designed such that *RMS* and *BadPix* scores are identical for all algorithms even though the shapes of the reconstructed structure differ greatly. Our *Porosity*, *Detail Fattening*, and *Fragmentation* metrics adequately reflect the performance differences.

more structure is missing on a_3 as compared to a_2. However, the more widespread distribution of the reconstructed parts in a_3 leads to better performance at the *Porosity* metric.

Detail Fattening. The *Detail Fattening* metric is designed analogously to the *Foreground Fattening* metric. It penalizes overestimated disparity values at the discontinuity area of fine structures, leading to high scores for a_2 and a_5, and to rather low scores for a_4 and a_3. As intended, a_0 and a_1 yield perfect *Detail Fattening* scores since all background disparities next to the fine structure are correct.

Fragmentation. The *Fragmentation* metric, as depicted in the fourth row of Figure 6.3, correctly quantifies that no fragmentation occurs on a_0 and a_1. It adequately reflects that fragmentation is mild on a_2 and a_3 but severe on the noisy results of a_4 and a_5.

RMS* and *BadPix. All algorithms have identical *RMS* and *BadPix* scores even though their reconstruction of the fine structure is very different. By contrast, the *Porosity*, *Detail Fattening*, and *Fragmentation* metrics do reflect the performance differences. For applications like the autonomous navigation of robots, it is important to know where the vehicle can move. Noisy results like a_5 are often sufficient for safe navigation while big missing parts like in a_2 or thinning like in a_1 may lead to the planning of impossible routes and collisions. Our metrics allow for a detailed quantitative comparison to support algorithm selection based on such application priorities.

6.1.5. Conclusions

We derived six systematic test cases for each of the three geometric entities to test the validity and expressiveness of our metrics. We showed that our metric scores and visualizations adequately reflect both, pronounced and subtle, performance differences between algorithms which are not quantified by *RMS* or *BadPix* scores. As intended, our metrics are invariant to disparity errors which are not related to the specific performance aspects of the respective metrics.

The artificial test cases demonstrate that our specific metrics complement the expressiveness of the general *RMS* and *BadPix* metrics. Thereby, they support the quantitative evaluation of algorithm performance with respect to application-specific priorities.

6.2. Comparison with Human Rankings

As discussed in Section 4.1, metrics should be semantically meaningful to support decision making for practitioners. Following requirement R3, each of our metrics should have a concise interpretation and its scores should be consistent with that interpretation.

We perform a user study to examine how well this requirement is satisfied by our metrics. Three human graders are asked to sort six stereo algorithms by performance according to our metric concepts. We use the obtained rankings to study three aspects: 1) Do our metric descriptions represent meaningful concepts with a concise interpretation? If this is the case, human graders should agree with each other and sort the algorithms in a similar way. 2) What kind of visual information is relevant for human graders to make performance differences consistently discernible? 3) Do our metric quantifications correspond to human performance assessment? If humans consistently rank algorithms according to a given performance aspect, our corresponding metric should produce a similar ranking.

In the next section, we describe the experimental setup. In the subsequent sections, the results of the user study are discussed for our metrics at continuous surfaces, depth discontinuities, and fine structures as well as for the prevalent per-pixel metrics *RMS* and *BadPix*.

6.2.1. Experimental Setup

We ask three human graders to sort six stereo algorithms according to nine different performance aspects. First, we ask graders to sort the algorithm results by general appearance. We then ask graders to sort the algorithms according to our proposed metric concepts. The graders have up to three years of experience in working with computer vision algorithms. The six algorithms are `CVF` [48], `Elas` [29], `PM` [96], `SPSS` [150], as well as `RSGM` and `SGBM` which are both based on Hirschmüller [40].

We perform each ranking experiment twice. For the *basic* variant, each grader is provided with information as depicted on Figure 6.5a. The input image and the reference disparity map are provided on the left. On the right, disparity maps of the six algorithms are placed on top of each other in random order. The instruction at the top asks the grader to sort the algorithms by performance according to one of the metrics. A brief description of the metric is provided as part of the instruction. For the *detailed* variant, the same procedure

is applied but additional visualizations are provided as depicted on Figure 6.5b. For the reference on the left, an additional normal map is displayed. For the algorithm results, four visualizations are provided: the disparity map as for the *basic* variant, the reference disparity map for direct comparison, a difference map $r - a$ which depicts the signed disparity error, and the normal map.

These two experiments are repeated for each performance aspect. All visualizations are provided in high resolution. Graders can zoom and pad the canvas as they wish to inspect the algorithm results. They are instructed to indicate ties by placing algorithms on the same horizontal position. Metric scores are considered equal when they agree up to the first decimal. In case of ties, the average rank is assigned to all algorithms which is indicated by underlined ranks. In order to check intra-grader consistency, a randomly chosen subset of the experiments is added twice to the list of experiments.

Including the intra-grader consistency experiments, each grader is asked to perform a total of 24 rankings of the six algorithms on three different scenes (see Figure 6.5) according to nine different performance aspects. For each grader, processing time per experiment takes between four to ten minutes, resulting in a total effort of more than two hours of strenuous mental effort with high attention to detail. Due to this effort and the relatively large number of metrics, our study is limited to one examined scene per metric. Nonetheless, we obtain a total of 86 rankings from the three graders, the two per-pixel metrics, and our proposed performance metrics.

In the following sections, these rankings are used to assess how well human graders agree on the different performance aspects and how well our metric rankings correlate with human rankings.

(a) Ranking experiment with *basic* visualization

(b) Ranking experiment with *detailed* visualization

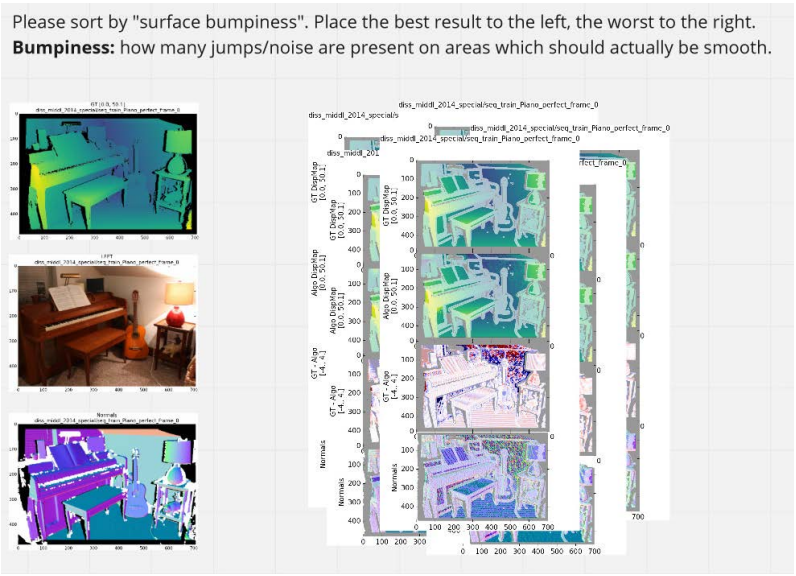


Figure 6.4.: **Setup for the User Study.** Each ranking experiment is performed twice. For the *basic* visualization, visual information is limited to input images and disparity maps (a). The *detailed* visualization provides input images, disparity maps, error maps, and normal maps (b).

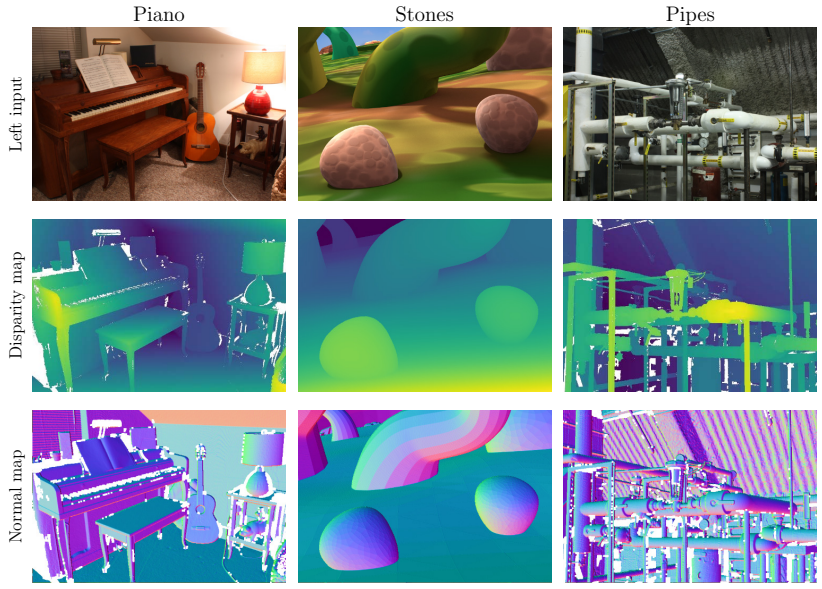


Figure 6.5.: **Scenes for the User Study.** We use the *Piano*, *Stones*, and *Pipes* scenes for continuous surfaces, depth discontinuities, and fine structures, respectively.
The *Piano* and *Pipes* scene are from the Middlebury 2014 dataset by Scharstein et al. [111]. The *Stones* scene is from the Monkaa dataset by Mayer et al. [82].

6.2.2. Continuous Surfaces

We use the *Piano* scene of the Middlebury 2014 dataset [110] to compare our continuous surface metrics with human ratings. Figure 6.6 depicts algorithm results, normal maps, and surface metric visualizations for six stereo algorithms. Table 6.1 depicts the respective ranks assigned by three human graders G1-G3 and by our metrics.

Angular Error. As depicted on the third row of Figure 6.6, our *Angular Error* scores indicate that the algorithms `PM` and `SPSS` both perform well at continuous surfaces while `CVF` and `RSGM` perform rather poorly. This is consistent with human rankings as depicted on Table 6.1a. When provided with the *detailed* visualization, relative rankings for the surface orientation are very consistent between graders as well as for the same grader when asked to rank repeatedly. For each algorithm, ranks are either identical or differ at most between two adjacent ranks among all human ratings and our algorithmic metric rating.

When provided with the *basic* visualization, rankings are fairly inconsistent among graders as well as between graders and our metric. The algorithms `CVF` and `PM` feature the highest inter-grader ranking variance. Indeed, their disparity maps on Figure 6.6 look very smooth with only few artifacts. Yet, on the normal maps, the piecewise planar estimates of `CVF` are clearly visible while the normal maps of `PM` are very smooth and more similar to the reference normal map in Figure 6.5. Apparently, the normal map visualizations as depicted on the second row of Figure 6.6 are key to viable human assessment of surface quality.

Bumpiness. We observe similar phenomena for the *Bumpiness* metric. As depicted on Table 6.1b, inter-grader agreement is high on the *detailed* and lower on the *basic* visualization. Our metric ranks correspond well with human ranks of the *detailed* visualization except for `CVF` and `Elas`. In contrast to our metric, human graders consistently rate `Elas` better than `CVF`. As depicted on Figure 6.6, `CVF` produces smooth piecewise planar results with severe steps in-between. `Elas` produces irregular but smooth results on the wall and good results on the floor. Apparently, the staircasing of `CVF` between the otherwise smooth surfaces is penalized more heavily by human assessment as compared to our metric. If required for application-specific evaluation, our metric could be adjusted accordingly by applying more specific statistical measures to the distribution of per-pixel bumpiness values instead of reporting their mean.

(a) *Angular Error*

		CVF	Elas	PM	RSGM	SGBM	SPSS
	score	53.0	34.2	24.6	61.1	33.0	17.6
	rank	5	4	2	6	3	1
detailed	G1	5	4	2	6	3	1
detailed	G2	5	3	2	6	4	1
detailed	G3	6	3	2	5	4	1
detailed	G1	6	3	1	5	4	2
detailed	G2	6	3	2	5	4	1
detailed	G3	6	3	1	5	4	2
basic	G1	1	3	6	5	4	2
basic	G2	3	5	2	6	4	1
basic	G3	6	3	1	5	4	2

(b) *Bumpiness*

		CVF	Elas	PM	RSGM	SGBM	SPSS
	score	63.7	87.4	23.5	180.2	67.3	15.3
	rank	3	5	2	6	4	1
detailed	G1	5	3	1	6	4	2
detailed	G2	6	3	2	5	4	1
detailed	G3	5	3	1	6	4	2
basic	G1	3	4	2	5	6	1
basic	G2	6	4	2	5	3	1
basic	G3	6	4	3	5	2	1

Table 6.1.: **Rank Comparison for Continuous Surface Results.** The tables denote ranks assigned to algorithms `CVF-SPSS` by graders G1-G3. Lower ranks are better. Details of the *basic* and *detailed* visualizations are explained on Figure 6.4. The second set of ranks for the *detailed* visualization stems from the repeated experiments to assess inter-grader consistency. Agreement is high among graders and compared to our *Angular Error* (a) and *Bumpiness* (b) metrics when graders are provided with the *detailed* visualization which includes normal maps.

6.2.3. Discontinuities

We use a frame of the Monkaa dataset by Mayer et al. [82] to compare our discontinuity metrics with human ratings. Figure 6.7 depicts algorithm results, difference maps, and discontinuity metric visualizations with scores and ranks. Table 6.2 depicts the respective ranks assigned by the human graders and our metrics.

Foreground Fattening. The algorithm `SPSS` produces by far the most severe edge fattening as depicted on the first and third row of Figure 6.7. `SPSS` is consistently ranked last by the human graders and by our *Fattening* metric as depicted on Table 6.2a.

Apart from `SPSS`, ratings differ strongly between graders, between visualization variants, and with respect to our *Fattening* metric. Apparently, it is difficult for human graders to consistently discern differences in edge fattening performance. For our given experimental setup, this may be caused by multiple reasons. First, agreement between graders and our metric is higher on the *basic* visualization as compared to the *detailed* visualization. For the later, graders may be influenced and distracted by the high errors on the ground (see second row of Figure 6.7). Second, based on the provided visualizations, it is difficult for human graders to tell exactly where the discontinuity should be located, especially if the overall shape of the estimated object appears reasonable. Displaying the reference discontinuity on top of the algorithm might help graders to assess the exact location of the discontinuity. Third, performance differences between `Elas`, `PM`, `RSGM`, and `SGBM` are very subtle on the *Stones* scene and thus hard to distinguish. As illustrated on the third row of Figure 6.7, `CVF` and `SPSS` produce the best and worst *Foreground Fattening* results. For those algorithms, ranks are mostly consistent among graders and with respect to our metric (see Table 6.2a).

Foreground Thinning. All algorithms exhibit rather low thinning of foreground objects except for `RSGM` which misses the upper part of the plant. As depicted on the last row of Figure 6.7, `PM`, `Elas`, `SGBM`, and `SPSS` produce little thinning with almost perfect scores. Their relative performance difference is below practical relevance for most applications.

As depicted on Table 6.2b, the graders and our metric are mostly consistent in assigning `RSGM` the lowest rank. For the remaining algorithms, ranks are highly inconsistent. We hypothesize that this is due to similar reasons as described for the *Fattening* metric. Comparing the results of `PM`, `SGBM`, and `SPSS` indicates that human graders are likely to be influenced more strongly by the shape of the edge rather

(a) *Foreground Fattening*

		CVF	Elas	PM	RSGM	SGBM	SPSS
	score	2.8	20.4	13.2	17.0	22.9	43.0
	rank	1	4	2	3	5	6
detailed	G1	4	5	3	2	1	6
	G2	1	3	4	5	2	6
	G3	4	3	5	1	2	6
detailed	G1	2	5	1	4	3	6
	G2	1	3	4	5	2	6
	G3	4	3	5	1	2	6
basic	G1	1	4	3	2	6	5
	G2	1	5	2	3	4	6
	G3	1	4	3	2	5	6

(b) *Foreground Thinning*

		CVF	Elas	PM	RSGM	SGBM	SPSS
	score	5.3	1.8	0.6	8.8	0.0	0.9
	rank	5	4	2	6	1	3
detailed	G1	2	6	1	4	5	3
	G2	4	5	1	6	3	2
	G3	3	6	1	4	5	2
basic	G1	3	4	2	6	1	5
	G2	5	3	2	6	4	1
	G3	3	2	1	6	4	5

(c) *Edge Quality*

		CVF	Elas	PM	RSGM	SGBM	SPSS
basic	G1	2	5	1	4	3	6
	G2	1	5	2	4	3	6
	G3	1	6	2	3	4	5

Table 6.2.: **Rank Comparison for Depth Discontinuity Results.** For *Fattening* (a) and *Thinning* (b), consistency is low except for the best and worst performing algorithms. Human graders consistently rank algorithms by the general concept of *Edge Quality* (c).

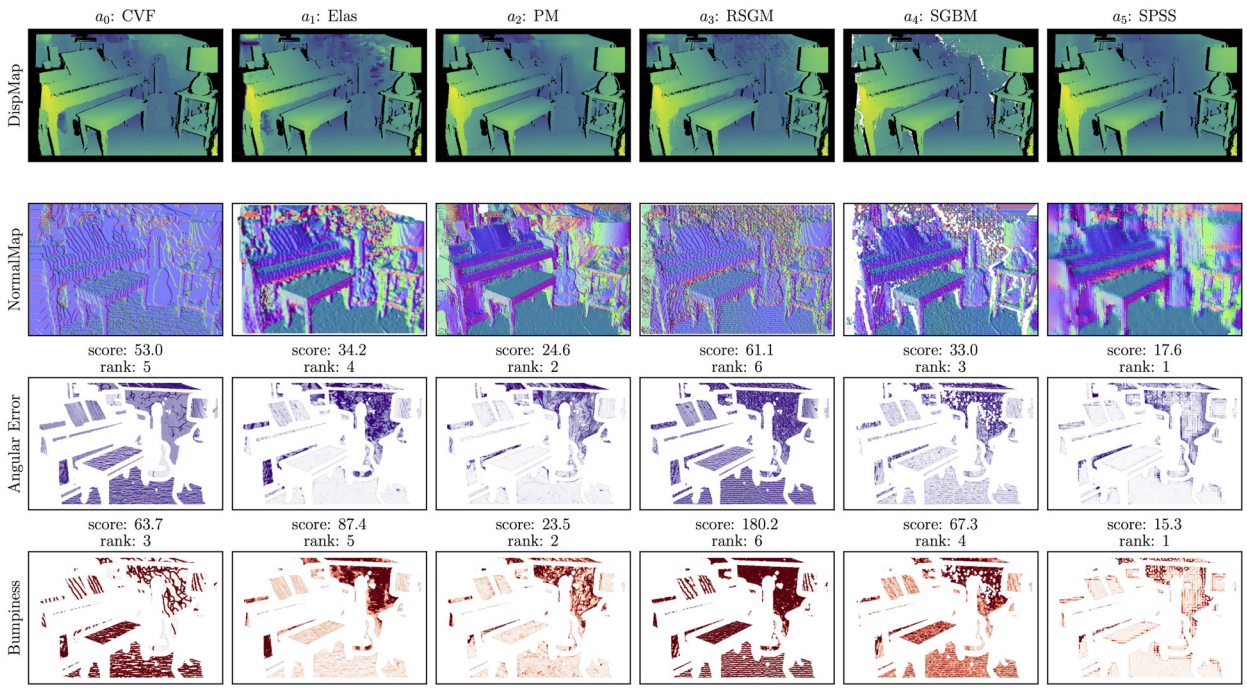


Figure 6.6.: **Algorithm Performance at Continuous Surfaces on *Piano* [111].** Differences in algorithm performance are considerably more pronounced on the normal maps in the second row than on the disparity maps in the top row. The reference disparity and normal map is shown on Figure 6.5.

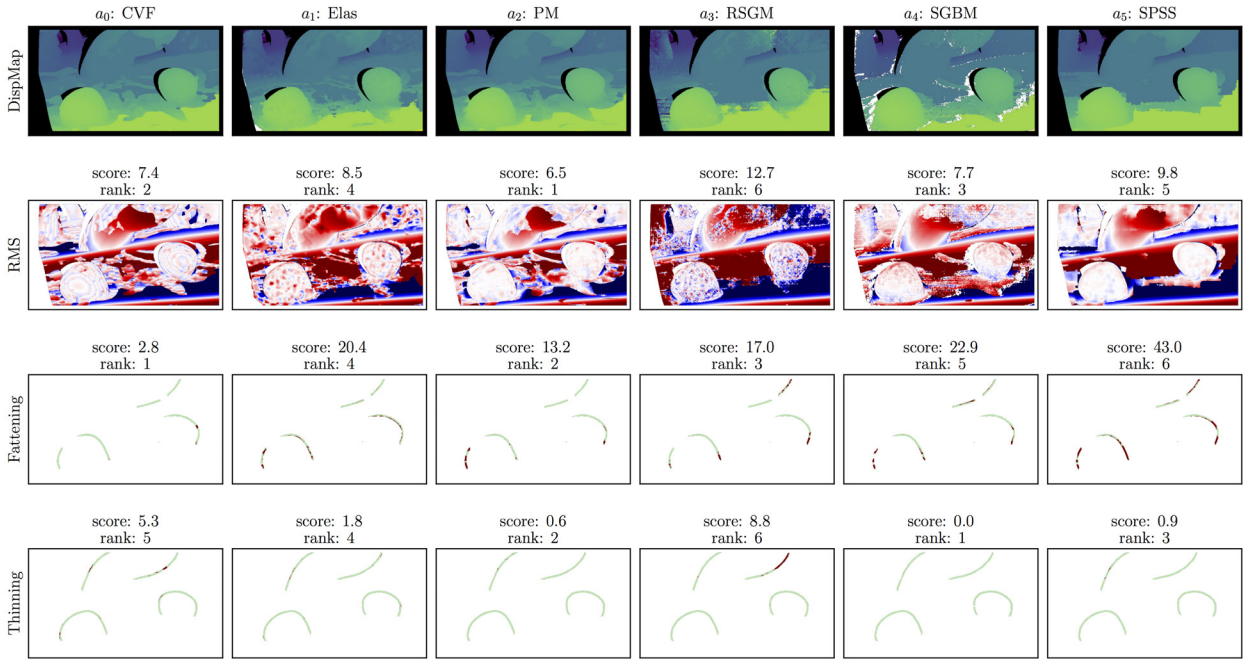


Figure 6.7.: **Algorithm Performance at Discontinuities on *Stones* [82].** It is difficult to visually discern subtle differences in discontinuity performance without the metric visualizations of the lower rows. Displaying reference discontinuities on top of the algorithm results might support human graders. The reference disparity map is shown on Figure 6.5.

than its exact location. All three algorithms exhibit almost no thinning. Yet, PM which produces crisp discontinuities, is ranked better than SGBM and SPSS which produce more irregular discontinuities.

Edge Quality. We conduct an additional experiment to test our hypothesis that the "visual appeal" of edges is an intuitive and concise concept for human graders. Table 6.2c depicts ranking results based on the *basic* visualization when graders are asked to sort the algorithms based on *Edge Quality*. Elas and SPSS are consistently assigned the worst ranks. Their discontinuities are most jagged and tortuous. CVF and PM consistently score best. They feature the most crisp and clean discontinuities. As future work, *Edge Quality* could be defined as an additional metric quantifying the shape of discontinuities rather than their location.

6.2.4. Fine Structures

We use the *Pipes* scene of the Middlebury 2014 dataset [111] to compare our fine structure metrics with human ratings. Figure 6.8 shows algorithm results and fine structure metric visualizations with scores and ranks. Table 6.3 denotes the respective ranks assigned by human graders and our metrics.

Porosity. As depicted on the first and second row of Figure 6.8, the algorithms Elas, RSGM, and SPSS almost flawlessly reconstruct the vertical pipe, while SGBM achieves moderate performance, CVF and PM are missing big parts of the structure. This is reflected by human rankings and by our *Porosity* scores on Table 6.3a. SGBM is unanimously ranked below the trio of top performing algorithms and above the duo of poorly performing algorithms.

The *Porosity* scores indicate a tie between Elas, RSGM, and SPSS with metric values adequately indicating very good performance. All human graders assigned the top three ranks to those three algorithms. However, ranking variance within this spectrum is high. During the study, two graders positioned the algorithms with substantial partial overlap, possibly indicating the performance similarity, though no grader indicated a tie. On this scene, human assessment of *Porosity* may have been influenced by edge fattening effects. Among the six algorithms, Elas, RSGM, and SPSS produce the most notable foreground fattening (see third row of Figure 6.8). The graders potentially assessed these seeming dents of the imperfect discontinuities outside the actual structure. Visual performance assessment may be supported by additionally overlaying reference discontinuities on the algorithm results.

(a) *Porosity*

		CVF	Elas	PM	RSGM	SGBM	SPSS
	score	18.5	0.1	14.5	0.1	2.3	0.1
	rank	6	2	5	2	4	2
detailed	G1	5	2	6	3	4	1
	G2	5	1	6	3	4	2
	G3	6	3	5	2	4	1
basic	G1	6	1	5	2	4	3
	G2	6	3	5	1	4	2
	G3	5	3	6	1	4	2

(b) *Detail Fattening*

		CVF	Elas	PM	RSGM	SGBM	SPSS
	score	1.1	9.8	0.2	5.7	1.9	5.6
	rank	2	6	1	5	3	4
detailed	G1	2	5	1	4	3	6
	G2	2	6	1	5	3	4
	G3	2	6	1	5	4	3
basic	G1	1	6	2	4	3	5
	G2	2	6	1	4	3	5
	G3	2	6	1	4	3	5

(c) *Fragmentation*

		CVF	Elas	PM	RSGM	SGBM	SPSS
	score	50.0	0.0	66.7	0.0	90.0	0.0
	rank	4	2	5	2	6	2
detailed	G1	5	2	4	2	6	2
	G2	4	2	5	2	6	2
	G3	4	2	5	2	6	2
basic	G1	4	2	5	2	6	2
	G2	4	2	6	2	5	2
	G3	4	2	6	2	5	2

Table 6.3.: **Rank Comparison for Fine Structure Results.** For *Porosity* (a), agreement is high for CVF, PM, and SGBM. Graders report different rankings for the top three algorithms Elas, RSGM, and SPSS which are rated as tied by our metric. For *Detail Fattening* (b) and *Fragmentation* (c), agreement is high among graders and with our metrics.

Detail Fattening. For fine structure fattening, human and metric ranks are mostly consistent for both, the *basic* and the *detailed* visualization. `PM` and `CVF` are consistently ranked the two top performing algorithms. `SPSS`, `RSGM`, and `Elas` are ranked the worst performing algorithm with respect to fattening. While it was hard for human graders to consistently rate *Foreground Fattening* on the *Stones* scene, it appears to be more feasible on the single, almost vertical bar in *Pipes*.

Fragmentation. Our *Fragmentation* metric and all human graders on both visualizations report a tie for algorithms `Elas`, `RSGM`, and `SPSS` which is not surprising as all three algorithms reconstruct the full pipe as a single block. For the remaining algorithms, the majority of human rankings is consistent with our metric ranks.

Graders reported that they were unsure about whether the mere number of fragments or the size of the fragments should be taken into account. Our *Fragmentation* metric as defined in Section 4.2.3 is focused on the number of fragments with an optional parameter for minimal fragment size. We argue that this is sufficient since the sampling of the structure is quantified by the *Porosity* metric.

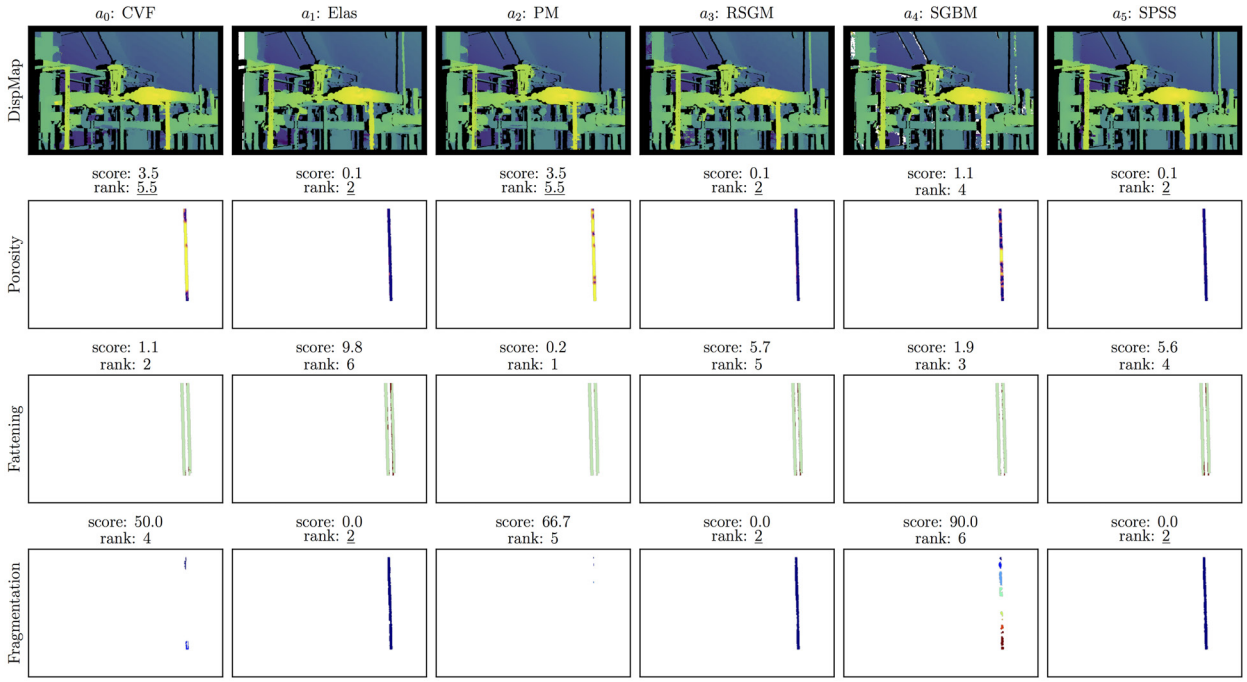


Figure 6.8.: **Algorithm Performance at Fine Structures on *Pipes* [111].** Evaluation is focused on the rightmost vertical pipe. For our metric visualizations on rows two to four, we show a crop focusing on this pipe. The reference disparity map is provided on Figure 6.5.

6.2.5. General Metrics

For each scene, we first asked graders to sort the algorithm results by general appearance before asking to sort by more specific phenomena. We compare these general rankings with *RMS* and *BadPix* scores as depicted on Table 6.4. Algorithm disparity maps as well as metric visualizations together with scores and ranks for each scene are shown on Figure 6.9. Similar to our *Foreground Fattening* and *Foreground Thinning* metrics, ranks tend to be highly inconsistent among graders, visualizations, and metrics except for cases of very high or very low algorithm performance.

On the *Piano* scene, `SPSS` is consistently ranked very high for all graders, metrics, and visualization variants. The *BadPix* score is considerably better for `SPSS` while scores are poorer and very similar to each other for all other algorithms. Apparently, it is hard for humans to consistently rate such subtle differences in general performance. As shown in Section 6.2.2, ranking on *Piano* is more consistent when graders are asked to assess surface quality.

On the *Stones* scene, `PM` and `RSGM` are consistently ranked best and worst on both metrics and by all graders when provided with the *detailed* visualization. When provided with the *basic* visualization, graders tend to rate `SPSS` to feature the lowest performance. As shown on the top row of Table 6.4b, the disparity map of `SPSS` yields the most blocky result and the strongest fattening at object edges. When provided with the *detailed* visualization, `SPSS` is ranked second and third which is in line with the *BadPix* ranking.

On the *Pipes* scene, human ratings are very consistent for the *basic* visualization. Apparently, graders are influenced by the appearance of the rightmost pipe followed by the quality of the object edges. The three top ranked algorithms `SPSS`, `Elas`, and `RSGM` reconstruct the pipe in the respective order. The second last and last ranked algorithms `CVF` and `PM` almost completely miss the pipe with `PM` also producing more foreground bleeding on the lower right vertical bar (see Figure 6.9c).

(a) *Piano*

		CVF	Elas	PM	RSGM	SGBM	SPSS
	RMS	2.8	3.7	2.8	2.7	3.2	2.1
	rank	3.5	6	3.5	2	5	1
	BadPix2	15.3	15.9	17.0	15.1	17.7	8.6
	rank	3	4	5	2	6	1
detailed	G1	6	5	2	4	3	1
detailed	G2	6	5	2	3	4	1
detailed	G3	6	3	1	5	4	2
basic	G1	3	6	2	4	5	1
basic	G2	2	5	3	6	4	1
basic	G3	2	1	4	5	6	3

(b) *Stones*

		CVF	Elas	PM	RSGM	SGBM	SPSS
	RMS	7.4	8.5	6.5	12.7	7.7	9.8
	rank	2	4	1	6	3	5
	BadPix2	41.4	48.5	37.0	61.5	48.5	46.7
	rank	2	4.5	1	6	4.5	3
detailed	G1	5	3	1	6	4	2
detailed	G2	4	5	1	6	3	2
detailed	G3	2	5	1	6	4	3
basic	G1	3	5	2	4	6	1
basic	G2	4	2	5	1	3	6
basic	G3	2	5	1	3	4	6
basic	G1	2	3	1	4	5	6
basic	G2	5	3	4	1	2	6
basic	G3	4	2	1	3	5	6

(c) *Pipes*

		CVF	Elas	PM	RSGM	SGBM	SPSS
	RMS	6.2	6.0	6.5	6.9	6.0	7.6
	rank	3	1.5	4	5	1.5	6
	BadPix2	12.5	12.4	13.1	13.8	11.5	11.4
	rank	4	3	5	6	2	1
detailed	G1	4	5	2	5	3	1
detailed	G2	6	2	3	5	1	4
detailed	G3	6	3	1	5	4	2
basic	G1	5	2	6	3	4	1
basic	G2	5	2	6	3	4	1
basic	G3	5	2	6	3	4	1

Table 6.4.: **Rank Comparison for General Appearance.** Rating agreement is low on all three scenes except for the *detailed* visualization on *Piano* and the best and worst performing algorithms.

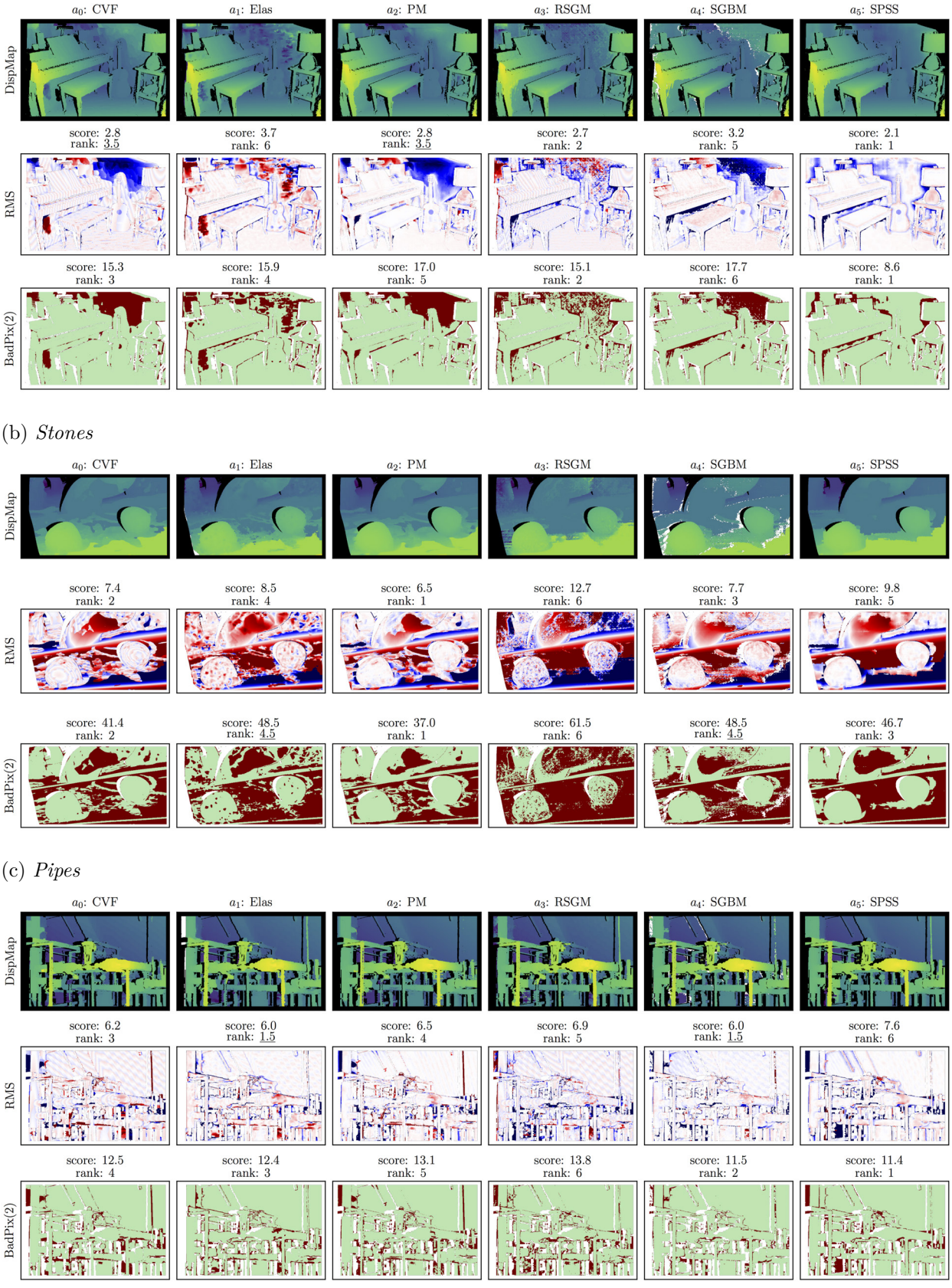


Figure 6.9.: **General Algorithm Performance on all Scenes.** On the *Pipes* scene, differences in scores and error visualizations are subtle between algorithms for both metrics. Differences among groups of algorithms are small on the *Piano* and the *Stones* scene. The reference disparity maps are shown on Figure 6.5.

6.2.6. Conclusions

We conducted a user study to evaluate two aspects: the semantic coherence and tangibility of our proposed metric concepts and the ranking consistency between human assessment and our algorithmic scores. Three human graders were asked to rank six algorithms according to the different concepts.

High ranking agreement between graders showed that the orientation and bumpiness of continuous surfaces as well as the fragmentation and porosity of fine structures are intuitive and visually discernible concepts. Rankings obtained from our algorithmic quantification of these phenomena is consistent with human assessment. For continuous surfaces, *detailed* visualizations including normal maps resulted in much higher consistency while plain disparity maps were found to be sufficient for fine structures.

For fattening and thinning at depth discontinuities as well as for general algorithm performance, intra- and inter-grader consistency was low except for extreme cases of very good or very poor algorithm performance. Consistently discerning subtle differences in algorithm performance appears to be difficult for humans. For discontinuities, we found that humans consistently rank the quality of the shape while rating the correctness of the location is difficult. Displaying the reference discontinuities on top of the algorithm results might provide useful additional information to human graders.

Our user study had to be limited to two rankings per metric and grader due to the large number of metrics and the effort required to carefully rank the algorithms. More images per metric and more graders would be required to perform a statistical analysis and to derive more definite conclusions.

6.3. Evaluation of Stratified Scenes

In this section, we test the applicability and expressiveness of our stratified scenes. First, we perform an experimental algorithm evaluation for each scene individually. Second, we compare the average performance of 12 algorithms to analyze the solvability and local difficulty of the scenes.

6.3.1. Experimental Algorithm Evaluation

To analyze the stratified scenes, we perform visual inspection and quantitative degradation analysis as described in Section 5.2.3. For each scene, we evaluate four different algorithms to test whether the visual inspection of disparity maps and error maps reveals different algorithm strengths and weaknesses. We further test to what degree the spatially increasing difficulties cause algorithm performance to deteriorate.

Backgammon: Thin Structures and Narrow Gaps

The degradation graphs on Figure 6.10 and the algorithm results on Figure 6.11 show that the *Backgammon* scene challenges algorithms at robustly reconstructing thin structures and narrow gaps. As intended, both visualizations highlight performance differences between the algorithms.

As highlighted by Figure 6.11, the algorithms a_0 and a_1 struggle with severe foreground fattening between the narrow gaps. From top to bottom of the scene, fattening increases considerably with narrower gaps (see Figure 6.10a). For a_1 and a_2, fattening is substantially worse on the left image area where disparity differences between foreground and background are high (see Figure 6.11). At this area, it is harder to match correspondences since bigger areas of the background are occluded between the views. The quantitative analysis on Figure 6.10b reveals that a_3 does not perform well at thinning which is more difficult to see on the qualitative evaluation in Figure 6.11.

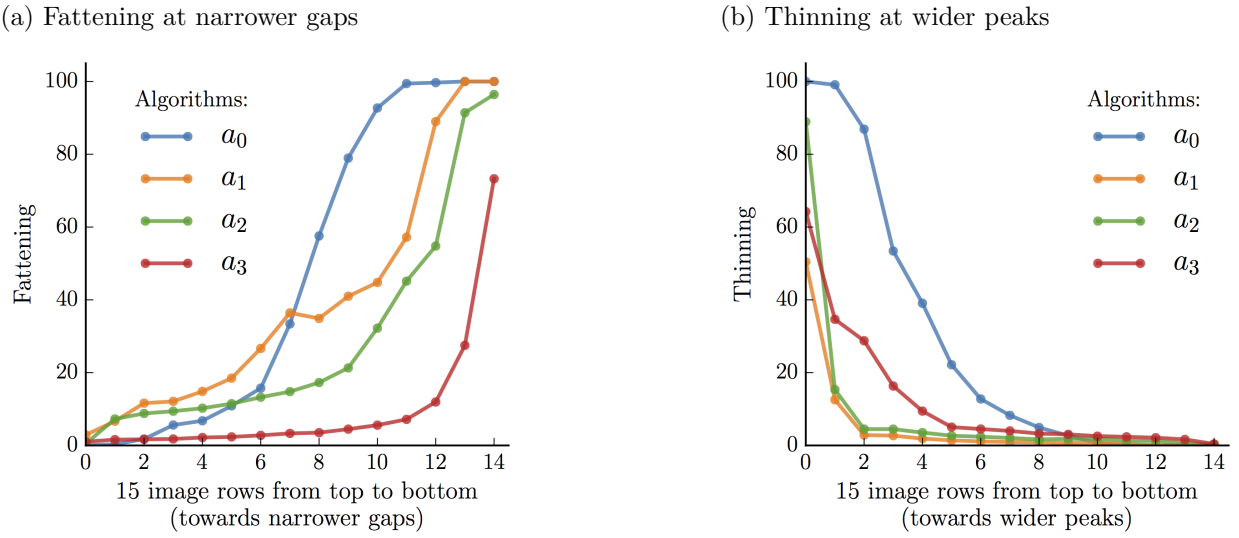


Figure 6.10.: **Quantitative Degradation Analysis on *Backgammon*.** We quantify *Fattening* and *Thinning* for each image row as depicted on the mask in Figure 5.4. The leftmost data point on the graph corresponds to the topmost row on the *Backgammon* scene, featuring wide gaps and thin peaks.
Fattening increases and thinning decreases with narrower gaps and wider peaks towards the bottom of the scene. Algorithm a_0 performs poorly on both metrics. The relative ranking of the other algorithms is reversed for the *Fattening* metric (a) as compared to the *Thinning* metric (b).

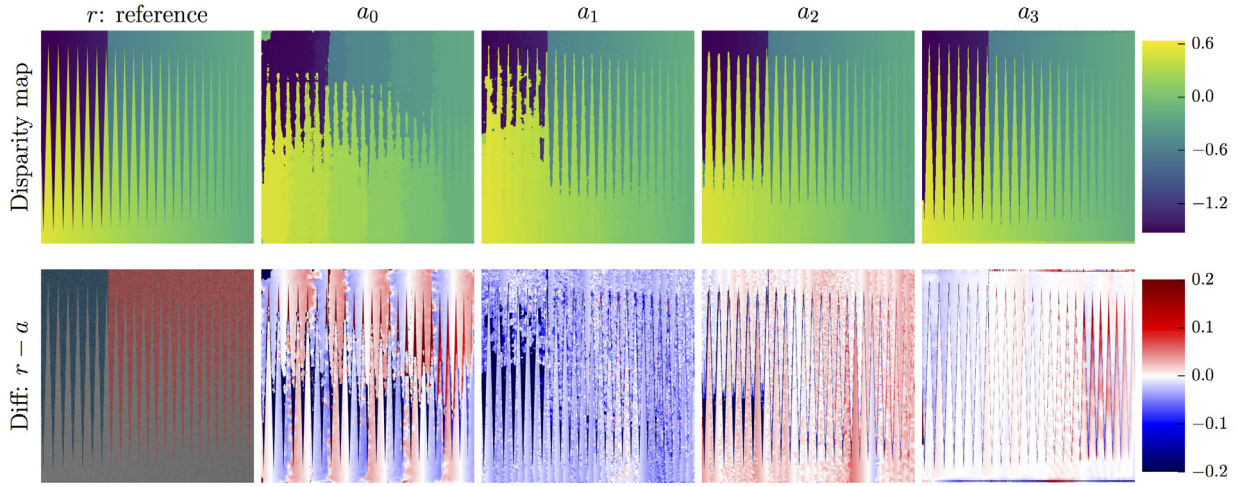


Figure 6.11.: **Qualitative Evaluation on *Backgammon*.** The top row depicts reference and algorithm disparity maps. The first image on the bottom row shows the input image of the center view. The remaining images depict the signed per-pixel disparity differences $r - a$ between the reference and algorithm disparity maps.
The visualizations on the *Backgammon* scene reveal considerable differences in algorithm performance at fine structures and narrow gaps. Algorithm a_1 produces severe fattening between the gaps, especially at regions with big disparity differences. Algorithm a_3 performs well between the gaps but exhibits thinning at the fine peaks.

Pyramids: Planar and Non-Planar Surfaces

The algorithm results on Figure 6.12 and 6.13 show that the *Pyramids* scene fulfills its purpose of revealing different algorithm strengths and weaknesses at surface reconstruction.

The visualization on the second row in Figure 6.13 highlights that algorithm a_0 produces a smooth reconstruction of the fronto-parallel base plane but poor reconstructions of the slanted object surfaces. The errors on the hemisphere reveal that the algorithm is able to reconstruct the flatter regions of the hemisphere while performance degrades at the steeper outer areas. The *Pyramids* scene further reveals that a_2 performs much better at the planar pyramid surfaces than at the hemispheres. Indeed, a_2 applies local plane fitting as a refinement step which is causing the observed behavior. Both, the pyramids and hemispheres tend to be flattened by all algorithms. The convex objects on the upper image area appear red, indicating under-estimated disparities, while concave objects on the lower area appear blue. This effect is particularly strong for a_2.

On Figure 6.12, ground truth disparities are plotted against algorithm disparities. Algorithm a_3 produces almost perfect results. It features almost the same algorithm disparity on the y axis for each ground truth disparity on the x axis. a_1 is also very accurate, but the flattened maximum of the hemisphere is clearly visible on the lower left corner of the plot. The plot further reveals the offset of a_2 and the staircasing of a_0.

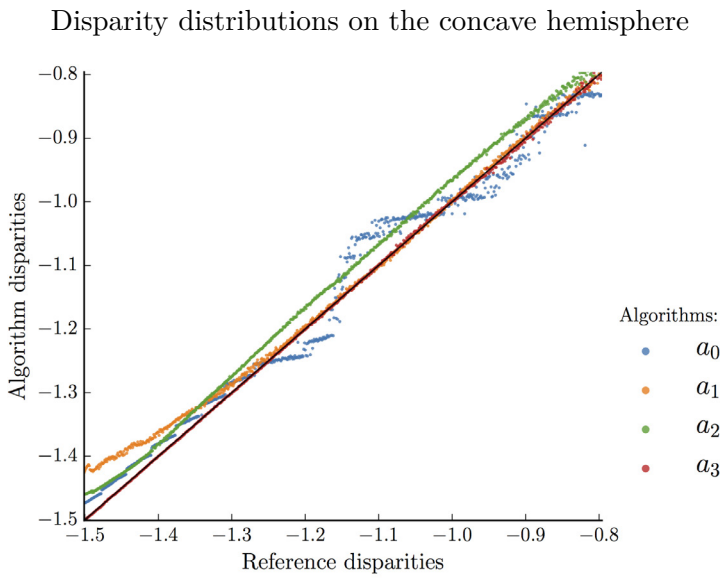


Figure 6.12.: **Disparity Distributions on *Pyramids*.** To identify bias and staircasing, we plot ground truth disparities against the corresponding algorithm disparities. Algorithm a_3 yields almost perfect results as its disparities closely match the identity function. The bias of a_2 and the staircasing of a_0 are highlighted by this visualization.

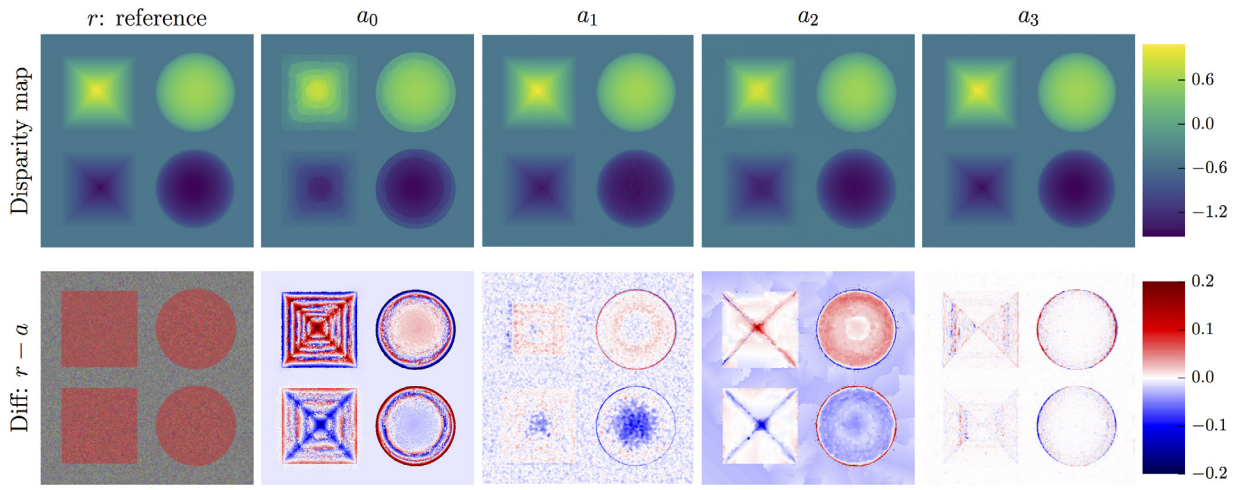


Figure 6.13.: **Qualitative Evaluation on *Pyramids*.** The *Pyramids* scene reveals that algorithm performance differs considerably at continuous surfaces. These surface properties are hardly visible on the disparity maps. Difference maps as depicted in the second row provide valuable information for qualitative evaluation.

Dots: Noise and Tiny Objects

The degradation plots on Figure 6.14 and the algorithm results in Figure 6.15 show that the *Dots* scene fulfills its purpose of challenging algorithms at finding a good trade-off between robustness to noise and sensitivity to details. Evaluation on *Dots* reveals that performance of all algorithms degrades considerably with higher noise levels. Depending on the regularization, performance degradation affects either mostly the accuracy of the background or the sensitivity to small objects.

As highlighted by Figure 6.15, algorithms a_0 and a_1 apply little regularization, resulting in good sensitivity but poor robustness. a_2 produces an almost perfect reconstruction of the background while a_3 produces very smooth but incorrect results.

No noise was applied on the first cell of the scene. Nonetheless, all four algorithms reconstruct only the four biggest of the seven evaluated dots as denoted by the leftmost data points on the quantitative degradation analysis in Figure 6.14b. For a_1, a_2, and a_3, the number of missing dots increases steadily with higher levels of noise. By contrast, a_0 produces severe artifacts on the background of the noisy cells. Some of these artifacts fulfill the conditions of a detected dot, resulting in low numbers of missing dots at high noise levels. This situation highlights that qualitative and quantitative evaluation should be performed jointly to maximize insights from the stratified scenes.

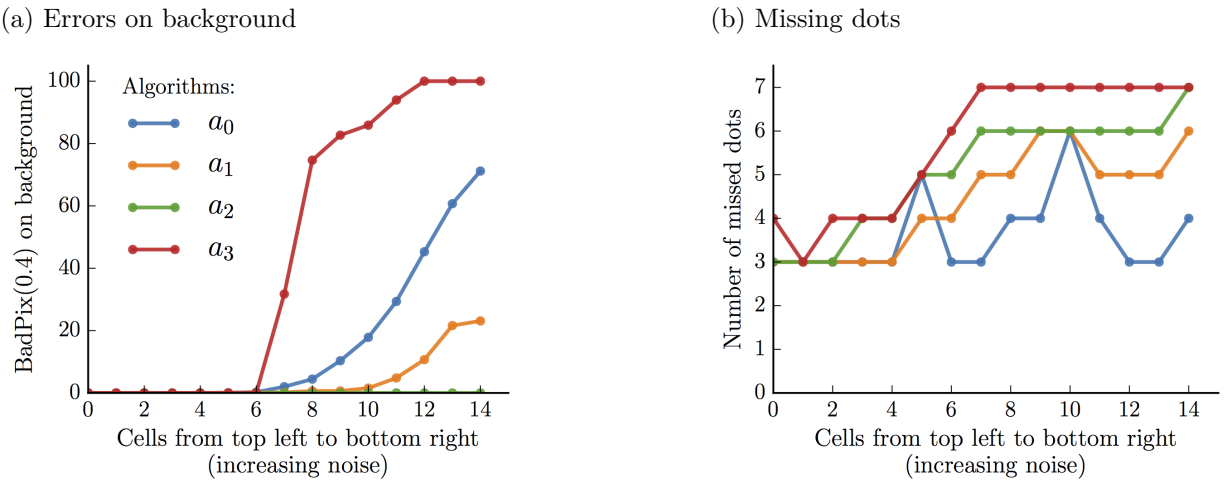


Figure 6.14.: **Quantitative Degradation Analysis on *Dots*.** For each cell as depicted on the mask in Figure 5.6 we compute *BadPix(0.4)* scores on the background (a) as well as the number of missing dots (b). We consider a dot detected if at least 80% of its surface is estimated with an error below 0.4. Results are plotted from left to right with increasing levels of noise. Background accuracy and sensitivity to detail degrade with increasing noise. Relative rankings of a_0, a_1, and a_2 are reversed for the two aspects.

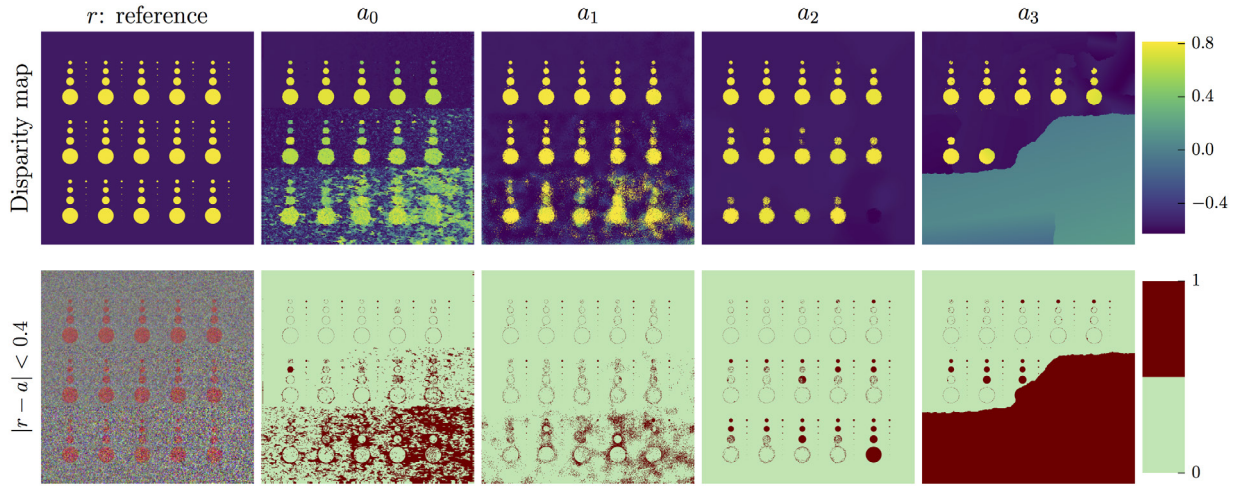


Figure 6.15.: **Qualitative Evaluation on *Dots*.** This scene reveals the difficulty of achieving both together, robustness to noise and sensitivity to small objects. Performance degrades for all algorithms with increasing levels of noise towards the lower right. Algorithms a_0 and a_1 remain sensitive to small dots but exhibit artifacts on the background. By contrast, a_2 applies strong regularization, resulting in a smooth background but many missing dots.

Stripes: Texture and Contrast at Occlusion Regions

The degradation plots in Figure 6.16 and the algorithm results on Figure 6.17 show that the *Stripes* scene fulfills its purpose of challenging algorithms at dealing with low levels of texture and contrast. The visualizations reveal that most algorithms are severely affected by low contrast at occlusion regions. Furthermore, they highlight algorithm differences at handling decreasing levels of texture.

The algorithms a_0 and a_1 apply strong regularization when little matching information is available due to low levels of texture. Foreground texture is high and background texture is low on the top left image area. As depicted on Figure 6.17, this setup causes a_0 and a_1 to interpolate the poorly textured background areas with foreground geometry. The opposite effect occurs on the bottom right image area. The poorly textured low contrast stripes are interpolated with background geometry. The implementation of a_0 relies on the *figural continuity assumption*, i.e. image gradients are used as priors for handling occlusions at depth discontinuities. This assumption is violated by the bright stripes, causing a_0 to miss all low contrast stripes. As depicted on Figure 6.16b, a_2 and a_3 perform well for high and moderate levels of texture. Performance degrades abruptly below a certain level of texture. By contrast, a_0 and a_1 yield poor performance even for high levels of texture.

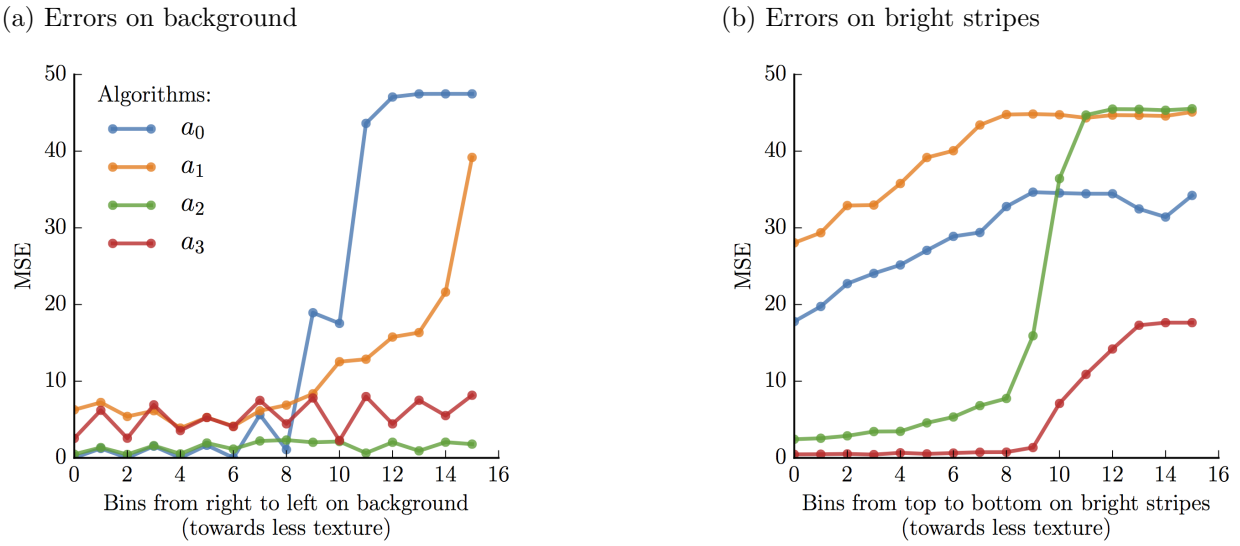


Figure 6.16.: **Quantitative Degradation Analysis on *Stripes*.** To quantify performance degradation, we compute a range of *MSE* scores (see Equation 3.5) along the axes of decreasing texture. (a) We compute the *MSE* separately for each background area between two stripes. The leftmost data point on the graph corresponds to the rightmost background area with the highest level of texture. (b) We further compute the *MSE* on horizontal image bins from top to bottom on the bright stripes. The leftmost data point corresponds to the topmost stripe area with the highest level of texture.

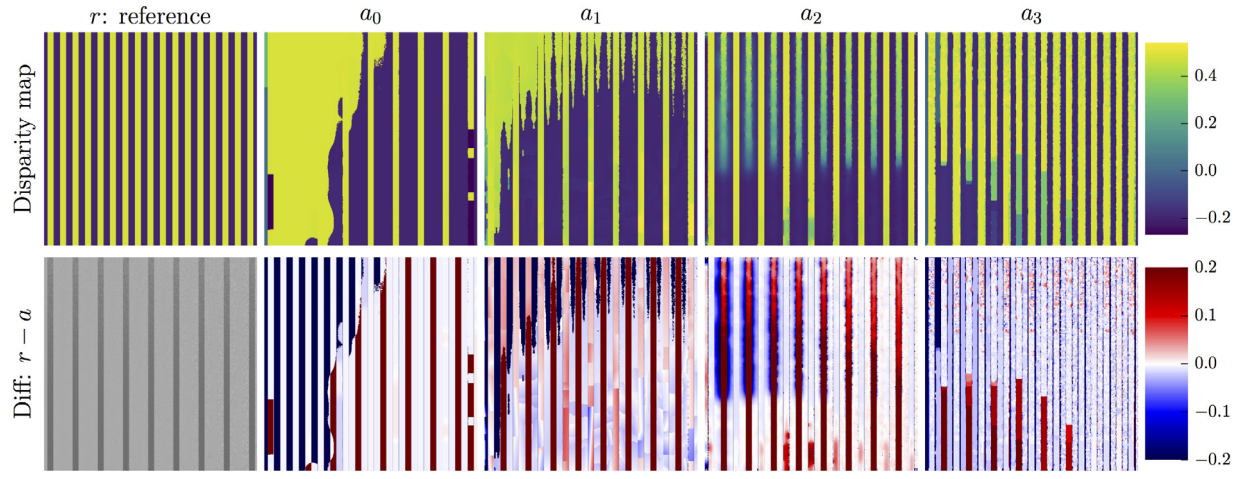


Figure 6.17.: **Qualitative Evaluation on *Stripes*.** The *Stripes* scene reveals that algorithms struggle severely with low levels of texture and contrast towards the bottom of the image. Algorithms a_0, a_1, and a_2 perform much worse at reconstructing the bright, low contrast stripes as compared to the dark, high contrast stripes.

6.3.2. Meta Analysis

As an additional analysis, we evaluate the results of 12 different algorithms: five baselines and seven state-of-the-art algorithms as described in our survey paper [56]. For each pixel on the stratified scenes, we compute the median and the minimal absolute disparity error among all algorithms. The median error map serves as an approximate average performance while the minimum error map serves as an upper performance limit.

The median absolute errors are depicted on the top row of Figure 6.18. Regions of high error correlate well with our intended challenges. The spatially increasing difficulty of *Backgammon*, *Dots*, and *Stripes* is clearly visible. The average algorithm struggles mostly with high levels of noise on *Dots*, narrow gaps on *Backgammon*, and low texture on *Stripes*. By contrast, the noise-free area on *Dots*, the well-textured fronto-parallel surface on *Pyramids*, and the well-textured high contrast areas on *Stripes* are accurately reconstructed.

The minimum absolute errors among all algorithms are depicted on the second row of Figure 6.18. There are only minor errors at the most challenging image regions, e.g. those regions with the highest amount of noise on *Dots* or the lowest amount of texture on *Stripes*. These results show that all image regions on the stratified scenes can be solved given the appropriate algorithm and parameterization. As intended, it is mainly the spatial variation within the image that makes these scenes difficult.

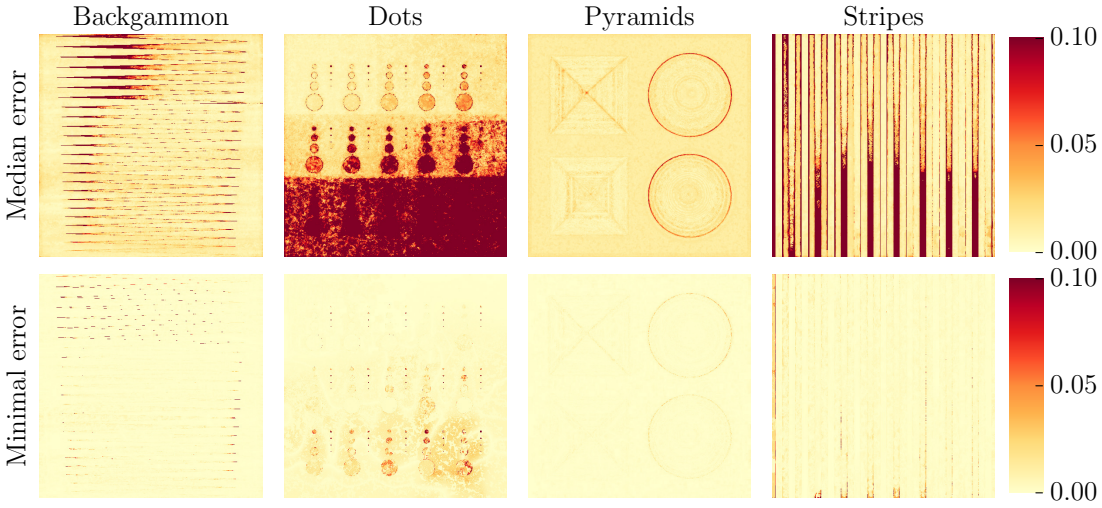


Figure 6.18.: **Average and Top Performance of 12 Algorithms.** We compare absolute disparity errors of 12 state-of-the-art algorithms. On the top row, the median errors per pixel show that the average algorithm struggles at the intended challenge areas on the stratified scenes. On the bottom row, the minimal errors show that all image areas can be solved.

7

Case Study: 4D Light Field Benchmark

As a case study, we present and evaluate the *4D Light Field Benchmark*, a public academic benchmark for depth estimation algorithms on 4D light fields. This benchmark was designed and created in close collaboration with Ole Johannsen from the CVIA group at Konstanz university. We introduced the dataset and evaluation methodology in [44] and presented a taxonomy and survey of light field algorithms in [56]. This chapter is strongly based on the two publications. It consolidates the contributions and puts a stronger focus on the underlying performance evaluation methodology. We use the *4D Light Field Benchmark* as a case study to demonstrate and evaluate how to jointly apply our previously proposed metrics, test data concepts, visualizations, and evaluation methodologies from Sections 4.2 and 5.2 in order to accomplish a comprehensive performance evaluation of light field algorithms.

We first provide a brief benchmark overview. Second, we specify and discuss considerations on benchmark design and explain our design choices. Third, we introduce the actual benchmark, namely the test data, evaluation metrics, and participation modalities. Fourth, we conduct a performance evaluation of five light field algorithms to test the applicability and usefulness of our evaluation methodology.

7.1. Benchmark Overview

For the *4D Light Field Benchmark*, we provide 28 carefully designed, densely sampled, synthetic RGB light fields. All input images are rendered in Blender with a camera setup of 9×9 views in a regular grid (similar to Figure 7.1) and a resolution of 512×512 pixels.

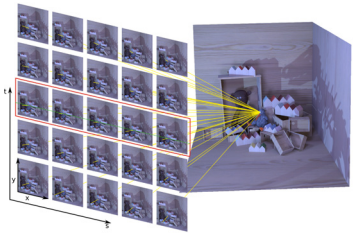


Figure 7.1.: **Example of the Camera Setup.** The light fields are rendered with a regular grid of cameras. For our benchmark, we provide 9×9 views.

As depicted in Figure 7.2, the 12 benchmark scenes are grouped into stratified, test, and training scenes. We further provide 16 additional scenes (see Figure 7.7). For all but the test scenes, we provide reference disparity maps for the center view in regular (512×512) and high (5120×5120) resolution. For region specific evaluation

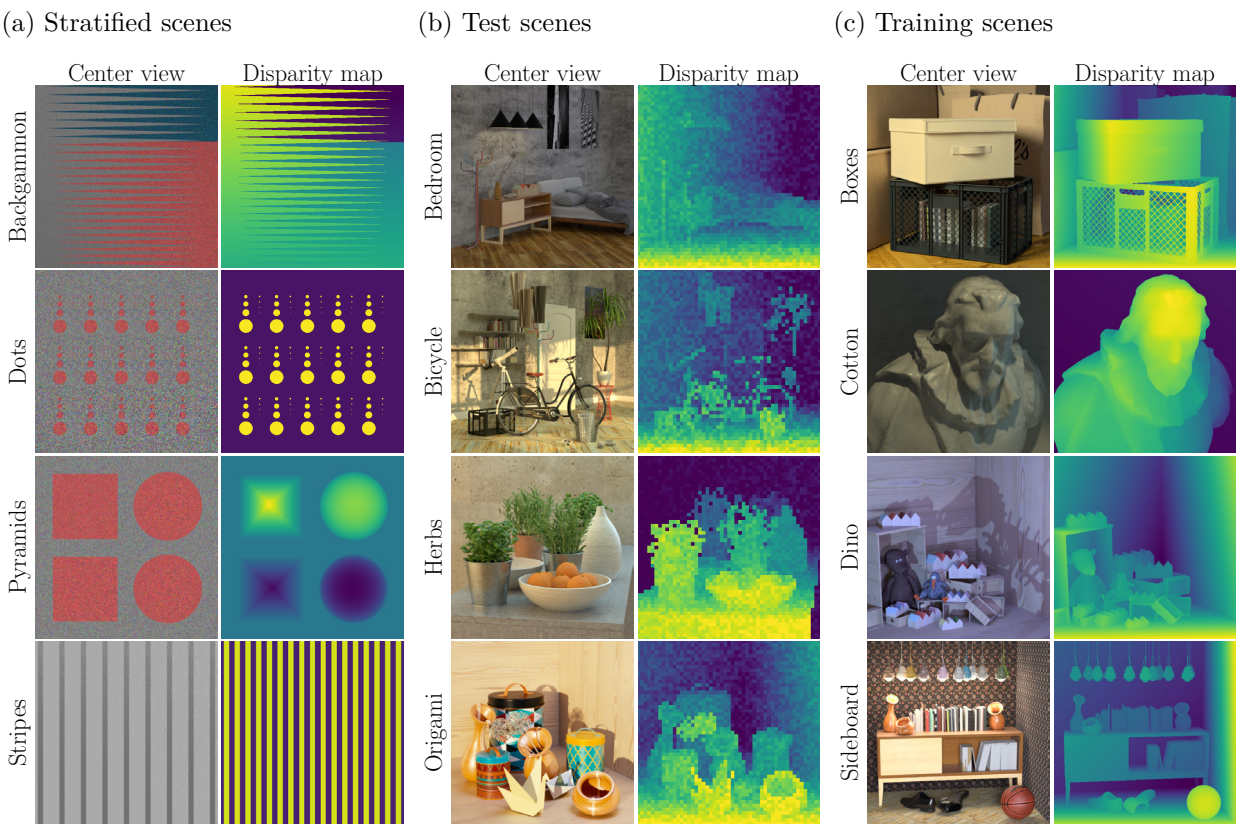


Figure 7.2.: **Benchmark Scenes of the *4D Light Field Benchmark*.** The 12 benchmark scenes consist of four stratified, four test, and four training scenes. We provide high resolution reference disparity maps for the stratified and training scenes. Reference data for the test scenes is not released to avoid overfitting and to ensure a fair evaluation.

at fine structures, discontinuities, and continuous surfaces, we apply adaptations of our proposed geometry-aware evaluation metrics (see Section 4.2). We also compute *MSE* and *BadPix* scores as general evaluation metrics. In addition, we maintain an online benchmark website[1] with currently more than 30 algorithm submissions on the public leaderboard. We support 3D point clouds, radar charts, and scatter plots as interactive visualizations. The source code of our evaluation toolkit and the Blender addon [5] for creating light field datasets is publicly available[2].

7.2. Considerations on Benchmark Design

In order to create a sound and useful benchmark, we follow the evaluation aspects as discussed in Section 2.1. We first define the purpose of the academic benchmark and its evaluation objectives. We then derive what kind of reference data, scene content, evaluation metrics, and ranking is appropriate to achieve this goal. For each evaluation aspect, we review the current situation and identify

[1] `http://www.lightfield-analysis.net` [2018-09-30]
[2] `https://github.com/lightfield-analysis/` [2018-09-30]

important requirements. We then derive and briefly describe our benchmark implementation. More specific technical details of the actual implementation are presented in Section 7.3.

7.2.1. Benchmark Purpose

In recent years, depth estimation from light fields has become an established area of computer vision research. More mature computer vision communities such as stereo or optical flow have benefited greatly from commonly accepted and widely used benchmarks [111, 28, 9]. These benchmarks enabled objective comparison and boosted scientific progress [56]. In the light field community, no comparable benchmark exists. Commonly used error metrics and suitable test datasets with reference data are missing.

For this reason, newly proposed methods often resort to showing qualitative results on a few real-world datasets [124, 57, 141], or to computing quantitative results based on widely varying datasets and error metrics [39, 147, 54, 57, 141, 124]. In both cases, it is difficult to objectively evaluate and compare depth estimation performance between different algorithms.

To overcome these difficulties, we propose a novel light field benchmark. We aim at establishing a commonly accepted evaluation methodology for light field depth estimation algorithms. Our benchmark should help consolidate existing research, enable objective comparisons, and guide the community towards open challenges. In order to support a thorough analysis and to avoid overfitting, our evaluation methodology should explicitly quantify and visualize diverse performance aspects.

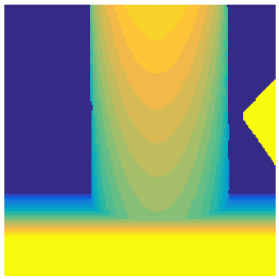
To achieve these goals, we create a publicly available light field dataset with accurate reference data, design a set of complementary performance metrics, and publish a benchmarking website for online performance comparison. To make our evaluation methodology accessible and easy to use, we release the source code for the data generation and the performance evaluation. Thereby, fellow researchers can create additional datasets, reproduce our results, and run additional evaluations with the same metrics and visualizations.

7.2.2. Evaluation Objectives

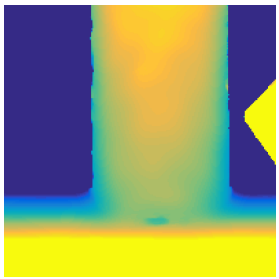
Light field algorithms find more and more real-world applications such as movie set reconstruction or industrial inspection. These applications require a highly accurate depth reconstruction which correctly recovers smooth surfaces, occlusion areas, and fine details.

(a) Center view

(b) Reference disparity map r



(c) Algorithm disparity map a



(d) Difference: $r - a$

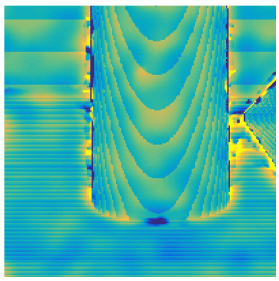


Figure 7.3.: **Issues with Limited Reference Data Accuracy.** The reference data by Wanner et al. [148] is stored in 8 bit, leading to stair-casing artifacts due to the discrete disparity labels (b). Recent algorithms achieve more accurate results (c), limiting the expressiveness of the evaluation (d) (image courtesy: [44]).

Satisfying all of these requirements jointly is challenging for vision-based reconstruction methods. Thanks to the dense and regular sampling of the scene, algorithms for depth estimation from light fields are highly accurate [124, 147, 39, 54, 57] and are capable of taking occlusions into account to recover fine details [141].

For this benchmark, we aim at designing test data and metrics such that the four crucial performance aspects can be assessed and compared. We evaluate for: high accuracy, smooth surfaces, crisp occlusion areas, and coherent fine structures.

7.2.3. Data Generation

As described in Section 2.2.5, the dense and regular light field setup allows for highly accurate depth reconstructions as compared to other computer vision methods. This circumstance renders ground truth creation particularly challenging. This difficulty is reflected by the fact that only one [148] of the available light field datasets [1, 94, 104, 60, 149, 148] provides independently measured ground truth.

Only Wanner et al. [148] provide synthetic and real-world 4D light fields with reference data. However, as discussed in [44], the synthetic input data contains problematic rendering artifacts. Furthermore, the accuracy of the discrete reference disparities has been surpassed by state-of-the-art algorithms (see Figure 7.3).

In order to provide a light field dataset with high quality reference data, we create synthetically rendered scenes and improve upon the work by Wanner et al. [148]. This setup allows us to systematically vary scene parameters and to create near-perfect, high resolution ground truth.

7.2.4. Scene Design

Light field applications depend on depth estimations with high accuracy and good performance at smooth surfaces, discontinuities, and fine structures. The difficulty of reconstructing these geometric entities depends on the local level of texture and noise as well as on the size of fine structures and the orientation of smooth surfaces. In order to properly evaluate algorithm performance, we design our scenes such that these geometric challenges are present with various levels of difficulty.

In contrast to training datasets, evaluation datasets benefit from being compact. A systematic and considerate compilation of images supports specific and meaningful algorithm evaluation [156, 66, 111]. It maximizes information gain for a given benchmarking effort and

keeps dataset creation costs down. In order to achieve this goal, we apply the concept of *stratified scenes* as presented in Section 5.2. We create four puristic scenes with spatially increasing difficulty of the relevant challenges (see Figure 7.2a). This setup allows for decoupled performance analysis and for immediate visual inspection.

Yet, in complex real-world scenarios, these challenges may occur in numerous deviations and combinations. We therefore create additional, photorealistically rendered scenes composed of real-world objects (see Figure 7.2b and 7.2c). These scenes allow to obtain an intuition for real-world algorithm performance.

7.2.5. Metrics and Rankings

Typically, there is no single best algorithm for all applications. Different algorithms and different parameter setups yield very different strengths and weaknesses. Therefore, we use multi-dimensional performance analysis. We apply the geometry-aware metrics as introduced in Section 4.2 to reflect this diversity and to specifically quantify performance for the different requirements.

We deliberately refrain from defining an explicit overall ranking. Instead, we create and provide interactive visualizations on our benchmarking website. Thereby, users are able to select the metrics and frames which correspond to their application priorities. Radar charts highlight the different strengths and weaknesses between algorithms. The leaderboard is initially sorted by algorithm name and may be sorted by the user according to various criteria.

7.3. The 4D Light Field Benchmark

In this section, we first specify the technical details of the data generation for the *4D Light Field Benchmark*. We then describe the data and metrics for the evaluation methodology and explain how we adapted the dataset concepts and metrics from Sections 4.2 and 5.2. Finally, we briefly comment on the algorithms and on the participation modalities.

7.3.1. Technical Dataset Details

All scenes for the *4D Light Field Benchmark* are rendered with Blender [5]. We use the internal renderer for the stratified scenes and the Cycles renderer for the photorealistic scenes. Apart from the $9 \times 9 \times 512 \times 512$ RGB light fields, we provide camera parameters and disparity ranges for each scene. We use a camera setup

(a) Input image region of the *Bedroom* scene (b) Low resolution reference r (c) High resolution reference r (d) Algorithm disparity map a

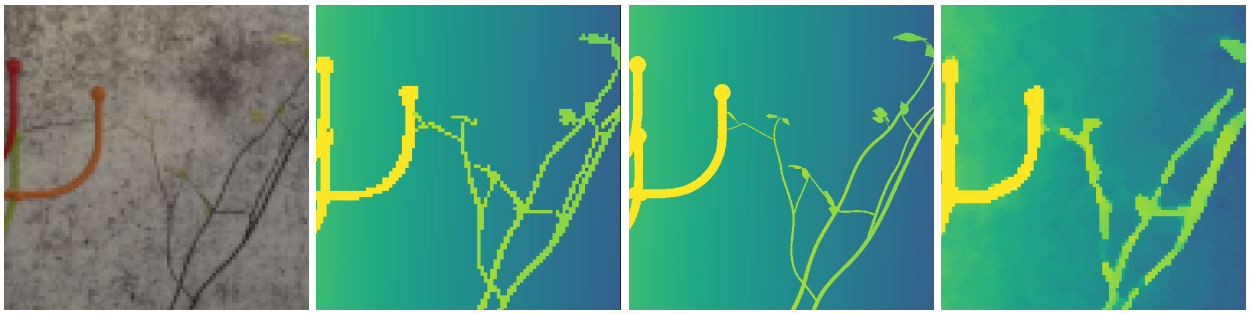


Figure 7.4.: **Disparity Map Resolution.** We provide reference disparity maps in low (512×512) and high (5120×5120) resolution. In order to reduce the impact of artifacts, algorithm performance at fine structures and discontinuities is evaluated on the high resolution reference data.

with parallel optical axes but shift the sensors towards a common focus plane (see [44] for further details). Our setup results in an approximate disparity range of $[-1.5px, 1.5px]$ for most scenes. We further provide ground truth disparity maps and evaluation masks for all but the four test scenes.

As discussed in the previous section, high quality reference data is important but currently missing in the light field community. As shown in Figure 7.3, the accuracy of the discrete disparity labels by Wanner et al. [148] was surpassed by recent light field algorithms. To alleviate this issue, we create 16bit floating point disparity maps.

Furthermore, our reference data must be fit for evaluation at depth discontinuities and fine structures. As depicted on Figure 7.4b, artifacts occur at these image regions on the low resolution disparity maps (512×512 pixels). To minimize the influence of these artifacts and to allow for precise and meaningful evaluation, we create high resolution reference disparity maps at 5120×5120 pixels (Figure 7.4c). For depth reconstruction, applications typically prefer a clear distinction between foreground and background depth. Interpolated depth values at the object boundary do not correspond to any object in the real scene. With our high resolution evaluation, we are able to satisfy this requirement. On the low resolution, an object boundary may go through a pixel which would actually be 60% foreground and 40% background. On our high resolution map, this corresponds to 6 pixels with foreground disparities and 4 pixels with background disparities. The algorithm result is scaled up with nearest neighbor interpolation. Computing *BadPix* scores on this setup favors algorithm results with crisp occlusion boundaries and penalizes interpolated disparity values.

7.3.2. Data and Metrics

The performance evaluation for the *4D Light Field Benchmark* is based on our metrics and test data concepts as proposed in Section 4.2 and 5.2. As discussed in Chapter 5, data and metrics strongly influence each other and should be considered jointly for a thorough performance evaluation. In this section, we explain how we measure performance on the photorealistic and the stratified scenes. For each of the four light field application requirements, we describe the interplay of scene content, evaluation mask, and performance metrics. For all metrics, lower scores are better.

Photorealistic Scenes

For our benchmark, we use four test and four training scenes with photorealistic rendering as depicted in Figure 7.2. The scene content is designed to allow for explicit performance evaluation of continuous surfaces, depth discontinuities, and fine structures.

General Performance. As general evaluation metrics, we use *MSE*∗100 and *BadPix(0.07)* (see Equations 3.5 and 3.2). These metrics are commonly used in the light field community [148]. The scores are computed for all images and on all image regions except for a narrow image boundary of 15px.

High Accuracy. As demonstrated in our algorithm survey [56] and illustrated in Figure 3.4a, *BadPix* scores with a threshold of 0.07 often yield only marginal differences between state-of-the-art light field algorithms. Furthermore, rankings may change considerably depending on the threshold. Due to the level of accuracy of our reference data, we are able to evaluate for high accuracy performance by adding *BadPix(0.03)* and *BadPix(0.01)* scores to the evaluation.

As discussed in [56], we further add the metric *Q25*∗100 as a notion of "best case accuracy". It quantifies the accuracy at the 25th percentile of the disparity estimates on a given scene, i.e. the maximum error on the best 25% disparity estimates.

Continuous Surfaces. To assess performance at planar surfaces, our photorealistic scenes contain walls and floors with various orientations and levels of texture (see Figure 7.2b and 7.2c). The low texture boards and boxes in *Sideboard*, *Bedroom*, and *Boxes* are particularly challenging. For non-planar continuous surfaces, the statue in the *Cotton* scene, the pots and vases in *Origami*, *Sideboard*, and *Herbs*, as well as the spherical oranges and balls in *Herbs* and *Sideboard* feature numerous variations of convex and concave curvature.

To evaluate performance at these image regions, we apply our

(a) Center view (b) Complex occlusions and low texture (c) Complex occlusions and low texture

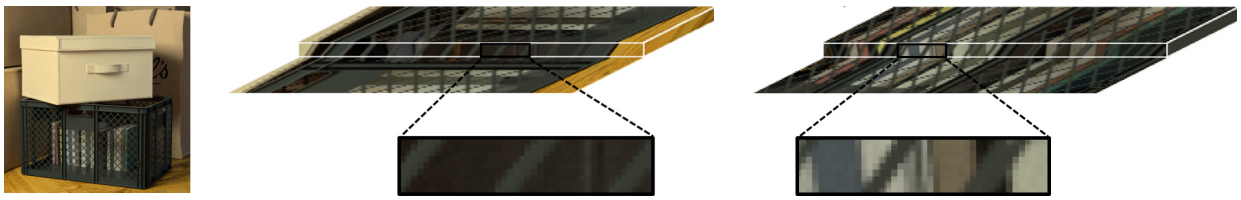


Figure 7.5.: **Challenges on the *Boxes* Scene.** (a) The grid in the *Boxes* scene is particularly challenging due to complex occlusions and low texture. (b, c) The epipolar lines of the books behind the grid are cut off at both ends. Furthermore, the low texture of the books makes the slope of their epipolar lines hard to determine.

metrics as described in Section 4.2.1. We use the *Median Angular Error (MAE)*, an application of the *Angular Error* metric (see Equation 4.1). We further use a simplified version of the *Bumpiness* metric (see Equation 4.4). With H denoting the Hessian matrix of $a - r$, the difference of the estimated algorithm disparity map a and the reference disparity map r, bumpiness is defined as follows:

$$\text{Bumpiness}_{\mathcal{M}}(a, r) = \frac{\sum_{x \in \mathcal{M}} \min(0.05, \|\mathrm{H}(x)\|_F)}{|\mathcal{M}|} * 100. \tag{7.1}$$

$\|\mathrm{H}(x)\|_F$ denotes the Frobenius norm, i.e. the Euclidean norm of the vectorized Hessian matrix at pixel x. All *MAE* and *Bumpiness* scores are computed separately for planar and non-planar continuous surfaces $\mathcal{M}_p$ and $\mathcal{M}_n$.

Discontinuity Regions. Performance at discontinuity regions is assessed on all test and training scenes. Our scenes contain numerous variations with respect to texture, geometry, orientation, and disparity differences, e.g. the silhouette in *Cotton*, the houses in *Dino*, and the low texture discontinuities in *Boxes* (see Figure 7.2). As metric, we compute *BadPix(0.07)* scores on evaluation masks with automatically extracted discontinuity regions.

Fine Structures. To evaluate fine structure performance, our scenes contain structured grids, e.g. in *Boxes* and *Bicycle*. We also add irregular structures of different density, orientation, and thickness, e.g. at the plants in *Herbs*, *Bedroom*, *Origami*, and *Bicycle*, or at the lamp wires in *Sideboard* and *Bedroom* (see Figure 7.2). Geometric entities such as the plants in *Herbs* or the grid in *Boxes* feature complex occlusions. Figure 7.5 illustrates why these image areas are particularly challenging for EPI based light field algorithms. As metrics, we quantify *Fine Fattening* as defined in Equation 4.12. In a similar way, we define *Fine Thinning*:

$$\text{Fine Thinning}_{\mathcal{M}}(a, r, t) = \frac{100}{|\mathcal{M}|} |\{x \in \mathcal{M} \colon (r(x) - a(x)) > t\}|. \tag{7.2}$$

For both metrics, we apply a threshold of $t = 0.15$.

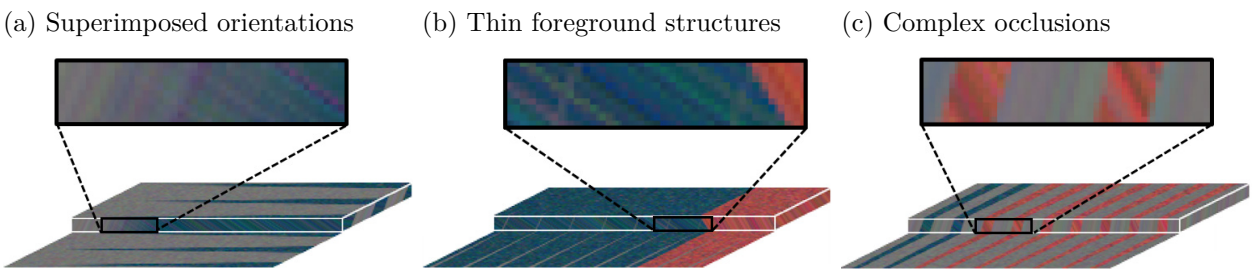


Figure 7.6.: **Challenges on the *Backgammon* Scene.** (a) At the narrow slits, orientations of the foreground and background are superimposed on the EPI images. (b) At the thin peaks, the foreground structure merges with the background. (c) Between the peaks, complex occlusion occurs and epipolar lines of the background are cut off at both ends.

Stratified Scenes

For specific, decoupled performance evaluation, we apply the concept of stratified scenes as described in Section 5.2. We use the *Backgammon*, *Pyramids*, *Dots*, and *Stripes* scenes with spatially increasing difficulties as depicted in Figure 7.2. Figure 7.6 illustrates why the *Backgammon* scene is particularly challenging.

7.3.3. Algorithms

For this case study, we evaluate one baseline algorithm and four state-of-the-art algorithms which were submitted by participants of the *4D Light Field Benchmark*. In addition, we include two artificial algorithms: `PerPixBest` and `PerPixMedian`. An in-depth evaluation of 14 algorithms is provided in our survey paper [56]. Interactive evaluations of more than 30 algorithms are available online on our benchmark website. In this section, we provide brief descriptions of the five algorithms as depicted in Table 7.1. For more detailed algorithm descriptions, we refer to the respective algorithm publications and to the taxonomy by Ole Johannsen and Bastian Goldlücke in our survey paper [56].

As a baseline, we include the multi-view algorithm `LF` [54]. A cost volume is built based on the similarity of input image patches and image gradient patches. A multi-label graph cut optimization with 100 labels is performed to derive a discrete disparity map which is then iteratively refined by fitting quadratic functions. `OBER` [113] is based on sparse, subpixel accurate line fits on the epipolar plane images. For each pixel, it then iteratively minimizes the variance along the EPI lines and a smoothness metric which is based on bilateral filtering. `OBER` uses only the central horizontal line of 9 cameras. `OFSY` [119] utilizes focal stack symmetry to build a cost volume with 330 labels. Sublabel accurate optimization is performed to derive an initial disparity map which is then refined taking surface normal orientations into account. `SC_GC` [116] is based on similarity of the angular patches. A disparity map is derived from the result-

Algorithm	Data term	Camera setup	Occlusion handling	Cost volume	Refinement
LF [54]	multi-view	full	implicit	100	iterative quadratic refinement
OBER [113]	EPI images	horizontal	implicit	-	bilateral filtering
OFSY [119]	focal stack	crosshair	explicit	330	surface normal regularization
SC_GC [116]	angular patches	full	explicit	256	local plane fitting
SPO [160]	EPI images	crosshair	implicit	256	guided filtering

Table 7.1.: **Light Field Algorithm Overview.** For our case study, we include the baseline algorithm LF and four state-of-the-art light field algorithms with different approaches for disparity map estimation.

ing cost volume with 256 labels. Occlusions are taken into account explicitly. As refinement, local planes are fitted to the disparity estimates. SPO [160] is based on estimating orientations in the epipolar plane images. A cost volume with 256 labels is created based on histograms from the two sides of the estimated epipolar lines. Cost volume regularization is performed individually for each depth label. The disparity with the lowest cost is used for the final result without further refinement. The artificial PerPixBest algorithm is used as an approximate upper performance limit. For each pixel, we compare the disparity estimates of all five algorithms and keep the disparity with the lowest absolute error. Similarly, the artificial PerPixMedian algorithm is used as an approximate average performance. For each pixel, we take the disparity estimate with the median disparity error of all five algorithms.

7.3.4. Benchmark Modalities

To be listed on the public benchmark table, participants must submit estimated disparity maps of the center views for the 12 benchmark scenes as depicted on Figure 7.2. The same parameter settings must be used for all scenes. As input, participants may use the provided disparity ranges and any subset of the $9 \times 9 \times 512 \times 512$ RGB input images. The reference data of the training, stratified, and additional scenes (see Figure 7.7) may be used to optimize parameter settings. The reference data for the four test scenes is kept hidden.

Upon submission, all scores and visualizations are computed on our evaluation server and the results are added to the public leaderboard. The submitted disparity maps of the training and stratified scenes are made available for download to allow other researchers to conduct further experiments.

As methods improve, our benchmark setup may lose in expressiveness to differentiate algorithm performance. Therefore, we update the benchmark regularly and actively involve the community through

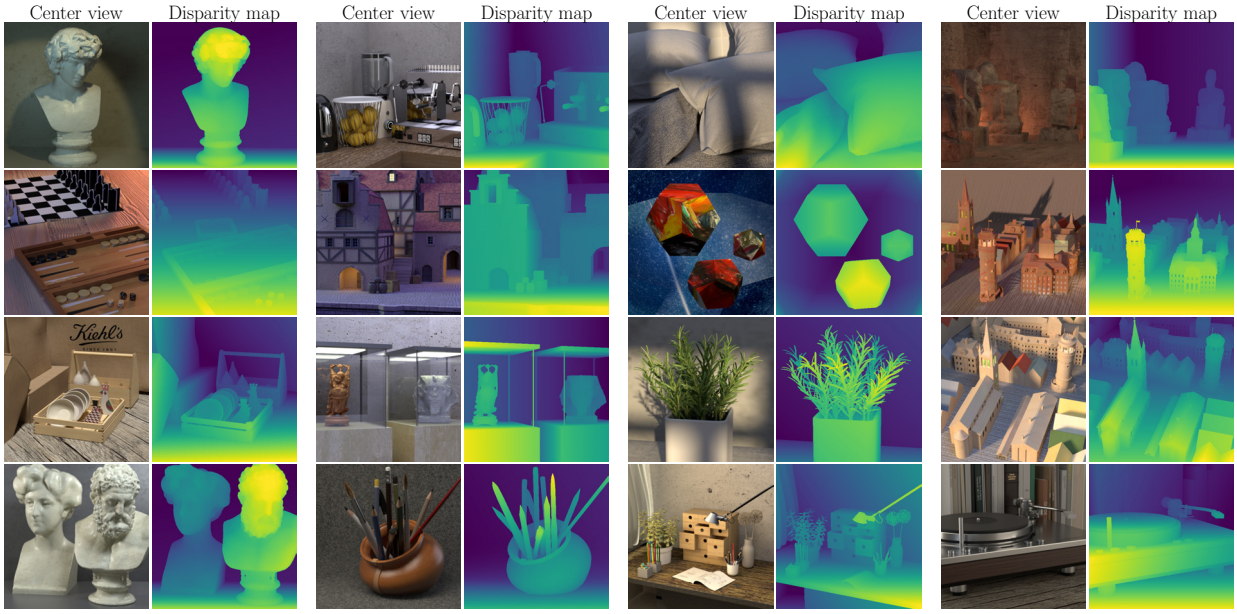


Figure 7.7.: **Additional Scenes.** We provide 16 supplemental photorealistic scenes with input and reference data. They are not part of the official benchmark but can be used for algorithm development and additional experiments.

workshops and mailing lists. Upon request from fellow researchers, we created four complementary additional scenes. The metrics *Q25*, *BadPix(0.03)*, *BadPix(0.01)*, and the *MAE* were added to the benchmark after accuracy of the submissions had increased [56].

7.4. Experimental Validation of Data and Metrics

In this section, we evaluate the performance of five light field algorithms. We first apply the prevalent evaluation methodology by comparing *MSE* and *BadPix(0.07)* scores. We then apply our proposed evaluation methodology to test its applicability and usefulness. We check whether it reveals additional insights about relative algorithm strengths and weaknesses with respect to high accuracy, continuous surfaces, depth discontinuities, and fine structures, as well as their overall performance profiles.

7.4.1. General Performance

The most common quantitative performance evaluation in the light field community consists of reporting *MSE* and *BadPix(0.07)* scores for a given set of scenes. Table 7.2 depicts these scores for the test and training scenes of our benchmark. `SPO` performs best on the vast majority of scenes. It also features the best mean and median

	MSE					*BadPix(0.07)*				
	LF	OBER	OFSY	SC_GC	SPO	LF	OBER	OFSY	SC_GC	SPO
Mean	9.1	4.2	6.8	4.7	**3.8**	17.2	9.5	10.4	10.2	**8.2**
Median	8.0	1.9	3.1	2.0	**1.7**	18.6	9.0	11.3	10.2	**8.8**
Boxes	17.4	**6.5**	9.6	12.3	9.1	23.0	16.4	19.2	18.6	**15.9**
Cotton	9.2	**0.8**	2.7	1.9	1.3	7.8	**1.8**	3.0	6.3	2.6
Dino	1.2	0.5	0.8	0.5	**0.3**	19.0	2.9	3.4	6.0	**2.2**
Sideboard	5.1	1.8	2.5	1.8	**1.0**	22.0	**8.1**	10.4	9.7	9.3
Bedroom	0.5	0.4	0.4	0.3	**0.2**	13.9	6.9	**4.7**	5.1	4.9
Bicycle	11.7	7.0	10.9	5.9	**5.6**	19.8	17.1	16.1	12.2	**10.9**
Herbs	21.3	14.5	24.4	13.0	**11.2**	18.1	10.0	12.3	12.7	**8.3**
Origami	6.8	2.1	3.6	2.2	**2.0**	14.2	13.1	13.7	**10.7**	11.7

Table 7.2.: ***MSE* and *BadPix(0.07)* Scores.** The prevalent evaluation methodology identifies `SPO` as the best algorithm since it features the lowest mean and median scores for *MSE* and *BadPix(0.07)*.

scores on both metrics. With these results, `SPO` would typically be declared the best performing algorithm, potentially discarding all other algorithms. A closer look at the table reveals additional information about the algorithms and scenes. For many scores, `OBER`, `OFSY`, and `SC_GC` yield only a small performance difference to `SPO`. The baseline algorithm `LF` is clearly outperformed by the more recent algorithms. From a scene perspective, *Cotton*, *Dino*, and *Bedroom* can be identified as rather easy scenes in terms of general accuracy. *MSE* scores are very high for *Herbs*, which is probably due to the high disparity range of this scene. *BadPix(0.07)* scores are high for *Boxes* and *Bicycle*, which may be caused by relatively large image regions with complex occlusions.

7.4.2. High Accuracy

To evaluate high accuracy performance, we apply our metrics *BadPix(0.03)*, *BadPix(0.01)*, and *Q25*. As depicted on the top row of Figure 7.8, results look similar for *BadPix(0.07)*. Errors occur mostly at the depth discontinuities of the statue's silhouette. `OBER` performs best, closely followed by `SPO` and `OFSY`. However, looking at the *BadPix(0.01)* and *Q25* results on the second and third row reveals that `OFSY` produces by far the most accurate disparity maps. It provides highly accurate estimates at the planar background and the low curvature areas of the statue. By contrast, `SPO` features the worst high accuracy performance even though it was identified as the best performing algorithm at *MSE* and *BadPix(0.07)*. `SPO` is based on discrete disparity labels and does not perform disparity refinement. Hence, `OFSY` may be a better choice as compared to `SPO` when highly accurate results are required.

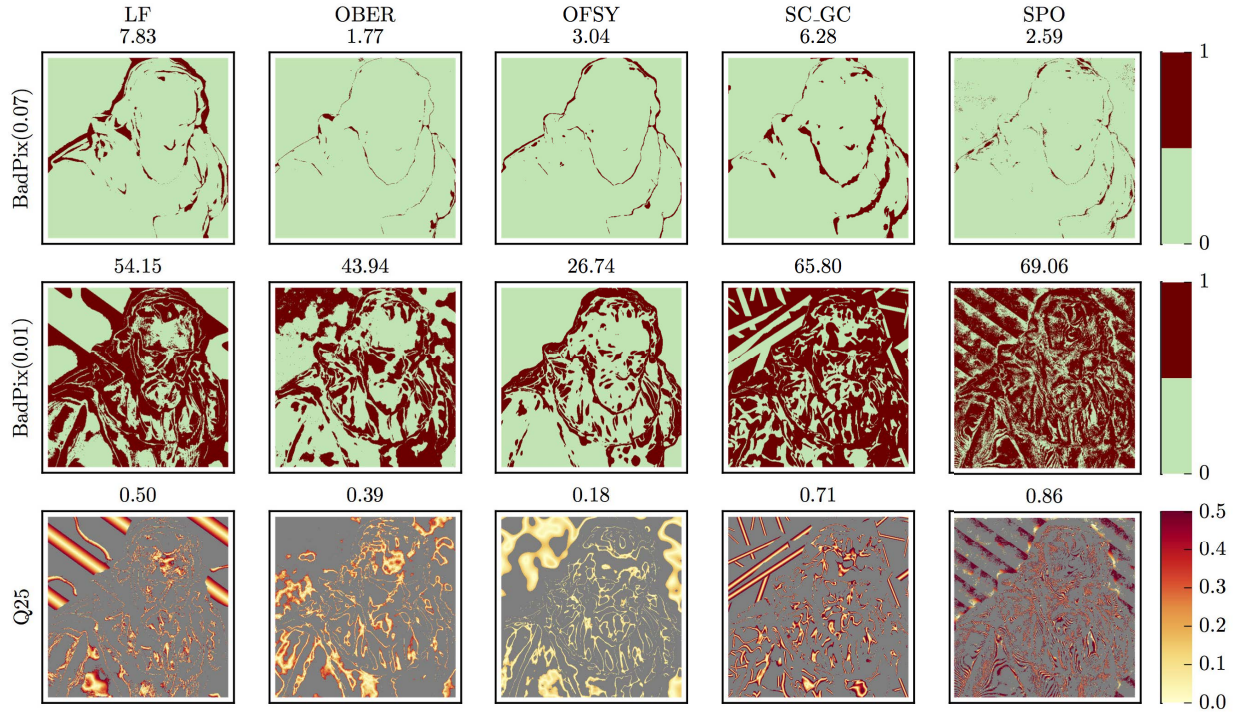


Figure 7.8.: **High Accuracy Evaluation.** Dark red areas on the first and second row indicate disparity errors above $t = 0.07$ and $t = 0.01$ respectively. Light green areas are considered correct according to the respective metric. On the bottom row, absolute disparity errors $|a - r|$ are shown for all pixels whose error is below the 25th error quantile of the given algorithm. Dark red indicates high errors, light yellow indicates low errors. Pixels with disparity errors above the 25th error quantile are depicted in gray.
BadPix(0.07) results on the top row look rather similar for all algorithms. Errors occur mostly at strong depth discontinuities. The *BadPix(0.01)* and *Q25* results on the second and third row reveal that `OFSY` produces highly accurate disparity maps, clearly outperforming all other algorithms. `SPO` scored best on the general evaluation but performs worst for high accuracy.

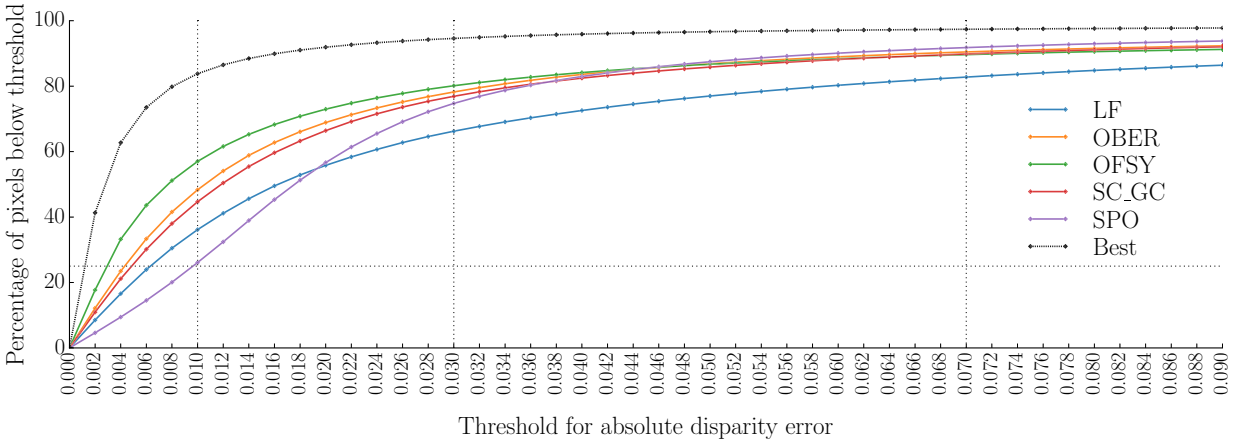


Figure 7.9.: **Impact of the Error Threshold.** The percentage of correct disparity estimates on the test and training scenes is plotted for increasing error thresholds. Scores are very similar for $t = 0.07$ and above. For stricter thresholds, relative algorithm ranks change considerably. `OFSY` performs best at strict thresholds. `SPO` performs worst at strict thresholds but outperforms all algorithms for bigger thresholds.

Figure 7.9 illustrates how relative algorithm performance changes for a range of thresholds. The vertical lines illustrate our choice of *BadPix* thresholds. At the common $t = 0.07$, relative algorithm performance is very similar. By contrast, scores and relative rankings differ considerably for stricter thresholds. SPO outperforms all algorithms for thresholds of $t \geq 0.05$ but relative performance decreases sharply for stricter thresholds. By contrast, the percentage of bad pixels decreases steadily with looser thresholds for OFSY, OBER, and SC_GC. OFSY performs best for strict thresholds but is outperformed by all other non-baseline algorithms for bigger thresholds.

7.4.3. Continuous Surfaces

To evaluate performance at continuous surfaces, we apply our *MAE* and *Bumpiness* metrics. As depicted on the top row of Figure 7.11, the disparity maps look almost indistinguishable. By contrast, the normal maps in the second row reveal insights about algorithm peculiarities. SPO scores worst on the *MAE* and *Bumpiness* metrics. SPO does not apply smoothing post-processing steps, resulting in a noisy normal map. OFSY applies a refinement step which incorporates the normal orientations. This leads to very smooth surfaces and top *MAE* scores for both, planar and non-planar surfaces. OBER also performs well on continuous surfaces. The results are slightly noisier than those of OFSY but the surface orientations are mostly accurate. Hence, OFSY and OBER are the most suitable algorithms when smooth surfaces are required. SPO may benefit considerably from a smoothing post-processing step.

(a) Center view

(b) Reference disparity map r

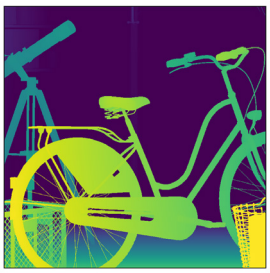


Figure 7.10.: **Detail of the *Bicycle* Scene.** The complex geometry of the bicycle and the poorly textured door in the background are challenging for depth estimation algorithms.

7.4.4. Depth Discontinuities

To evaluate performance at depth discontinuities, we compute *BadPix(0.07)* scores at a discontinuity evaluation mask. Figure 7.12 depicts an image area of the *Bicycle* scene which is particularly challenging. The door in the background features little texture (see Figure 7.10a). The narrow gaps and thin structures of the bicycle yield diverse depth discontinuities with complex occlusions.

The *Bicycle* scene reveals a notable advantage of the SC_GC method. SC_GC is capable of fitting a joint plane to background areas which are partially occluded by the bike. As a result, it features very crisp discontinuities and almost no thinning. The histogram approach of SPO also works well for occlusion regions and creates only few artifacts. By contrast, LF uses only few labels and applies a strong regularization, leading to severe edge fattening.

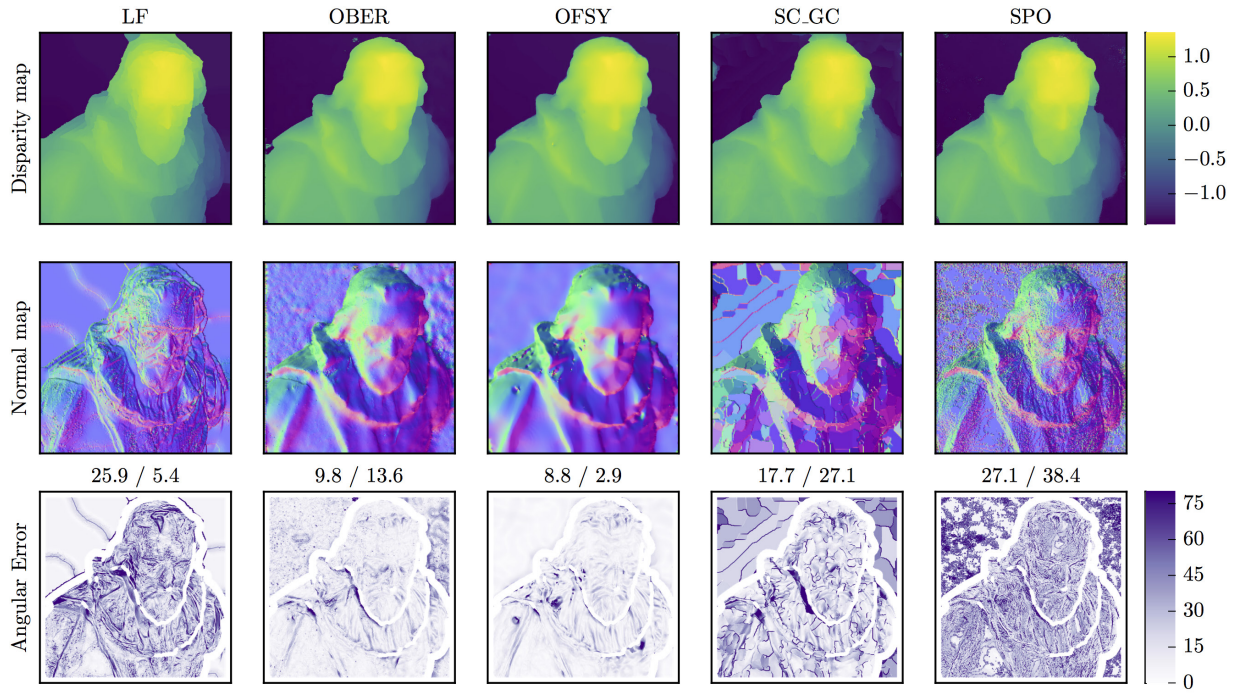


Figure 7.11.: **Evaluation at Continuous Surfaces.** The disparity maps look similar for all algorithms. By contrast, the normal maps and *MAE* metric visualizations reveal insights about algorithm peculiarities. The *Mean Angular Error* scores at the bottom row indicate performance at non-planar / planar surfaces respectively.
The locally fitted plane patches of `SC_GC` are clearly visible. The missing refinement step of `SPO` leads to noisy results and poor scores. By contrast, the normal-based refinement of `OFSY` leads to very smooth results which is adequately quantified by our metrics.

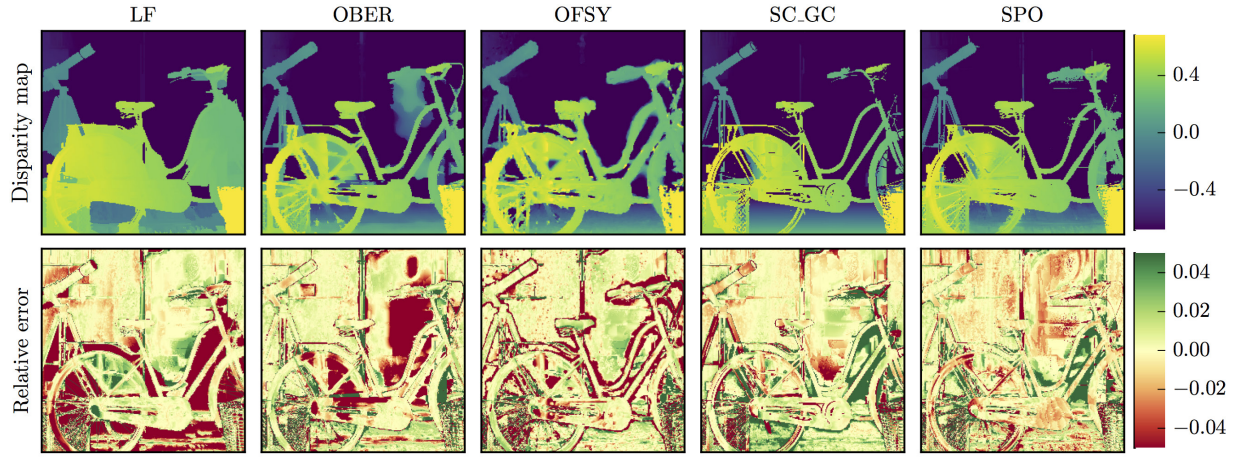


Figure 7.12.: **Evaluation at Depth Discontinuities.** Accurately reconstructing depth discontinuities is challenging at the narrow gaps and thin structures of the bicycle in front of the poorly textured door (compare Figure 7.10a). On the second row, green colors indicate above-average performance and red colors indicate below-average performance. `SC_GC` and `SPO` perform well at producing crisp discontinuities while `LF`, `OBER`, and `OFSY` suffer from severe fattening and blurry discontinuities.

The discontinuities of OFSY are blurry. Apparently, OFSY's explicit occlusion handling does not work very well on this challenging image area. OBER produces mostly crisp disparity edges but yields some fattening artifacts. The most severe "foreground bleeding" occurs in vertical directions at the low texture door in the background. This issue may be improved by taking vertical views into account.

The second row of Figure 7.12 depicts our MedianDiff visualization which highlights the local relative performance among the five algorithms. For each pixel, we compare the median absolute disparity error of the five algorithms with the absolute error $|r - a_i|$ of the current algorithm a_i. Pixels where a_i performs better than the median error are depicted in green. Pixels where a_i performs worse are depicted red. This visualization highlights that SC_GC and SPO feature above average performance at the bicycle boundaries, while OFSY produces general edge fattening, and OBER suffers from strong artifacts at the low-texture door.

7.4.5. Fine Structures

We compute *Fine Thinning* and *Fine Fattening* scores to quantify the sensitivity and specificity at fine structures. As depicted in Figure 7.14, reconstructing coherent fine structures which meet both requirements is challenging for most algorithms.

SC_GC and SPO feature little *Fine Fattening* but strong *Fine Thinning*. On SPO, the plant is severely fragmented but some parts are reconstructed. On SC_GC, the plant is almost completely lost and even one of the hooks on the left is lost. However, the discontinuities of SC_GC are very crisp. We assume that both phenomena can be attributed to the plane fitting refinement.

OBER and OFSY both produce rather fuzzy reconstructions of the fine structures. They yield low *Fine Thinning* but severe *Fine Fattening*. For OBER, the thinning is mostly occurring at horizontal structures. This is probably due to OBER's camera setup. Incorporating views of the vertical direction may improve the *Fine Thinning* performance considerably.

For reconstructions with maximum sensitivity and at least moderate accuracy, OBER appears to be the most suitable algorithm. For an application where fragmented reconstructions are good enough but fattening is detrimental, SPO would be the best choice. For all algorithms, adjusting the influence of the regularization may further improve performance at fine structures towards the desired characteristics.

(a) Center view

(b) Reference disparity map r

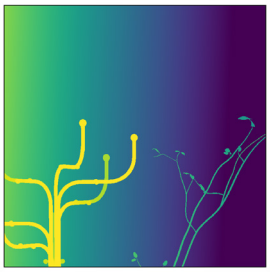


Figure 7.13.: **Detail of the *Bedroom* Scene.** Evaluating on high resolution is crucial for the very thin twigs of the plant.

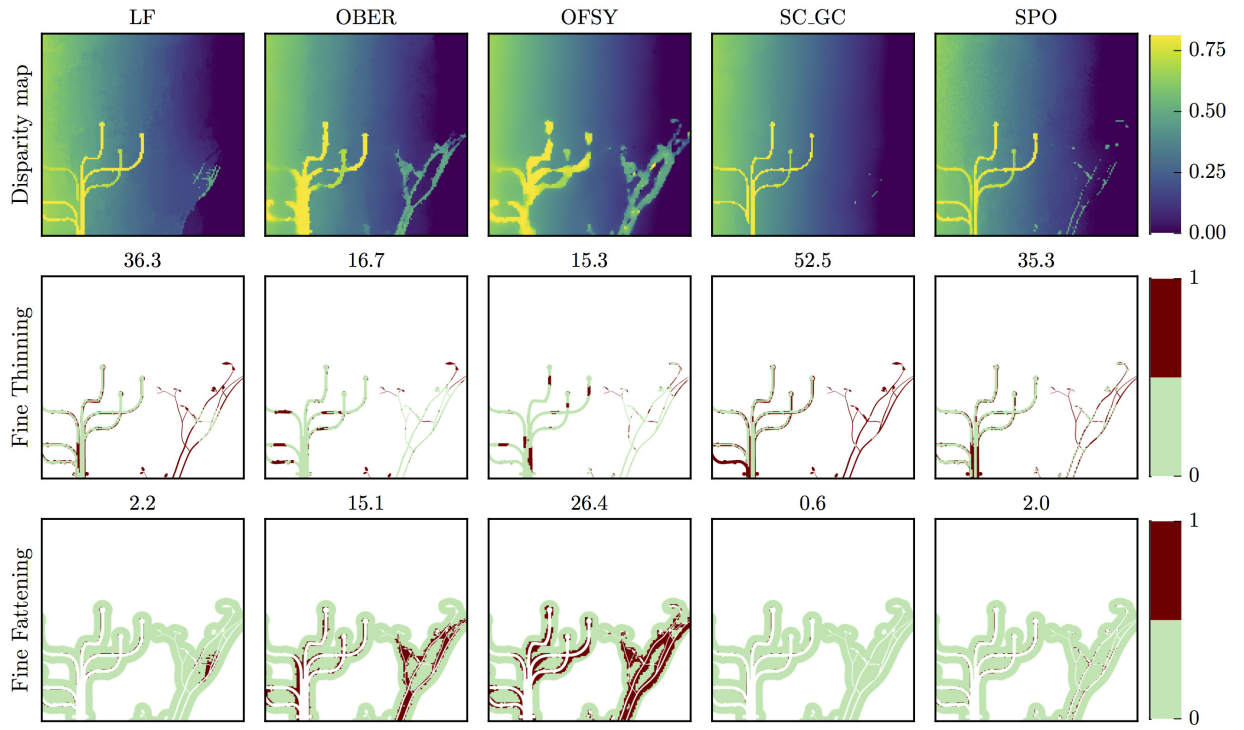


Figure 7.14.: **Evaluation at Fine Structures.** We use a crop of the *Bedroom* scene (see Figure 7.2b) to highlight performance differences at fine structures. The input image and reference disparity map r of the crop are depicted on Figure 7.13. `SC_GC` and `SPO` yield almost no *Fine Fattening* but severe *Fine Thinning*. For `SC_GC` the thin twigs of the plant are almost completely lost. By contrast, `OBER` and `OFSY` are more sensitive to the fine structures but suffer from severe *Fine Fattening*.

7.4.6. Overall Performance

The metrics and visualizations of the previous sections revealed specific and in-depth insights into individual strengths and weaknesses of the five algorithms. In order to identify the best algorithm for a specific application, a comprehensive overview of the performance aspects is required. We argue that averaging metrics with different meanings and scales to obtain a final ranking is not the best solution. Instead, we make use of radar charts to put the relative performance differences into context.

Figure 7.15 depicts a radar chart with median scores based on all test and training scenes. Each axis denotes one metric; each line represents one algorithm. Lower values towards the center indicate better performance. The radar chart provides a good overview of the relative strengths and weaknesses of the different algorithms. It reveals that none of the algorithms clearly outperforms the other algorithms on all evaluation aspects.

The general evaluation based on *MSE* and *BadPix(0.07)* scores at the beginning of this section identified `SPO` as the best performing algorithm. The radar chart highlights the small margin by which `SPO` outperforms the other algorithms on the *MSE* and *BadPix(0.07)* axis. It also clearly shows that all other algorithms outperform `SPO` when it comes to highly accurate results and good reconstructions

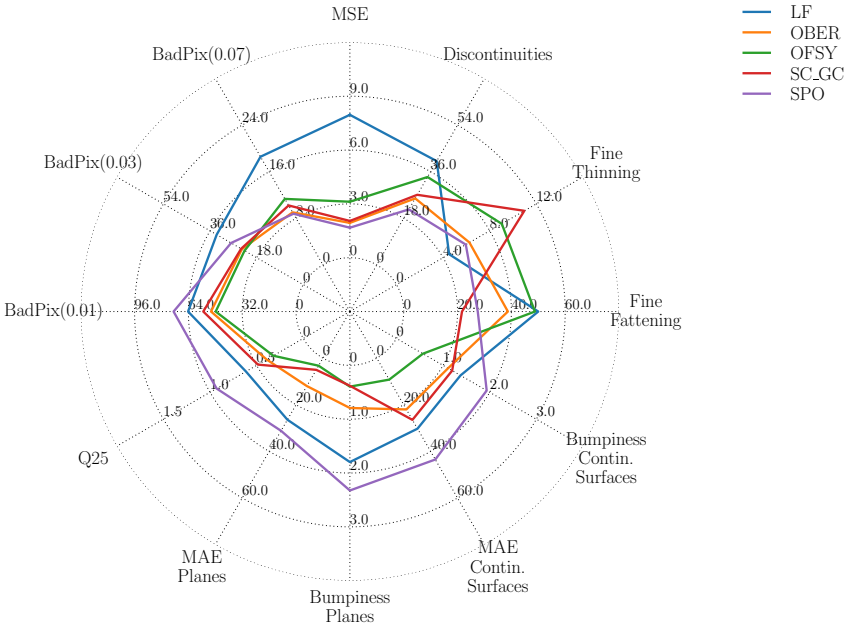


Figure 7.15.: **Multi-Dimensional Performance Evaluation.** The radar chart provides an overview of all median scores across the test and training scenes. Each axis represents one metric. Lower values towards the center are better.
The radar chart puts relative performance differences into context. For example, it reveals that `SPO` is good at discontinuities but poor at continuous surfaces while `OFSY` has opposite strengths and weaknesses.

at continuous surfaces. The biggest strength of `SPO` are complex discontinuities. It scores best at discontinuity regions and features the best trade-off between *Fine Fattening* and *Fine Thinning*. These image regions are also where high errors are most likely to occur, negatively effecting *MSE* and *BadPix(0.07)* scores.

The radar chart reveals that `OFSY` features almost the opposite performance profile. `OFSY` scores best at planar and non-planar surfaces which is probably due to the high number of labels, the sublabel-refinement, and the normal map regularization. The accurate surface reconstruction also leads to top scores on the high accuracy metrics *Q25*, *BadPix(0.03)*, and *BadPix(0.01)*. However, the explicit occlusion handling seems to require further improvement as `OFSY` performs poorly at discontinuities, *Fine Fattening*, and *Fine Thinning*.

The local plane fitting of `SC_GC` leads to good scores at the two plane metrics. However, `OFSY` produces an even better planar surface reconstruction. `SC_GC` features moderate performance at non-planar surfaces, the general evaluation metrics, and for high accuracy.

The radar chart suggests that `OBER` is the most general algorithm. It does not yield outstanding results on individual metrics but there is also no single strong outlier with poor performance. All other algorithms have at least one considerable weakness.

7.5. Conclusion and Outlook

In this case study, we demonstrated that our proposed metric definitions, test data concepts, and visualizations can be applied to design and implement a comprehensive light field benchmark.

When applying the prevalent evaluation methodology, we identified `SPO` as the best performing algorithm because it featured the lowest average *MSE* and *BadPix(0.07)* scores on almost all scenes. We showed that our benchmark provides additional insights into algorithm performance for high accuracy, continuous surfaces, depth discontinuities, and fine structures. It revealed that `SPO` performs best at discontinuity regions but rather poorly at continuous surfaces. `OFSY` was identified as being most suitable for accurate continuous surfaces but not for fine structures. `OBER` was the most robust algorithm with the best overall performance.

Our publicly available dataset fills a gap for high quality light field data with reference data. We showed that our diverse scene content, high resolution reference data, and corresponding evaluation masks allow for specific quantitative performance evaluation, even at discontinuities and fine structures. Our metrics and visualizations, as proposed in Section 4.2, could easily be adjusted and applied to fit the slightly different light field setup, thus satisfying requirements R4 and R5. We showed that the metric scores adequately quantify the relevant phenomena. The metric visualizations support further qualitative understanding of algorithm peculiarities. Radar charts provide good performance overviews and allow for multi-dimensional performance comparisons.

Overall, our combination of scenes, metrics, and visualizations provides helpful tools to gain insights about algorithm performance. They support the selection of the most suitable algorithm for a given application, the parameter optimization for a given algorithm, and provide useful directions on which algorithm approaches could be combined to create better algorithms.

As future work, our metrics may be adjusted and the dataset may be extended. For applications which require accurate surface reconstructions, the simplified *Bumpiness* metric (Equation 7.1) could be replaced by the more specific *Smoothing* and *Bumpiness* metrics (Equation 4.5 and 4.4) to explicitly penalize smoothing at high curvature regions. If fine structures are important, our evaluation masks could be split into non-grid and grid-like fine structures. This would allow for a more specific quantification of *Fine Thinning*.

Concerning the dataset, our rendering setup may be used to create

a bigger dataset which is more suitable for training learning-based algorithms. A randomized object placement similar to the *Flying Things* by Mayer et al. [82] may be used. Furthermore, synthetic data with more challenging, non-Lambertian surface materials as well as real data would provide additional challenges and evaluation aspects.

Overall, our *4D Light Field Benchmark* successfully implements and satisfies the benchmark purposes and evaluation objectives as described in sections 7.2.1 and 7.2.2. On our benchmark website and with our publicly available dataset and evaluation toolkit, we enable objective comparisons of detailed algorithm performance profiles. More than 30 algorithm submissions and more than 700 downloads of the dataset suggest that our *4D Light Field Benchmark* is about to establish a widely accepted and commonly used evaluation methodology.

8 Conclusion

We showed that prevalent metrics struggle with quantifying performance differences at image regions which are highly relevant for stereo applications. To alleviate this problem, we introduced geometry-aware performance metrics and data generation concepts which allow for meaningful and comprehensive evaluations of depth estimation algorithms. We provide a summary of our contributions in Section 8.1. Future research directions are described in Section 8.2.

8.1. Summary

In Chapter 2, we showed that performance evaluation of computer vision algorithms is affected by numerous aspects such as the requirements and priorities of the application domain, the characteristics of the input and reference data, as well as the choice of error metrics and their visualizations.

In Chapter 3, we provided comprehensive surveys of existing error metrics and test data concepts for depth estimation algorithms. We derived a user-friendly metric taxonomy based on data availability. We described the prevalent per-pixel evaluation methodology and explained notable but rarely used methods for evaluating with sparse, weak, or missing reference data. We concluded that most error metrics are either widely established but too general for meaningful performance profiles or very specific but not widely applicable.

In Chapter 4, we addressed this issue by proposing metrics which explicitly take into account local data characteristics and application-specific priorities while still being widely applicable and easy to implement. We first derived five requirements for expressive, complementary, semantically meaningful, widely applicable, and easy to use performance metrics. Following these requirements, we introduced geometry-aware stereo metrics to quantify algorithm performance at continuous surfaces, depth discontinuities, and fine structures. For each of these geometric entities, we motivated their practical relevance and proposed specific metrics to formally quantify algorithm

performance. At continuous surfaces, we quantified the *Angular Error*, *Bumpiness*, and *Smoothing*. At discontinuities, we quantified *Foreground Fattening* and *Foreground Thinning*. At fine structures, we quantified *Porosity*, *Detail Fattening*, and *Fragmentation*.

In Chapter 5, we proposed methods for obtaining meaningful performance evaluations despite reference data deficiencies. First, we described how to apply our metrics to sparse and weak reference data. Second, we introduced the concept of *stratified scenes*. We proposed a systematic and puristic scene design with spatially increasing difficulty to allow for specific evaluation of isolated algorithm challenges. We demonstrated the design principles by deriving four stratified scenes which feature combinations of challenging fine structures, slanted surfaces, noise, texture, and contrast. In Section 6.3 we demonstrated that our scenes allow for insightful visual comparison as well as automated quantitative degradation analysis.

In Chapter 6 we performed a twofold evaluation of the proposed metrics. First, we used systematic test cases to evaluate the specificity and complementary expressiveness of our metrics. We showed that our metric scores and visualizations adequately reflect meaningful performance differences which are not quantified by *RMS* or *BadPix* scores. Second, we performed a user study to test whether our metrics conform with visual performance assessment. High ranking agreement between human graders showed that surface and fine structure quality are visually discernible concepts. We found that intra- and inter-grader consistency is low when assessing depth discontinuity performance. Rankings obtained from our metrics conformed with human assessment on performance aspects that featured high human agreement.

In Chapter 7, we presented a case study which demonstrated the applicability and customizability of our proposed stereo metrics for evaluating light field algorithms. We described the design, implementation, and application of the *4D Light Field Benchmark* to demonstrate the joint application of our proposed evaluation metrics and test data concepts. We showed how considerations on evaluation objectives and application priorities affected our scene design, reference data creation, and metric selection. Our proposed visualizations and metrics revealed specific strengths and weaknesses of five light field algorithms. We identified considerable performance differences with respect to high accuracy, continuous surfaces, depth discontinuities, and fine structures which were not quantified by previous evaluation methods.

8.2. Future Research Directions

We discussed outlooks and specific next steps with respect to our geometry-aware metrics and the *4D Light Field Benchmark* in Sections 6.2.6 and 7.5. In this section, we discuss three more general research directions that we consider particularly valuable.

First, algorithm evaluation should ideally be performed on a component level. As described in Section 2.1.3, we maximized the applicability of our evaluation methodology by using the final algorithm results without requiring access to algorithm source code. However, individual processing steps such as smoothing considerably affect algorithm performance. Therefore, algorithms should ideally be evaluated on a modular basis in order to truly understand the impact of different algorithmic components. Instead of identifying the most suitable algorithm based on final results, we could then identify the most suitable combination of components for a given application.

Second, our methodology could be extended for optical flow evaluation. This thesis is focused on evaluating depth estimation results. Similar application requirements and geometric challenges apply for optical flow estimation. Our geometry-aware metrics and test data concepts could be extended to quantify the surface quality of optical flow results, as well as their accuracy at motion discontinuities and their sensitivity to fine structures.

Third, apart from geometric challenges, radiometric challenges such as low contrast or reflections should be taken into account more explicitly. With our stratified scenes, algorithms can be evaluated with respect to both types of challenges. The proposed geometry-aware metrics explicitly take local scene geometry into account but they are agnostic to radiometric challenges. Our evaluation of stereo datasets [154] and our recently published *WildDash* segmentation benchmark [157] showed that labels for radiometrically challenging image regions allow for valuable insights into algorithm performance and robustness. The region labels were obtained via cumbersome and potentially error-prone manual annotation. As future work, many of these regions and their level of difficulty should be identified automatically. This would allow for meaningful performance analyses on any dataset and with respect to both, geometric and radiometric, challenges.

A

List of Co-Authored Publications

Parts of this work were previously published as:

[65] Kondermann, D., Nair, R., Meister, S., Mischler, W., Güssefeld, B., **Honauer, K.**, Hofmann, S., Brenner, C., and Jähne, B. "Stereo Ground Truth with Error Bars". In: *Asian Conference on Computer Vision*. Vol. 9007. Springer International Publishing, 2015, pp. 595–610. DOI: 10.1007/978-3-319-16814-2_39.

[46] **Honauer, K.**, Maier-Hein, L., and Kondermann, D. "The HCI Stereo Metrics: Geometry-Aware Performance Analysis of Stereo Algorithms". In: *International Conference on Computer Vision*. IEEE. 2015. pp. 2120–2128. DOI: 10.1109/ICCV.2015.245.

[66] Kondermann, D., Nair, R., **Honauer, K.**, Krispin, K., Andrulis, J., Brock, A., Güssefeld, B., Rahimimoghaddam, M., Hofmann, S., Brenner, C., and Jähne, B. "The HCI Benchmark Suite: Stereo And Flow Ground Truth With Uncertainties for Urban Autonomous Driving". In: *Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2016, pp. 19–28. DOI: 10.1109/CVPRW.2016.10.

[32] Güssefeld, B., **Honauer, K.**, and Kondermann, D. "Creating Feasible Reflectance Data for Synthetic Optical Flow Datasets". In: *International Symposium on Visual Computing*. Springer. 2016, pp. 77–90. DOI: 10.1007/978-3-319-50835-1_8.

[44] **Honauer, K.**, Johannsen, O., Kondermann, D., and Goldlücke, B. "A Dataset and Evaluation Methodology for Depth Estimation on 4D Light Fields". In: *Asian Conference on Computer Vision*. Springer International Publishing. 2016, pp. 19–34. DOI: 10.1007/978-3-319- 54187-7_2.

[56] Johannsen, O., **Honauer, K.**, Goldlücke, B., Alperovich, A., Battisti, F., Bok, Y., Brizzi, M., Carli, M., Choe, G., Diebold, M., et al. "A Taxonomy and Evaluation of Dense Light Field Depth Estimation Algorithms". In: *Conference on Computer Vision and Pattern Recognition Workshops*. IEEE. 2017, pp. 1795–1812. DOI: 10.1109/CVPRW.2017.226.

[154] Zendel, O., **Honauer, K.**, Murschitz, M., Humenberger, M., and Fernández, G. "Analyzing Computer Vision Data - The Good, the Bad and the Ugly". In: *Conference on Computer Vision and Pattern Recognition*. IEEE. 2017, pp. 1980–1990. DOI: 10.1109/CVPR.2017.706.

[157] Zendel, O., **Honauer, K.**, Murschitz, M., Steininger, D., and Fernández, G. "WildDash - Creating Hazard-Aware Benchmarks". In: *European Conference on Computer Vision*. IEEE. 2018.

B

List of Co-Organized Workshops

In order to raise awareness for and foster collaboration on performance evaluation research, I co-organized four workshops and co-hosted two accompanying challenges:

1. *Workshop on Performance Metrics for Correspondence Problems* at CVPR 2015 with Daniel Kondermann, Michael Gösele, Michael Wächter, and Bernd Jähne

2. *Workshop on Datasets and Performance Analysis in Early Vision* at ECCV 2016 with Michael Gösele, Michael Wächter, and Bernd Jähne

3. The *Light Field Depth Estimation Challenge* in conjunction with the *2nd Workshop for Light Fields in Computer Vision* at CVPR 2017 with Bastian Goldlücke, Ole Johannsen, and Jingyi Yu

4. The *Robust Vision Challenge* in conjunction with the *Robust Vision Workshop* at CVPR 2018 with Andreas Geiger, Matthias Nießner, Marc Pollefeys, Carsten Rother, Daniel Scharstein, Hassan Abu Alhaija, Angela Dai, Joel Janai, Torsten Sattler, Nick Schneider, Johannes Schönberger, Thomas Schöps, Jonas Uhrig, Jonas Wulff, and Oliver Zendel

C

List of Co-Published Benchmarks

In order to bring our research results on datasets and performance metrics into academic practice, I co-published three online benchmarking websites:

1. The *4D Light Field Benchmark*[1] with Ole Johannsen, Daniel Kondermann, and Bastian Goldlücke
2. The *HCI Benchmark Suite*[2] with Daniel Kondermann, Rahul Nair, Oliver Zendel, Karsten Krispin, Alexander Brock, and Bernd Jähne
3. The *WildDash Benchmark*[3] with Oliver Zendel, Markus Murschitz, Gustavo Fernández, and Daniel Steininger

Screenshots of the *4D Light Field Benchmark* are depicted on the following pages.

[1] `http://www.lightfield-analysis.net` [2018-09-30]
[2] `http://www.hci-benchmark.org` [2018-09-30]
[3] `http://www.wilddash.cc` [2018-09-30]

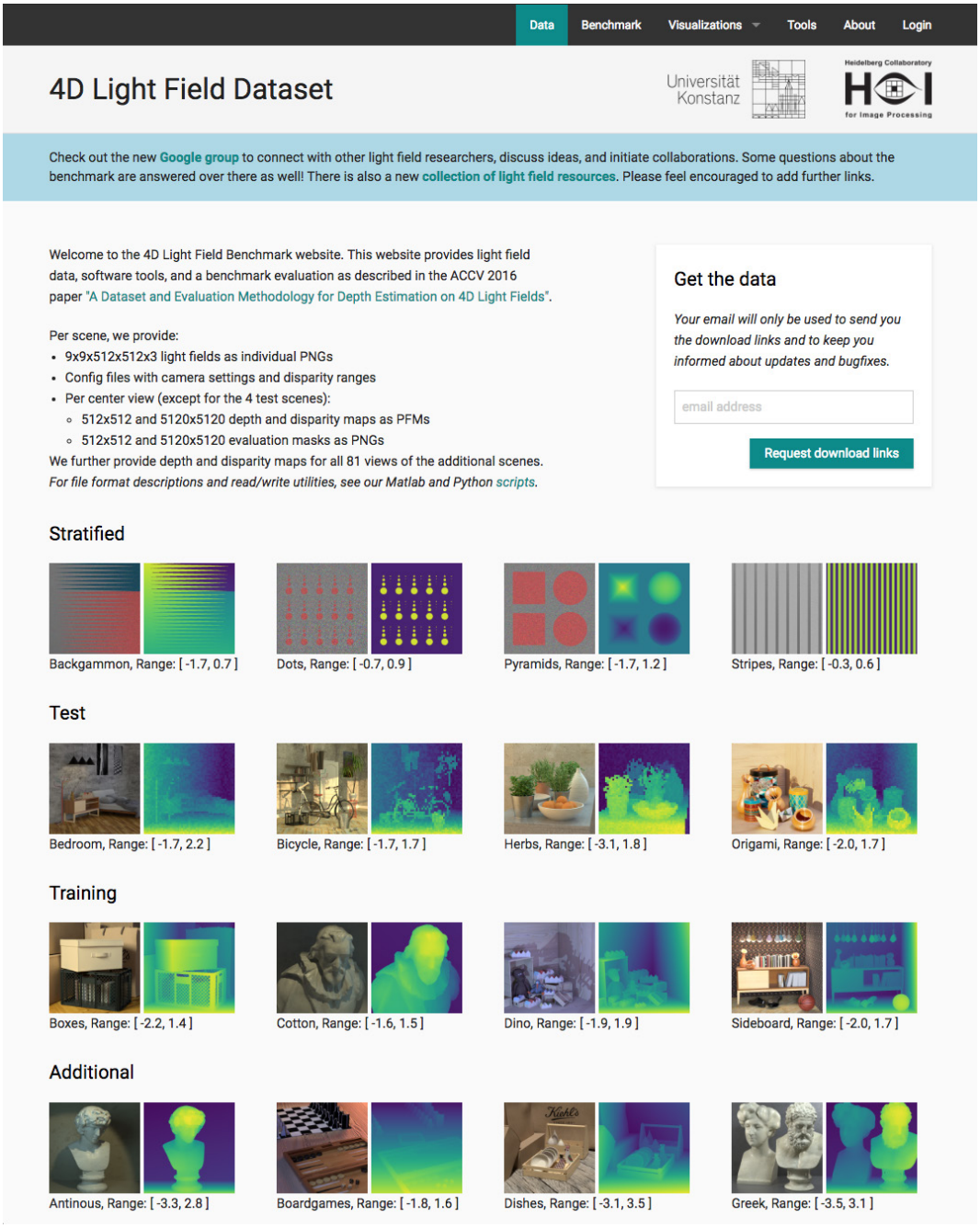


Figure C.1.: **Dataset.** The data page provides general dataset information and visualizations of the scenes. So far, more than 700 unique users submitted their email address to receive download links of our dataset packages.

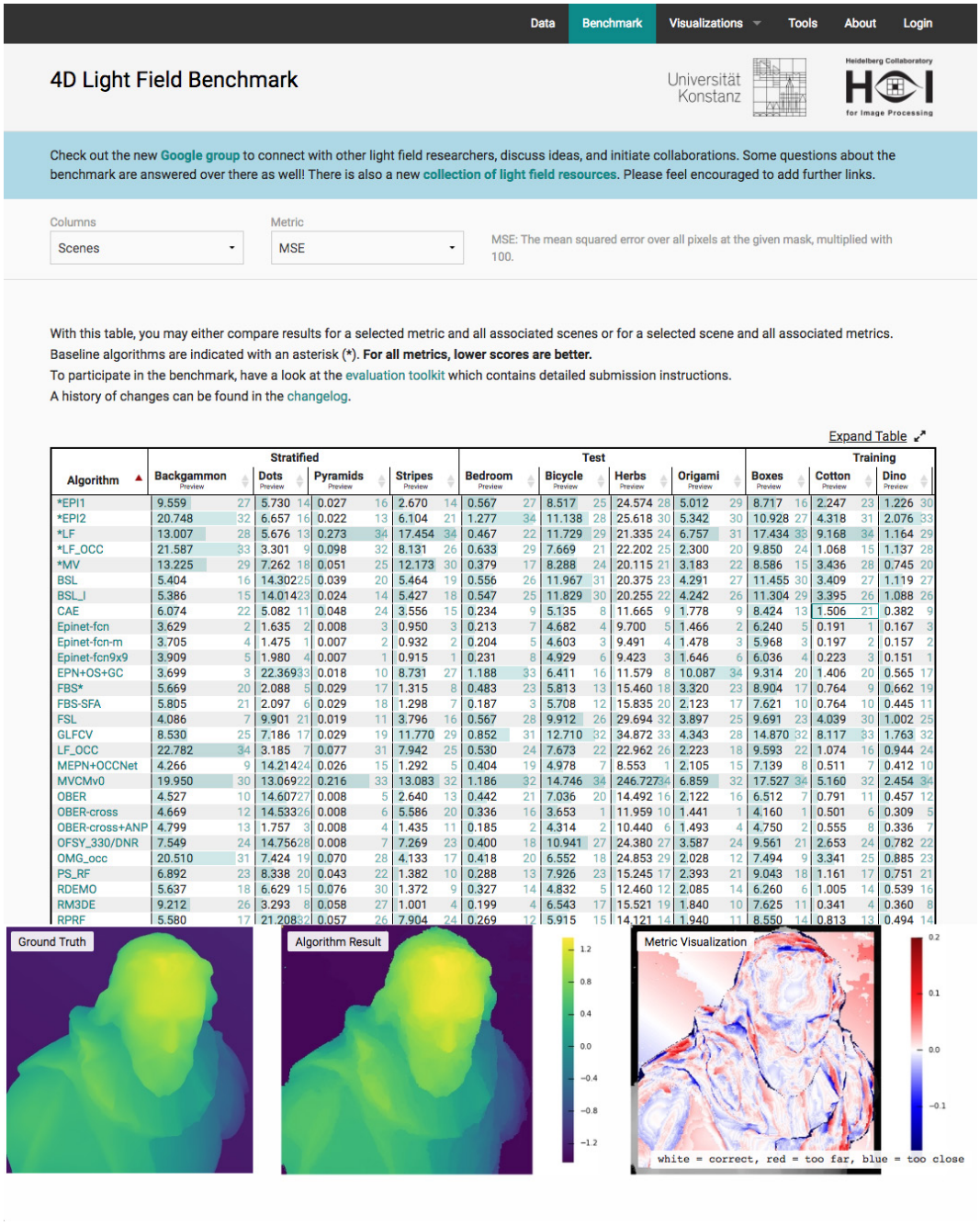


Figure C.2.: **Leaderboard.** Users might sort and adjust the table according to the different metrics and scenes of our benchmark. Disparity maps and error visualizations are depicted for each score to provide additional qualitative information.

Figure C.3.: **Radar Chart.** The radar chart visualization provides an overview of different algorithm characteristics. Users configure the chart by selecting scenes, metrics, and algorithms according to their priorities. We support exports to various file formats for further usage in reports and algorithm publications.

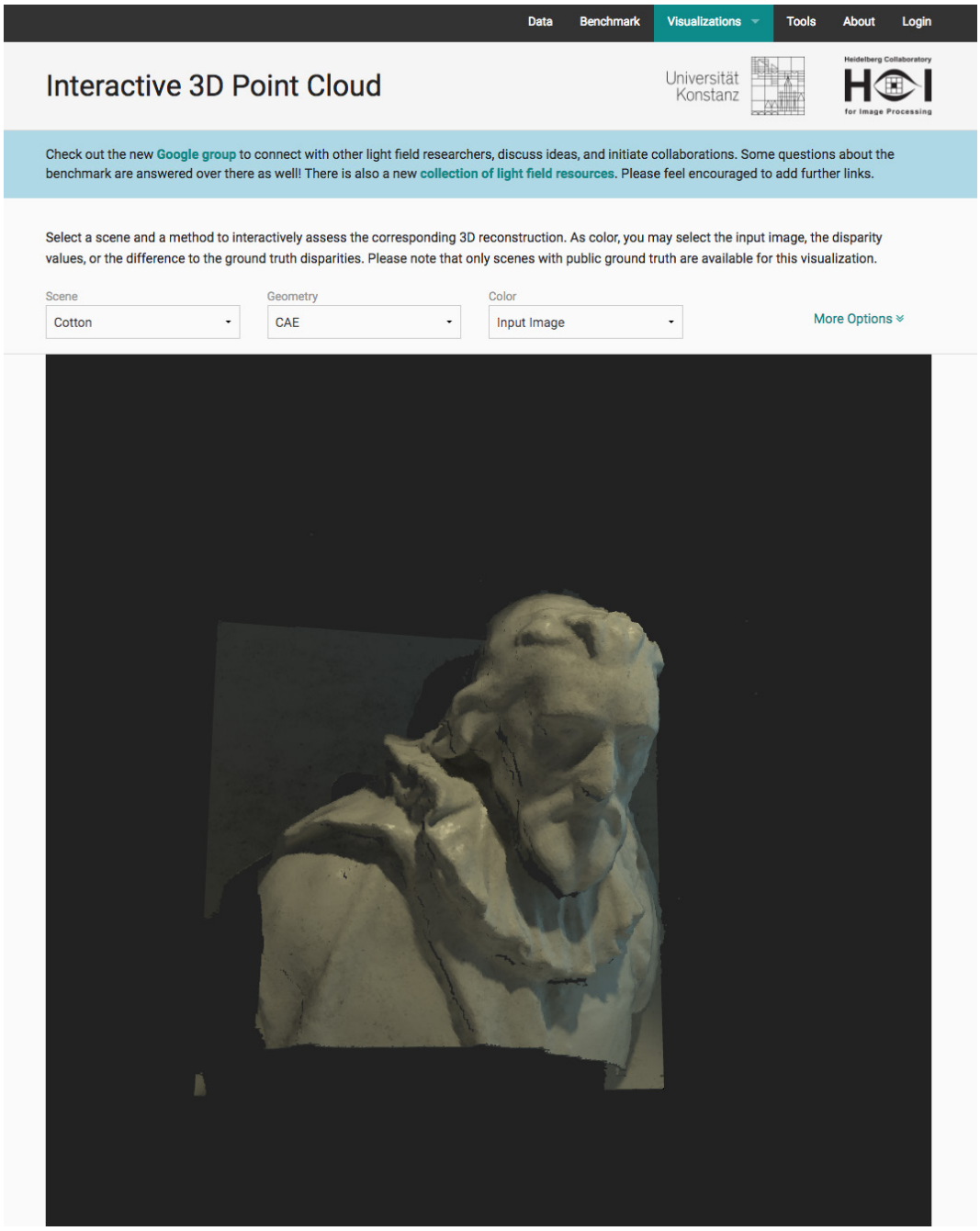


Figure C.4.: **3D Point Cloud.** For the training scenes, we provide interactive 3D point cloud visualizations of the ground truth geometry and the algorithm submissions. As colorization, users might select the original colors of the scene or the point-wise disparity error with respect to the ground truth.

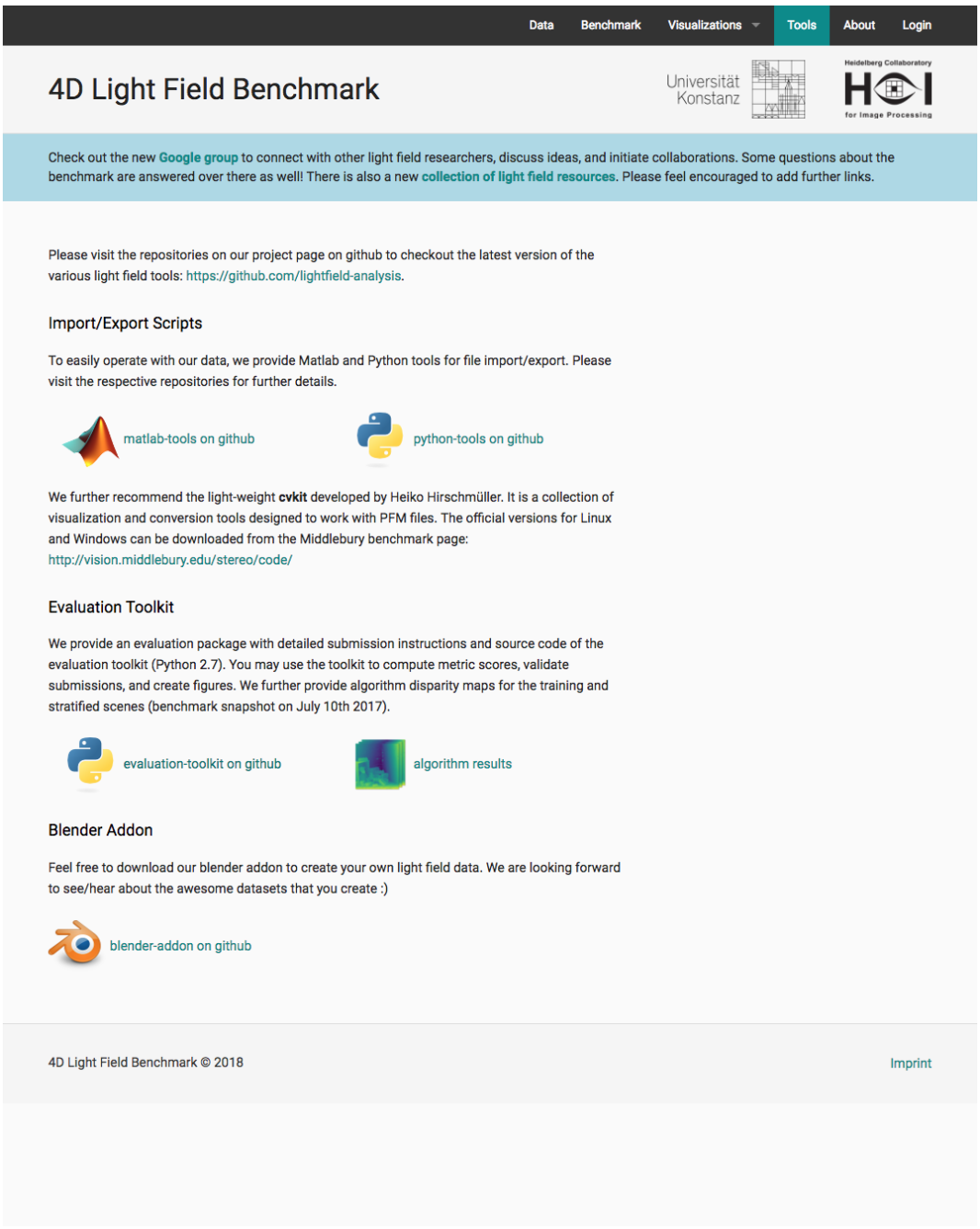


Figure C.5.: **Public Tools.** We publish our tools on Github to foster transparency and reproducibility of academic research. We provide Matlab and Python tools to easily operate with light field files, a Blender addon to create additional light field datasets, and an evaluation toolkit to reproduce our scores and figures.

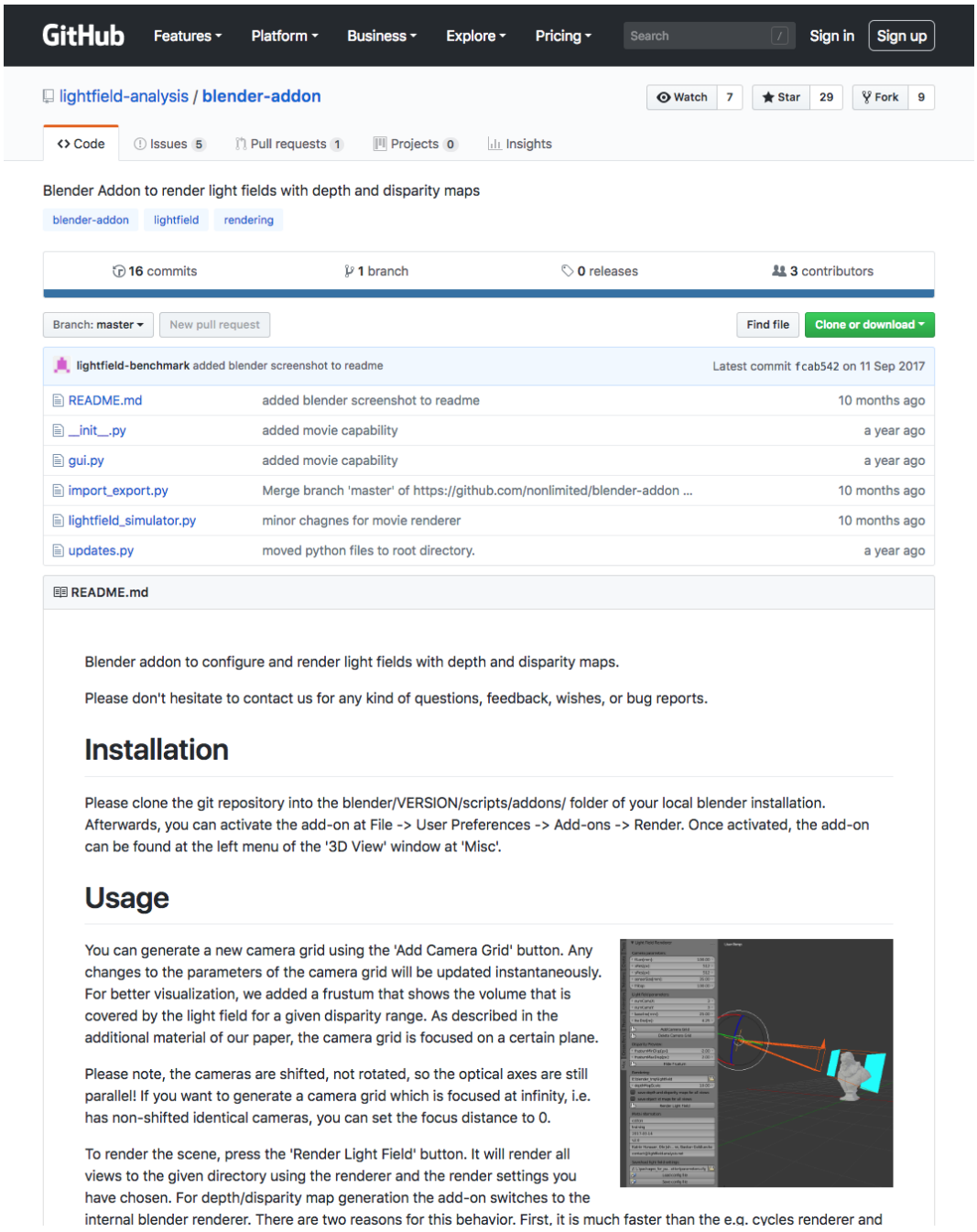


Figure C.6.: **Blender Addon.** With our Blender addon, synthetic light field images can be rendered with corresponding depth and disparity maps. Researchers might use this addon to create additional datasets. They might configure camera and light field parameters such as the baseline and the number of cameras.

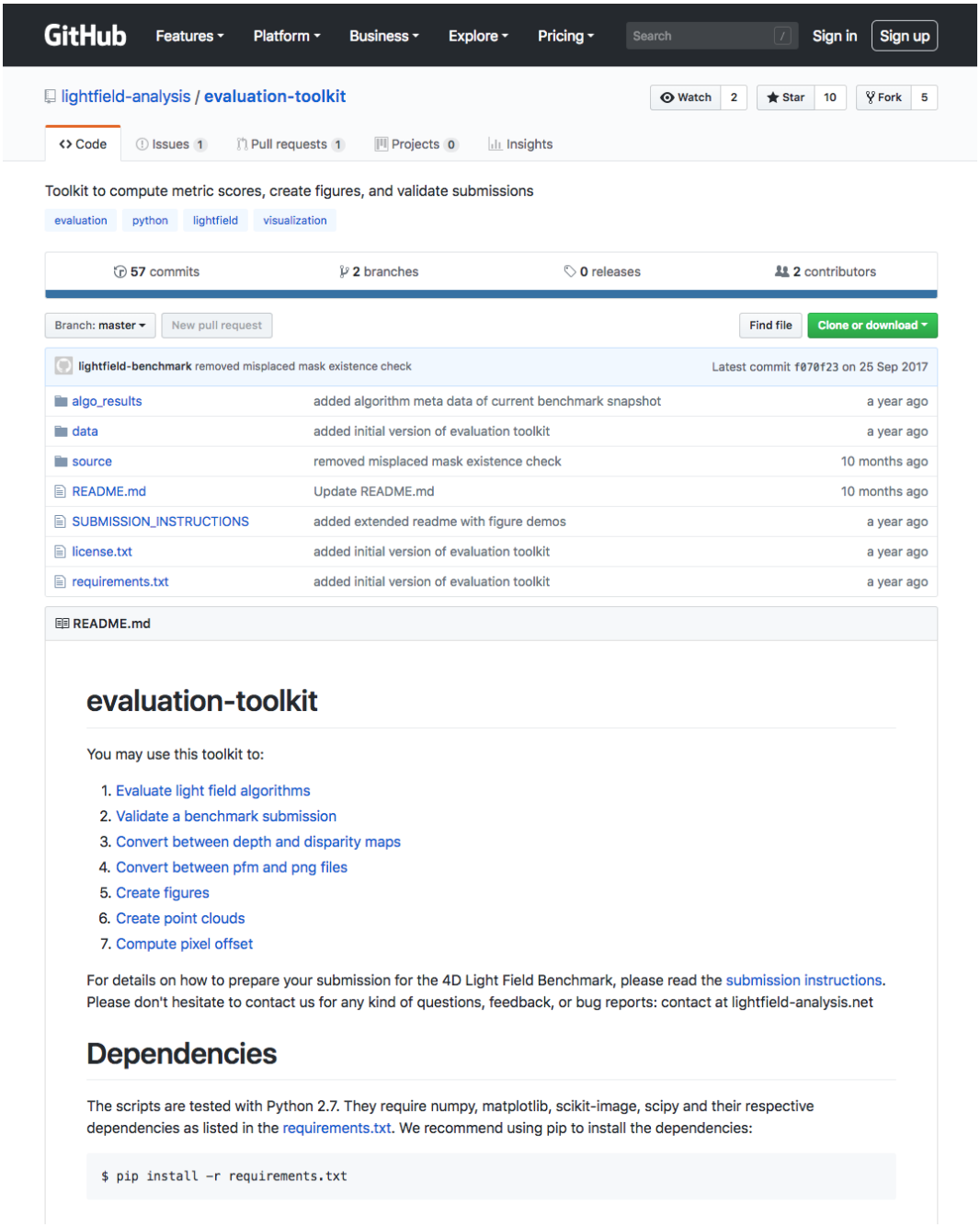


Figure C.7.: **Evaluation Toolkit.** The evaluation toolkit can be used to validate and evaluate submissions for the *4D Light Field Benchmark*. Furthermore, it can be used to reproduce various figures of our publications and to generate point cloud files.

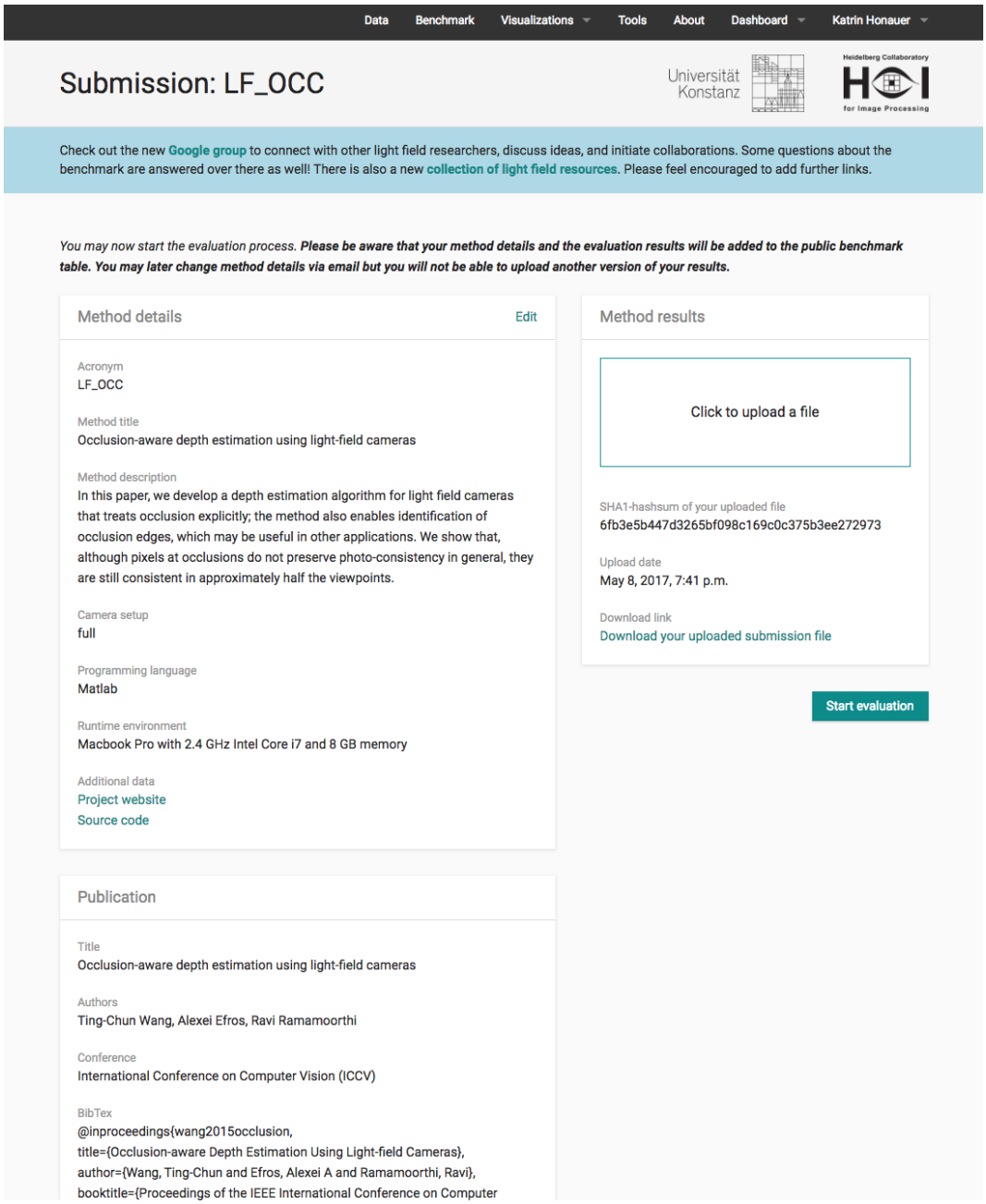


Figure C.8.: **Submission.** To participate in the benchmark, registered users enter their algorithm details and upload their algorithm results. Our server executes the publicly available code of the evaluation toolkit to validate and evaluate the submission. Valid submissions are added to the public leaderboard.

List of Tables

List of Figures

Bibliography

[1] Adams, A. *The (New) Stanford Light Field Archive*. http://lightfield.stanford.edu. [2018-09-30] (cit. on p. 102).

[2] Badino, H., Franke, U., and Pfeiffer, D. "The Stixel World - A Compact Medium Level Representation of the 3D-World". In: *Joint Pattern Recognition Symposium*. Vol. 5748. Springer. 2009, pp. 51–60. DOI: 10.1007/978-3-642-03798-6_6 (cit. on pp. 61, 64).

[3] Baker, H. *Depth from Edge and Intensity Based Stereo*. Tech. rep. Stanford University, Department of Computer Science, 1982 (cit. on p. 16).

[4] Barnard, S. and Fischler, M. "Computational Stereo". In: *ACM Computing Surveys* 14.4 (1982), pp. 553–572. DOI: 10.1145/356893.356896 (cit. on pp. 8, 35, 37).

[5] Blender Foundation. *Blender - A Free and Open-Source 3D Creation Suite*. http://www.blender.org. [2018-09-30]. Blender Institute, Amsterdam, 2016 (cit. on pp. 100, 103).

[6] Bobick, A. and Intille, S. "Large Occlusion Stereo". In: *International Journal of Computer Vision* 33.3 (1999), pp. 181–200. DOI: 10.1023/A:1008150329890 (cit. on p. 28).

[7] Bolles, R., Baker, H., and Marimont, D. "Epipolar-Plane Image Analysis: An Approach to Determining Structure from Motion". In: *International Journal of Computer Vision* 1.1 (1987), pp. 7–55. DOI: 10.1007/BF00128525 (cit. on p. 19).

[8] Bolles, R., Baker, H., and Hannah, M. "The JISCT Stereo Evaluation". In: *DARPA Image Understanding Workshop*. 1993, pp. 263–274 (cit. on pp. 24, 27, 28).

[9] Butler, D., Wulff, J., Stanley, G., and Black, M. "A Naturalistic Open Source Movie for Optical Flow Evaluation". In: *European Conference on Computer Vision*. Vol. 7577. Springer Berlin Heidelberg, 2012, pp. 611–625. DOI: 10.1007/978-3-642-33783-3_44 (cit. on pp. 35, 37, 38, 101).

[10] Butler, D., Wulff, J., Stanley, G., and Black, M. *The MPI Sintel Flow Dataset*. http://sintel.is.tue.mpg.de. [2018-09-30] (cit. on p. 1).

[11] Cabezas, I. and Trujillo, M. "A Non-Linear Quantitative Evaluation Approach for Disparity Estimation Pareto Dominance Applied in Stereo Vision". In: *International Joint Conference on Computer Vision and Computer Graphics Theory and Applications*. Vol. 1. INSTICC. SciTePress, 2011, pp. 704–709. DOI: 10.5220/0003374607040709 (cit. on p. 10).

[12] Cabezas, I., Padilla, V., and Trujillo, M. "A Measure for Accuracy Disparity Maps Evaluation". In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Vol. 7042. Springer Berlin Heidelberg, 2011, pp. 223–231. DOI: 10.1007/978-3-642-25085-9_26 (cit. on pp. 24, 26, 27, 42).

[13] Cabezas, I., Trujillo, M., and Florian, M. "An Evaluation Methodology for Stereo Correspondence Algorithms". In: *International Conference on Computer Vision Theory and Applications*. Vol. 2. INSTICC. SciTePress, 2012, pp. 154–163. DOI: 10.5220/0003850801540163 (cit. on pp. 6, 10, 21).

[14] Cabezas, I., Padilla, V., and Trujillo, M. "BMPRE: An Error Measure for Evaluating Disparity Maps". In: *International Conference on Signal Processing*. Vol. 2. IEEE. 2012, pp. 1051–1055. DOI: 10.1109/ICoSP.2012.6491759 (cit. on pp. 24, 27).

[15] Cabezas, I., Padilla, V., Trujillo, M., and Florian, M. "On the Impact of the Error Measure Selection in Evaluating Disparity Maps". In: *World Automation Congress*. IEEE. 2012, pp. 1–6. ISBN: 978-1-4673-4497-5 (cit. on pp. 9, 24, 26).

[16] Cehovin, L., Lukezic, A., Leonardis, A., and Kristan, M. "Beyond Standard Benchmarks: Parameterizing Performance Evaluation in Visual Object Tracking". In: *International Conference on Computer Vision*. IEEE, 2017, pp. 3343–3351. DOI: 10.1109/ICCV.2017.360 (cit. on p. 9).

[17] Cehovin, L., Kristan, M., and Leonardis, A. "Is My New Tracker Really Better than Yours?" In: *Winter Conference on Applications of Computer Vision*. IEEE. 2014, pp. 540–547. DOI: 10.1109/WACV.2014.6836055 (cit. on p. 1).

[18] Christensen, H. and Förstner, W. "Editorial: Performance Characteristics of Vision Algorithms". In: *Machine Vision and Applications* 9.5-6 (1997), pp. 215–218. DOI: 10.1007/s001380050042 (cit. on p. 2).

[19] Clark, A. and Clark, C. *Performance Characterization in Computer Vision - A Tutorial*. Tech. rep. VASE Laboratory, University of Essex, 1999 (cit. on pp. 2, 21, 23).

[20] Courtney, P. and Thacker, N. "Performance Characterization in Computer Vision: The Role of Statistics in Testing and Design." In: *Imaging and Vision Systems: Theory, Assessment and Applications*. 2001. Nova Science Publishers, Inc. 2001, pp. 109–128. ISBN: 1-59033-033-1 (cit. on p. 9).

[21] Courtney, P., Thacker, N., and Clark, A. "Algorithmic Modelling for Performance Evaluation". In: *Machine Vision and Applications* 9.5-6 (1997), pp. 219–228. DOI: 10.1007/s001380050043 (cit. on pp. 21–23).

[22] Donath, A. and Kondermann, D. "Is Crowdsourcing for Optical Flow Ground Truth Generation Feasible?" In: *International Conference on Computer Vision Systems*. Springer Berlin Heidelberg. 2013, pp. 193–202. DOI: 10.1007/978-3-642-39402-7_20 (cit. on p. 8).

[23] Dubbelman, G., van der Mark, W., van den Heuvel, J., and Groen, F. "Obstacle Detection during Day and Night Conditions using Stereo Vision". In: *Intelligent Robots and Systems*. IEEE. 2007, pp. 109–116. DOI: 10.1109/IROS.2007.4399055 (cit. on p. 18).

[24] Führ, G., Fickel, G., Dal'Aqua, L., Jung, C., Malzbender, T., and Samadani, R. "An Evaluation of Stereo Matching Methods for View Interpolation". In: *International Conference on Image Processing*. IEEE. 2013, pp. 403–407. DOI: 10.1109/ICIP.2013.6738083 (cit. on pp. 8, 17, 24, 32, 33, 45).

[25] Förstner, W. *10 Pros and Cons Against Performance Characterization of Vision Algorithms*. Tech. rep. Institut fur Photogrammetrie, Universität Bonn, 1996 (cit. on pp. 2, 22).

[26] Förstner, W. “Diagnostics and Performance Evaluation in Computer Vision”. In: *Performance versus Methodology in Computer Vision*. NSF/ARPA Workshop. 1994, pp. 11–25 (cit. on pp. 6, 21, 23).

[27] Gaidon, A., Wang, Q., Cabon, Y., and Vig, E. “Virtual Worlds as Proxy for Multi-Object Tracking Analysis”. In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2016, pp. 4340–4349. DOI: 10.1109/CVPR.2016.470 (cit. on pp. 35, 37).

[28] Geiger, A., Lenz, P., and Urtasun, R. “Are we Ready for Autonomous Driving? The KITTI Vision Benchmark Suite”. In: *Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 3354–3361. DOI: 10.1109/CVPR.2012.6248074 (cit. on pp. 1, 7, 25, 31, 35, 38, 101).

[29] Geiger, A., Roser, M., and Urtasun, R. “Efficient Large-Scale Stereo Matching”. In: *Asian Conference on Computer Vision*. Springer Berlin Heidelberg, 2011, pp. 25–38. DOI: 10.1007/978-3-642-19315-6 (cit. on pp. 62, 79).

[30] Güney, F. and Geiger, A. “Displets: Resolving Stereo Ambiguities using Object Knowledge”. In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2015, pp. 4165–4175. DOI: 10.1109/CVPR.2015.7299044 (cit. on pp. 17, 24, 31).

[31] Güssefeld, B., Kondermann, D., Schwartz, C., and Klein, R. “Are Reflectance Field Renderings Appropriate for Optical Flow Evaluation?” In: *International Conference on Image Processing*. IEEE. 2014, pp. 1982–1986. DOI: 10.1109/ICIP.2014.7025397 (cit. on pp. 2, 8, 35, 37).

[32] Güssefeld, B., Honauer, K., and Kondermann, D. “Creating Feasible Reflectance Data for Synthetic Optical Flow Datasets”. In: *International Symposium on Visual Computing*. Springer. 2016, pp. 77–90. DOI: 10.1007/978-3-319-50835-1_8 (cit. on pp. 35, 37, 123).

[33] Hamilton, O., Breckon, T., Bai, X., and Kamata, S. “A Foreground Object Based Quantitative Assessment of Dense Stereo Approaches for Use in Automotive Environments”. In: *International Conference on Image Processing*. IEEE. 2013, pp. 418–422. DOI: 10.1109/ICIP.2013.6738086 (cit. on p. 18).

[34] Haralick, R. “Computer Vision Theory: The Lack Thereof”. In: *Computer Vision, Graphics, and Image Processing* 36.2-3 (1986), pp. 372–386. DOI: 10.1016/0734-189X(86)90082-4 (cit. on p. 2).

[35] Haralick, R. “Performance Assessment of Near-Perfect Machines”. In: *Machine Vision and Applications* 2.1 (1989), pp. 1–16. DOI: 10.1007/BF01214393 (cit. on pp. 21, 23).

[36] Haralick, R. “Performance Characterization in Computer Vision”. In: *Computer Vision and Image Understanding*. Vol. 60. Springer. 1993, pp. 1–9. DOI: 10.1006/cviu.1994.1055 (cit. on pp. 2, 7, 21–23, 35, 36).

[37] Hartley, R. and Zisserman, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. DOI: 10.1017/CBO9780511811685 (cit. on pp. 11, 12).

[38] *HCI Robust Vision Challenge*. ECCV 2012. https://hci.iwr.uni-heidelberg.de/Robust_Vision_Challenge_2012. [2018-09-30] (cit. on pp. 24, 34).

[39] Heber, S. and Pock, T. “Shape from Light Field meets Robust PCA”. In: *European Conference on Computer Vision*. Springer International Publishing, 2014, pp. 751–767. DOI: 10.1007/978-3-319-10599-4_48 (cit. on pp. 19, 101, 102).

[40] Hirschmüller, H. “Stereo Processing by Semiglobal Matching and Mutual Information”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.2 (2008), pp. 328–341. DOI: 10.1109/TPAMI.2007.1166 (cit. on p. 79).

[41] Hirschmüller, H. and Scharstein, D. “Evaluation of Cost Functions for Stereo Matching”. In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8. DOI: 10.1109/CVPR.2007.383248 (cit. on pp. 9, 13, 23, 37).

[42] Hirschmüller, H. and Scharstein, D. “Evaluation of Stereo Matching Costs on Images with Radiometric Differences”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.9 (2009), pp. 1582–1599. DOI: 10.1109/TPAMI.2008.221 (cit. on pp. 9, 13, 14, 23, 35, 36, 56).

[43] Hirschmüller, H., Innocent, P., and Garibaldi, J. “Real-Time Correlation-Based Stereo Vision with Reduced Border Errors”. In: *International Journal of Computer Vision* 47.1-3 (2002), pp. 229–246. DOI: 10.1023/A:1014554110407 (cit. on p. 50).

[44] Honauer, K., Johannsen, O., Kondermann, D., and Goldlücke, B. “A Dataset and Evaluation Methodology for Depth Estimation on 4D Light Fields”. In: *Asian Conference on Computer Vision*. Springer International Publishing, 2016, pp. 19–34. DOI: 10.1007/978-3-319-54187-7_2 (cit. on pp. 1, 3, 8, 26, 35, 41, 44–46, 51, 52, 55, 65–69, 99, 102, 104, 123).

[45] Honauer, K., Johannsen, O., Kondermann, D., and Goldlücke, B. *The 4D Light Field Benchmark*. http://lightfield-analysis.net. [2018-09-30] (cit. on pp. 10, 35, 37).

[46] Honauer, K., Maier-Hein, L., and Kondermann, D. “The HCI Stereo Metrics: Geometry-Aware Performance Analysis of Stereo Algorithms”. In: *International Conference on Computer Vision*. IEEE, 2015, pp. 2120–2128. DOI: 10.1109/ICCV.2015.245 (cit. on pp. 3, 9, 17, 35, 41–44, 46, 51, 52, 55, 123).

[47] Hong, L. and Chen, G. “Segment-Based Stereo Matching Using Graph Cuts”. In: *Conference on Computer Vision and Pattern Recognition*. IEEE. 2004, pp. 74–81. DOI: 10.1109/CVPR.2004.1315016 (cit. on p. 46).

[48] Hosni, A., Rhemann, C., Bleyer, M., Rother, C., and Gelautz, M. “Fast Cost-Volume Filtering for Visual Correspondence and Beyond”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.2 (2013), pp. 504–511. DOI: 10.1109/TPAMI.2012.156 (cit. on p. 79).

[49] Hsieh, Y., McKeown, D., and Perlant, F. “Performance Evaluation of Scene Registration and Stereo Matching for Cartographic Feature Extraction”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (1992), pp. 214–238. DOI: 10.1109/34.121790 (cit. on p. 17).

[50] Häusler, R. and Klette, R. “Benchmarking Stereo Data (not the Matching Algorithms)”. In: *Pattern Recognition*. Vol. 6376. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, pp. 383–392. DOI: 10.1007/978-3-642-15986-2_39 (cit. on p. 35).

[51] Häusler, R. and Kondermann, D. “Synthesizing Real World Stereo Challenges”. In: *German Conference on Pattern Recognition*. Springer. 2013, pp. 164–173. DOI: 10.1007/978-3-642-40602-7_17 (cit. on pp. 23, 35, 36, 38, 65, 66).

[52] Häusler, R., Morales, S., Hermann, S., and Klette, R. “Towards Benchmarking of Real-World Stereo Data”. In: *International Conference of Image and Vision Computing New Zealand*. IEEE, 2010. DOI: 10.1109/IVCNZ.2010.6148827 (cit. on pp. 35, 37).

[53] Jähne, B. *Digitale Bildverarbeitung und Bildgewinnung*. Springer-Verlag Berlin Heidelberg, 2012. ISBN: 978-3-642-04951-4. DOI: 10.1007/978-3-642-04952-1 (cit. on p. 11).

[54] Jeon, H.-G., Park, J., Choe, G., Park, J., Bok, Y., Tai, Y.-W., and Kweon, I.-S. “Accurate Depth Map Estimation from a Lenslet Light Field Camera”. In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2015, pp. 1547–1555. DOI: 10.1109/CVPR.2015.7298762 (cit. on pp. 19, 46, 50, 101, 102, 107, 108).

[55] Jeon, H.-G., Park, J., Choe, G., Park, J., Bok, Y., Tai, Y.-W., and Kweon, I.-S. “Depth from a Light Field Image with Learning-Based Matching Costs”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018). DOI: 10.1109/TPAMI.2018.2794979 (cit. on p. 19).

[56] Johannsen, O., Honauer, K., Goldlücke, B., Alperovich, A., Battisti, F., Bok, Y., Brizzi, M., Carli, M., Choe, G., Diebold, M., Gutsche, M., Jeon, H., Kweon, I.-S., Park, J., Park, J., Schilling, H., Sheng, H., Si, L., Strecke, M., Sulc, A., Tai, Y., Wang, Q., Wang, T., Wanner, S., Xiong, Z., Yu, J., Zhang, S., and Zhu, H. “A Taxonomy and Evaluation of Dense Light Field Depth Estimation Algorithms”. In: *Conference on Computer Vision and Pattern Recognition Workshops*. IEEE. 2017, pp. 1795–1812. DOI: 10.1109/CVPRW.2017.226 (cit. on pp. 3, 19, 41, 98, 99, 101, 105, 107, 109, 124).

[57] Johannsen, O., Sulc, A., and Goldlücke, B. “What Sparse Light Field Coding Reveals about Scene Structure”. In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2016, pp. 3262–3270. DOI: 10.1109/CVPR.2016.355 (cit. on pp. 19, 101, 102).

[58] Keller, C., Enzweiler, M., and Gavrila, D. “A New Benchmark for Stereo-Based Pedestrian Detection”. In: *Intelligent Vehicles Symposium*. IEEE. 2011, pp. 691–696. DOI: 10.1109/IVS.2011.5940480 (cit. on p. 18).

[59] Kelly, P., O’Connor, N., and Smeaton, A. “A Framework for Evaluating Stereo-Based Pedestrian Detection Techniques”. In: *Transactions on Circuits and Systems for Video Technology* 18.8 (2008), pp. 1163–1167. DOI: 10.1109/TCSVT.2008.928228 (cit. on p. 18).

[60] Kim, C., Zimmer, H., Pritch, Y., Sorkine-Hornung, A., and Gross, M. “Scene Reconstruction from High Spatio-Angular Resolution Light Fields”. In: *ACM Transactions on Graphics* 32.4 (2013), 73:1–73:12. DOI: 10.1145/2461912.2461926 (cit. on p. 102).

[61] Klette, R., Krüger, N., Vaudrey, T., Pauwels, K., van Hulle, M., Morales, S., Kandil, F., Häusler, R., Pugeault, N., Rabe, C., and Lappe, M. "Performance of Correspondence Algorithms in Vision-Based Driver Assistance Using an Online Image Sequence Database". In: *IEEE Transactions on Vehicular Technology* 60.5 (2011), pp. 2012–2026. DOI: 10.1109/TVT.2011.2148134 (cit. on pp. 35, 38).

[62] Klette, R., Krüger, N., Vaudrey, T., Pauwels, K., Van Hulle, M., Morales, S., Kandil, F., Häusler, R., Pugeault, N., Rabe, C., and Lappe, M. *Performance of Correspondence Algorithms in Vision-Based Driver Assistance using EISATS*. Tech. rep. University of Auckland, New Zealand, 2010, pp. 1–37 (cit. on pp. 23, 35, 37).

[63] Kondermann, D. "Ground Truth Design Principles: An Overview". In: *International Workshop on Video and Image Ground Truth in Computer Vision Applications*. ACM. 2013. DOI: 10.1145/2501105.2501114 (cit. on pp. 6–8, 23, 35, 36).

[64] Kondermann, D., Abraham, S., Brostow, G., Förstner, W., Gehrig, S., Imiya, A., Jähne, B., Klose, F., Magnor, M., Mayer, H., Mester, R., Pajdla, T., Reulke, R., and Zimmer, H. "On Performance Analysis of Optical Flow Algorithms". In: *International Conference on Theoretical Foundations of Computer Vision: Outdoor and Large-Scale Real-World Scene Analysis*. Springer Berlin Heidelberg, 2012, pp. 329–355. DOI: 10.1007/978-3-642-34091-8_15 (cit. on p. 6).

[65] Kondermann, D., Nair, R., Meister, S., Mischler, W., Güssefeld, B., Honauer, K., Hofmann, S., Brenner, C., and Jähne, B. "Stereo Ground Truth with Error Bars". In: *Asian Conference on Computer Vision*. Vol. 9007. Springer International Publishing, 2015, pp. 595–610. DOI: 10.1007/978-3-319-16814-2_39 (cit. on pp. 24, 31, 61, 123).

[66] Kondermann, D., Nair, R., Honauer, K., Krispin, K., Andrulis, J., Brock, A., Güssefeld, B., Rahimimoghaddam, M., Hofmann, S., Brenner, C., and Jähne, B. "The HCI Benchmark Suite: Stereo And Flow Ground Truth With Uncertainties for Urban Autonomous Driving". In: *Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2016, pp. 19–28. DOI: 10.1109/CVPRW.2016.10 (cit. on pp. 3, 7, 14, 25, 35, 37, 38, 44, 51, 61, 63, 102, 123).

[67] Kostková, J., Čech, J., and Šára, R. *The CMP Evaluation of Stereo Algorithms*. Tech. rep. Center for Machine Perception, Czech Technical University Prague, 2003 (cit. on pp. 16, 21, 24, 28, 29, 35, 36).

[68] Kostlivá, J., Čech, J., and Šára, R. "Feasibility Boundary in Dense and Semi-Dense Stereo Matching". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8. DOI: 10.1109/CVPR.2007.383350 (cit. on pp. 21, 24, 30, 31, 42).

[69] Kostlivá, J., Čech, J., and Šára, R. *ROC Based Evaluation of Stereo Algorithms*. Tech. rep. Center for Machine Perception, Czech Technical University Prague, 2007 (cit. on p. 24).

[70] Kristan, M., Matas, J., Leonardis, A., Vojir, T., Pflugfelder, R., Fernández, G., Nebehay, G., Porikli, F., and Cehovin, L. "A Novel Performance Evaluation Methodology for Single-Target Trackers". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.11 (2016), pp. 2137–2155. DOI: 10.1109/TPAMI.2016.2516982 (cit. on p. 10).

[71] Leclerc, Y., Luong, Q., and Fua, P. “Measuring the Self-Consistency of Stereo Algorithms”. In: *European Conference on Computer Vision*. Springer Berlin Heidelberg, 2000, pp. 282–298. DOI: 10.1007/3-540-45054-8_19 (cit. on pp. 24, 33).

[72] Leclercq, P. and Morris, J. “Robustness to Noise of Stereo Matching”. In: *International Conference on Image Analysis and Processing*. IEEE, 2003, pp. 606–611. DOI: 10.1109/ICIAP.2003.1234117 (cit. on pp. 21, 23, 24, 26, 35, 38).

[73] Li, L., Yu, X., Zhang, S., Zhao, X., and Zhang, L. “3D Cost Aggregation with Multiple Minimum Spanning Trees for Stereo Matching”. In: *Applied Optics* 56.12 (2017), pp. 3411–3420. DOI: 10.1364/AO.56.003411 (cit. on p. 42).

[74] Lowe, D. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110. DOI: 10.1023/B:VISI.0000029664.99615.94 (cit. on p. 37).

[75] Maier-Hein, L., Groch, A., Bartoli, A., Bodenstedt, S., Boissonnat, G., Chang, P., Clancy, N., Elson, D., Haase, S., Heim, E., Hornegger, J., Jannin, P., Kenngott, H., Kilgus, T., Müller-Stich, B., Oladokun, D., Röhl, S., dos Santos, T., Schlemmer, H., Seitel, A., Speidel, S., Wagner, M., and Stoyanov, D. “Comparative Validation of Single-Shot Optical Techniques for Laparoscopic 3-D Surface Reconstruction”. In: *IEEE Transactions on Medical Imaging* 33.10 (2014), pp. 1913–1930. DOI: 10.1109/TMI.2014.2325607 (cit. on p. 17).

[76] Maier-Hein, L., Mersmann, S., Kondermann, D., Stock, C., Kenngott, H., Sanchez, A., Wagner, M., Preukschas, A., Wekerle, A.-L., Helfert, S., Bodenstedt, S., and Speidel, S. “Crowdsourcing for Reference Correspondence Generation in Endoscopic Images”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2014, pp. 349–356. DOI: 10.1007/978-3-319-10470-6_44 (cit. on p. 8).

[77] Maier-Hein, L., Kondermann, D., Roß, T., Mersmann, S., Heim, E., Bodenstedt, S., Kenngott, H., Sanchez, A., Wagner, M., Preukschas, A., Wekerle, A.-L., Helfert, S., März, K., Mehrabi, A., Speidel, S., and Stock, C. “Crowdtruth Validation: A New Paradigm for Validating Algorithms that Rely on Image Correspondences”. In: *International Journal of Computer Assisted Radiology and Surgery* 10.8 (2015), pp. 1201–1212. DOI: 10.1007/s11548-015-1168-3 (cit. on pp. 17, 33).

[78] Maier-Hein, L., Mountney, P., Bartoli, A., Elhawary, H., Elson, D., Groch, A., Kolb, A., Rodrigues, M., Sorger, J., Speidel, S., and Stoyanov, D. “Optical Techniques for 3D Surface Reconstruction in Computer-Assisted Laparoscopic Surgery”. In: *Medical Image Analysis* 17.8 (2013), pp. 974–996. DOI: 10.1016/j.media.2013.04.003 (cit. on pp. 17, 41).

[79] Maimone, M. and Shafer, S. “A Taxonomy for Stereo Computer Vision Experiments”. In: *European Conference on Computer Vision Workshops*. 1996, pp. 59–79 (cit. on pp. 21, 35, 36).

[80] Malpica, W. and Bovik, A. “Range Image Quality Assessment by Structural Similarity”. In: *International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2009, pp. 1149–1152. DOI: 10.1109/ICASSP.2009.4959792 (cit. on pp. 21, 24, 29, 30).

[81] Marr, D. and Poggio, T. “A Computational Theory of Human Stereo Vision”. In: *Proceedings of the Royal Society of London B: Biological Sciences*. Vol. 204. 1156. The Royal Society, 1979, pp. 301–328. DOI: 10.1098/rspb.1979.0029 (cit. on p. 14).

[82] Mayer, N., Ilg, E., Häusser, P., Fischer, P., Cremers, D., Dosovitskiy, A., and Brox, T. “A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation”. In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2016, pp. 4040–4048. DOI: 10.1109/CVPR.2016.438 (cit. on pp. 8, 35, 37, 38, 81, 83, 84, 118).

[83] Mayhew, J. and Frisby, J. “The Computation of Binocular Edges”. In: *Perception* 9.1 (1980), pp. 69–86. DOI: 10.1068/p090069 (cit. on p. 16).

[84] Mei, X., Sun, X., Dong, W., Wang, H., and Zhang, X. “Segment-Tree Based Cost Aggregation for Stereo Matching”. In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2013, pp. 313–320. DOI: 10.1109/CVPR.2013.47 (cit. on p. 62).

[85] Meister, S. “On Creating Reference Data for Performance Analysis in Image Processing”. PhD thesis. 2014. DOI: 10.11588/heidok.00016193 (cit. on pp. 2, 7, 8).

[86] Meister, S. and Kondermann, D. “Real Versus Realistically Rendered Scenes for Optical Flow Evaluation”. In: *ITG Conference on Electronic Media Technology*. IEEE, 2011, pp. 1–6 (cit. on pp. 8, 35, 37).

[87] Meister, S., Izadi, S., Kohli, P., Hämmerle, M., Rother, C., and Kondermann, D. “When Can We Use KinectFusion for Ground Truth Acquisition”. In: *International Conference on Intelligent Robots and Systems: Workshop on Color-Depth Camera Fusion in Robotics*. IEEE, 2012 (cit. on p. 8).

[88] Menze, M. and Geiger, A. *Kitti Stereo Evaluation - 2012 and 2015*. http://www.cvlibs.net/datasets/kitti. [2018-09-30] (cit. on pp. 1, 2, 5, 10, 12, 21, 24, 25, 31, 35, 37, 41).

[89] Menze, M. and Geiger, A. “Object Scene Flow for Autonomous Vehicles”. In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2015, pp. 3061–3070. DOI: 10.1109/CVPR.2015.7298925 (cit. on pp. 24, 31).

[90] Milani, S., Ferrario, D., and Tubaro, S. “No-Reference Quality Metric for Depth Maps”. In: *International Conference on Image Processing*. IEEE, 2013, pp. 408–412. DOI: 10.1109/ICIP.2013.6738084 (cit. on pp. 17, 24, 34, 35, 38).

[91] Morales, S. and Klette, R. “A Third Eye for Performance Evaluation in Stereo Sequence Analysis”. In: *International Conference on Computer Analysis of Images and Patterns*. Springer. 2009, pp. 1078–1086. DOI: 10.1007/978-3-642-03767-2_131 (cit. on pp. 8, 24, 32, 33).

[92] Morales, S. and Klette, R. “Ground Truth Evaluation of Stereo Algorithms for Real World Applications”. In: *Asian Conference on Computer Vision Workshops*. Springer. 2011, pp. 152–162. DOI: 10.1007/978-3-642-22819-3_16 (cit. on pp. 24, 31).

[93] Morales, S., Vaudrey, T., and Klette, R. "Robustness Evaluation of Stereo Algorithms on Long Stereo Sequences". In: *Intelligent Vehicles Symposium*. IEEE. 2009, pp. 347–352. DOI: 10.1109/IVS.2009.5164302 (cit. on pp. 23, 35, 38).

[94] Mousnier, A., Vural, E., and Guillemot, C. *Lytro Dataset*. https://www.irisa.fr/temics/demos/lightField/index.html. [2018-09-30] (cit. on p. 102).

[95] Nair, R., Meister, S., Lambers, M., Balda, M., Hofmann, H., Kolb, A., Kondermann, D., and Jähne, B. "Ground Truth for Evaluating Time of Flight Imaging". In: *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*. Vol. 8200. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 52–74. DOI: 10.1007/978-3-642-44964-2_4 (cit. on p. 8).

[96] Nair, R., Fitzgibbon, A., Kondermann, D., and Rother, C. "Reflection Modeling for Passive Stereo". In: *International Conference on Computer Vision*. IEEE, 2015, pp. 2291–2299. DOI: 10.1109/ICCV.2015.264 (cit. on p. 79).

[97] Nakamura, Y., Matsuura, T., Satoh, K., and Ohta, Y. "Occlusion Detectable Stereo - Occlusion Patterns in Camera Matrix". In: *Conference on Computer Vision and Pattern Recognition*. IEEE. 1996, pp. 371–378. DOI: 10.1109/CVPR.1996.517099 (cit. on p. 32).

[98] Neilson, D. and Yang, Y.-H. "Evaluation of Constructable Match Cost Measures for Stereo Correspondence Using Cluster Ranking". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8. DOI: 10.1109/CVPR.2008.4587692 (cit. on pp. 9, 10, 35).

[99] Neri, A., Carli, M., and Battisti, F. "A Multi-Resolution Approach to Depth Field Estimation in Dense Image Arrays". In: *International Conference on Image Processing*. IEEE. 2015, pp. 3358–3362. DOI: 10.1109/ICIP.2015.7351426 (cit. on p. 19).

[100] Nielsen, M., Andersen, H., Slaughter, D., and Granum, E. "Ground Truth Evaluation of Computer Vision Based 3D Reconstruction of Synthesized and Real Plant Images". In: *Precision Agriculture* 8.1-2 (2007), pp. 49–62. DOI: 10.1007/s11119-006-9028-3 (cit. on p. 17).

[101] Okutomi, M., Katayama, Y., and Oka, S. "A Simple Stereo Algorithm to Recover Precise Object Boundaries and Smooth Surfaces". In: *International Journal of Computer Vision* 47.1-3 (2002), pp. 261–273. DOI: 10.1023/A:1014510328154 (cit. on p. 18).

[102] Park, H. and Lee, K. "Look Wider to Match Image Patches with Convolutional Neural Networks". In: *IEEE Signal Processing Letters* 24.12 (2017), pp. 1788–1792. DOI: 10.1109/LSP.2016.2637355 (cit. on p. 42).

[103] Pfeiffer, D., Gehrig, S, and Schneider, N. "Exploiting the Power of Stereo Confidences". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2013, pp. 297–304. DOI: 10.1109/CVPR.2013.45 (cit. on p. 41).

[104] Rerabek, M. and Ebrahimi, T. "New Light Field Image Dataset". In: *8th International Conference on Quality of Multimedia Experience*. 2016 (cit. on p. 102).

[105] Ros, G., Sellart, L., Materzynska, J., Vazquez, D., and Lopez, A. M. "The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes". In: *Conference on Computer Vision and Pattern Recognition*. IEEE. 2016. DOI: 10.1109/CVPR.2016.352 (cit. on pp. 35, 37).

[106] Rosenberger, C. and Chehdi, K. "Genetic Fusion: Application to Multi-Components Image Segmentation". In: *International Conference on Acoustics, Speech, and Signal Processing*. IEEE. 2000, pp. 2223–2226. DOI: 10.1109/ICASSP.2000.859280 (cit. on p. 34).

[107] Rubner, Y., Tomasi, C., and Guibas, L. J. "A Metric for Distributions with Applications to Image Databases". In: *International Conference on Computer Vision*. IEEE, 1998, pp. 59–66. DOI: 10.1109/ICCV.1998.710701 (cit. on p. 31).

[108] Särndal, C.-E., Swensson, B., and Wretman, J. *Model Assisted Survey Sampling*. Springer Series in Statistics. Springer Science & Business Media, 2003. ISBN: 978-0-387-40620-6 (cit. on p. 65).

[109] Scharstein, D. and Szeliski, R. "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms". In: *International Journal of Computer Vision* 47.1 (2002), pp. 7–42. DOI: 10.1023/A:1014573219977 (cit. on pp. 8, 9, 12, 13, 21, 23–25, 29, 48).

[110] Scharstein, D. and Szeliski, R. "High-Accuracy Stereo Depth Maps Using Structured Light". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2003, pp. 195–202. DOI: 10.1109/CVPR.2003.1211354 (cit. on pp. 37, 82).

[111] Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nešić, N., Wang, X., and Westling, P. "High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth". In: *German Conference on Pattern Recognition*. Springer International Publishing, 2014, pp. 31–42. DOI: 10.1007/978-3-319-11752-2_3 (cit. on pp. 7, 26, 31, 35, 37, 38, 43, 50, 54, 56, 62, 81, 84–86, 101, 102).

[112] Scharstein, D., Szeliski, R., and Hirschmüller, H. *Middlebury Stereo Evaluation - Version 3*. http://vision.middlebury.edu/stereo. [2018-09-30] (cit. on pp. 2, 5, 10, 12, 21, 24, 25, 30, 31, 35, 37, 41).

[113] Schilling, H., Diebold, M., Rother, C., and Jähne, B. "Trust your Model: Light Field Depth Estimation with Inline Occlusion Handling". In: *Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4530–4538 (cit. on pp. 19, 107, 108).

[114] Sellent, A. and Wingbermühle, J. "Quality Assessment of Non-Dense Image Correspondences". In: *European Conference on Computer Vision Workshops*. Springer Berlin Heidelberg, 2012. DOI: 10.1007/978-3-642-33868-7_12 (cit. on pp. 21, 24, 31).

[115] Shen, Y., Chaohui, L, Xu, P., and Xu, L. "Objective Quality Assessment of Noised Stereoscopic Images". In: *International Conference on Measuring Technology and Mechatronics Automation*. IEEE, 2011, pp. 745–747. DOI: 10.1109/ICMTMA.2011.470 (cit. on pp. 21, 24, 34).

[116] Si, L. and Wang, Q. “Dense Depth-Map Estimation and Geometry Inference from Light Fields via Global Optimization”. In: *Asian Conference on Computer Vision*. Springer. 2016, pp. 83–98. DOI: 10.1007/978-3-319-54187-7_6 (cit. on pp. 46, 107, 108).

[117] Sinha, S., Steedly, D., and Szeliski, R. “Piecewise Planar Stereo for Image-Based Rendering”. In: *International Conference on Computer Vision*. IEEE, 2009, pp. 1881–1888. DOI: 10.1109/ICCV.2009.5459417 (cit. on p. 46).

[118] Sinha, S., Scharstein, D., and Szeliski, R. “Efficient High-Resolution Stereo Matching Using Local Plane Sweeps”. In: *Conference on Computer Vision and Pattern Recognition*. IEEE. 2014, pp. 1582–1589. DOI: 10.1109/CVPR.2014.205 (cit. on p. 46).

[119] Strecke, M., Alperovich, A., and Goldlücke, B. “Accurate Depth and Normal Maps from Occlusion-Aware Focal Stack Symmetry”. In: *Conference on Computer Vision and Pattern Recognition*. IEEE. 2017, pp. 2529–2537. DOI: 10.1109/CVPR.2017.271 (cit. on pp. 107, 108).

[120] Sylwan, S. “The Application of Vision Algorithms to Visual Effects Production”. In: *Asian Conference on Computer Vision*. Springer Berlin Heidelberg. 2010, pp. 189–199. DOI: 10.1007/978-3-642-19315-6_15 (cit. on p. 17).

[121] Szeliski, R. “Prediction Error as a Quality Metric for Motion and Stereo”. In: *International Conference on Computer Vision*. IEEE, 1999, pp. 781–788. DOI: 10.1109/ICCV.1999.790301 (cit. on pp. 6, 8, 17, 21, 24, 32, 33, 42).

[122] Szeliski, R. and Zabih, R. “An Experimental Comparison of Stereo Algorithms”. In: *Vision Algorithms: Theory and Practice*. Springer Berlin Heidelberg, 2000, pp. 1–19 (cit. on pp. 21, 42).

[123] Taniai, T., Matsushita, Y., Sato, Y., and Naemura, T. “Continuous 3D Label Stereo Matching using Local Expansion Moves”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017). DOI: 10.1109/TPAMI.2017.2766072 (cit. on p. 42).

[124] Tao, M., Hadap, S., Malik, J., and Ramamoorthi, R. “Depth from Combining Defocus and Correspondence Using Light-Field Cameras”. In: *International Conference on Computer Vision*. IEEE, 2013, pp. 673–680. DOI: 10.1109/ICCV.2013.89 (cit. on pp. 101, 102).

[125] Thacker, N. and Courtney, P. “Statistical Analysis of a Stereo Matching Algorithm”. In: *British Machine Vision Conference* (1992), pp. 33.1–33.11. DOI: 10.5244/C.6.33 (cit. on p. 22).

[126] Thacker, N., Clark, A., Barron, J., Ross Beveridge, J., Courtney, P., Crum, W., Ramesh, V., and Clark, C. “Performance Characterization in Computer Vision: A Guide to Best Practices”. In: *Computer Vision and Image Understanding* 109.3 (2008), pp. 305–334. DOI: 10.1016/j.cviu.2007.04.006 (cit. on pp. 21–23).

[127] Tombari, F., Mattoccia, S., Di Stefano, L., and Addimanda, E. “Classification and Evaluation of Cost Aggregation Methods for Stereo Correspondence”. In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8. DOI: 10.1109/CVPR.2008.4587677 (cit. on p. 21).

[128] Tombari, F., Mattoccia, S., and Di Stefano, L. "Stereo for Robots: Quantitative Evaluation of Efficient and Low-Memory Dense Stereo Algorithms". In: *International Conference on Control Automation Robotics & Vision*. IEEE. 2010, pp. 1231–1238. DOI: 10.1109/ICARCV.2010.5707826 (cit. on pp. 6, 17, 18).

[129] Torralba, A. and Efros, A. "Unbiased Look at Dataset Bias". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2011, pp. 1521–1528. DOI: 10.1109/CVPR.2011.5995347 (cit. on p. 2).

[130] Tsai, C.-J. and Katsaggelos, A. "Dense Disparity Estimation with a Divide-and-Conquer Disparity Space Image Technique". In: *Transactions on Multimedia* 1.1 (1999), pp. 18–29. DOI: 10.1109/6046.748168 (cit. on p. 28).

[131] Tulyakov, S., Ivanov, A., and Fleuret, F. "Weakly Supervised Learning of Deep Metrics for Stereo Reconstruction". In: *International Conference on Computer Vision*. IEEE. 2017, pp. 1348–1357. DOI: 10.1109/ICCV.2017.150 (cit. on pp. 14, 28).

[132] van der Mark, W. and Gavrila, D. "Real-Time Dense Stereo for Intelligent Vehicles". In: *Transactions on Intelligent Transportation Systems* 7.1 (2006), pp. 38–50. DOI: 10.1109/TITS.2006.869625 (cit. on pp. 6, 14, 17, 18, 23, 24, 26, 27, 35, 38).

[133] van der Mark, W., van den Heuvel, J., and Groen, F. "Stereo Based Obstacle Detection with Uncertainty in Rough Terrain". In: *Intelligent Vehicles Symposium*. IEEE. 2007, pp. 1005–1012. DOI: 10.1109/IVS.2007.4290248 (cit. on p. 18).

[134] Vandewalle, P. and Varekamp, C. "Disparity Map Quality for Image-Based Rendering Based on Multiple Metrics". In: *International Conference on 3D Imaging*. IEEE, 2014, pp. 1–5. DOI: 10.1109/IC3D.2014.7032599 (cit. on pp. 17, 18, 21, 30).

[135] Varekamp, C., Hinnen, K., and Simons, W. "Detection and Correction of Disparity Estimation Errors via Supervised Learning". In: *International Conference on 3D Imaging*. IEEE, 2013, pp. 1–7. DOI: 10.1109/IC3D.2013.6732078 (cit. on pp. 24, 28, 33).

[136] Vargas, C., Cabezas, I., and Branch, J. "Stereo Correspondence Evaluation Methods: A Systematic Review". In: *International Symposium on Visual Computing*. Springer. 2015, pp. 102–111. DOI: 10.1007/978-3-319-27863-6_10 (cit. on p. 23).

[137] Vaudrey, T., Rabe, C., Klette, R., and Milburn, J. "Differences Between Stereo and Motion Behavior on Synthetic and Real-World Stereo Sequences". In: *International Conference of Image and Vision Computing New Zealand*. IEEE, 2008, pp. 1–6. DOI: 10.1109/IVCNZ.2008.4762133 (cit. on pp. 35, 37).

[138] Vianello, A. "Robust 3D Surface Reconstruction from Light Fields". PhD thesis. 2017. DOI: 10.11588/heidok.00023819 (cit. on p. 18).

[139] Vosoughi, S., Ameli, E., and Wildes, R. *Evaluation of Computer Vision Stereo Algorithms for Surgical Applications*. Tech. rep. [2018-09-30]. Department of Electrical Engineering & Computer Science, York University, 2014. URL: http://www.eecs.yorku.ca/research/techreports/2014/EECS-2014-01.pdf (cit. on p. 17).

[140] Wang, J., Yang, Z., and Wu, Y. “Effect of Calibration Error on Reconstruction Accuracy of Stereovision System”. In: *Open Automation and Control Systems Journal* 5 (2013), pp. 30–37. DOI: 10.2174/1874444301305010030 (cit. on pp. 35, 38).

[141] Wang, T.-C., Efros, A., and Ramamoorthi, R. “Occlusion-Aware Depth Estimation Using Light-field Cameras”. In: *International Conference on Computer Vision*. IEEE. 2015, pp. 3487–3495. DOI: 10.1109/ICCV.2015.398 (cit. on pp. 19, 101, 102).

[142] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. “Image Quality Assessment: From Error Visibility to Structural Similarity”. In: *Transactions on Image Processing* 13.4 (2004), pp. 600–612. DOI: 10.1109/TIP.2003.819861 (cit. on p. 29).

[143] Wang, Z., Simoncelli, E., and Bovik, A. “Multi-Scale Structural Similarity for Image Quality Assessment”. In: *Asilomar Conference on Signals, Systems and Computers*. Vol. 2. IEEE. 2003, pp. 1398–1402. DOI: 10.1109/ACSSC.2003.1292216 (cit. on p. 29).

[144] Wang, Z., Bovik, A., and Lu, L. “Why is Image Quality Assessment so Difficult?” In: *International Conference in Acoustics, Speech, and Signal Processing*. IEEE. 2002, pp. 3313–3316. DOI: 10.1109/ICASSP.2002.5745362 (cit. on p. 29).

[145] Wanner, S., Strähle, C., and Goldlücke, B. “Globally Consistent Multi-label Assignment on the Ray Space of 4D Light Fields”. In: *Conference on Computer Vision and Pattern Recognition*. 2013, pp. 1011–1018. DOI: 10.1109/CVPR.2013.135 (cit. on p. 19).

[146] Wanner, S. and Goldlücke, B. “Reconstructing Reflective and Transparent Surfaces from Epipolar Plane Images”. In: *German Conference on Pattern Recognition*. Springer Berlin Heidelberg, 2013, pp. 1–10. DOI: 10.1007/978-3-642-40602-7_1 (cit. on p. 19).

[147] Wanner, S. and Goldlücke, B. “Variational Light Field Analysis for Disparity Estimation and Super-Resolution”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.3 (2014), pp. 606–619. DOI: 10.1109/TPAMI.2013.147 (cit. on pp. 19, 101, 102).

[148] Wanner, S., Meister, S., and Goldlücke, B. “Datasets and Benchmarks for Densely Sampled 4D Light Fields”. In: *Vision, Modeling and Visualization*. The Eurographics Association, 2013, pp. 225–226. DOI: 10.2312/PE.VMV.VMV13.225-226 (cit. on pp. 24, 26, 48, 102, 104, 105).

[149] Wetzstein, G. *Synthetic Light Field Archive*. http://web.media.mit.edu/~gordonw/SyntheticLightFields/index.php. [2018-09-30] (cit. on p. 102).

[150] Yamaguchi, K., Mcallester, D., and Urtasun, R. “Efficient Joint Segmentation, Occlusion Labeling, Stereo and Flow Estimation”. In: *European Conference on Computer Vision*. Springer International Publishing. 2014, pp. 756–771. DOI: 10.1007/978-3-319-10602-1_49 (cit. on pp. 46, 62, 79).

[151] Yan, T. and Zhao, Q. “Fast Disparity Refinement with Occlusion Handling for Stereo Matching”. In: (2018). To appear in IEEE TIP 2018 (cit. on p. 42).

[152] Yuille, A. and Poggio, T. *A Generalized Ordering Constraint for Stereo Correspondence*. Tech. rep. [2018-09-30]. Massachusetts Institute of Technology Cambridge, Artificial Intelligence Lab, 1984. URL: http://18.7.29.232/handle/1721.1/6404 (cit. on p. 16).

[153] Zbontar, J. and LeCun, Y. “Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches”. In: *Journal of Machine Learning Research* 17.1-32 (2016), p. 2 (cit. on p. 41).

[154] Zendel, O., Honauer, K., Murschitz, M., Humenberger, M., and Fernández, G. “Analyzing Computer Vision Data - The Good, the Bad and the Ugly”. In: *Conference on Computer Vision and Pattern Recognition*. IEEE. 2017, pp. 1980–1990. DOI: 10.1109/CVPR.2017.706 (cit. on pp. 2, 35, 38, 121, 124).

[155] Zendel, O., Murschitz, M., Humenberger, M., and Herzner, W. “CV-HAZOP: Introducing Test Data Validation for Computer Vision”. In: *International Conference on Computer Vision*. IEEE. 2016, pp. 2066–2074. DOI: 10.1109/ICCV.2015.239 (cit. on p. 23).

[156] Zendel, O., Murschitz, M., Humenberger, M., and Herzner, W. “How Good Is My Test Data? Introducing Safety Analysis for Computer Vision”. In: *International Journal of Computer Vision* 125.95 (2017), pp. 1–15. DOI: 10.1007/s11263-017-1020-z (cit. on pp. 7, 35, 38, 102).

[157] Zendel, O., Honauer, K., Murschitz, M., Steininger, D., and Fernández, G. “WildDash - Creating Hazard-Aware Benchmarks”. In: *European Conference on Computer Vision*. 2018 (cit. on pp. 121, 124).

[158] Zhang, C., Li, Z., Cai, R., Chao, H., and Rui, Y. “As-Rigid-As-Possible Stereo under Second Order Smoothness Priors”. In: *European Conference on Computer Vision*. Springer, 2014, pp. 112–126. DOI: 10.1007/978-3-319-10605-2_8 (cit. on p. 46).

[159] Zhang, S., Xie, W., Zhang, G., Bao, H., and Kaess, M. “Robust Stereo Matching with Surface Normal Prediction”. In: *International Conference on Robotics and Automation*. IEEE. 2017, pp. 2540–2547. DOI: 10.1109/ICRA.2017.7989295 (cit. on p. 42).

[160] Zhang, S., Sheng, H., Li, C., Zhang, J., and Xiong, Z. “Robust Depth Estimation for Light Field via Spinning Parallelogram Operator”. In: *Computer Vision and Image Understanding* 145 (2016), pp. 148–159. DOI: 10.1016/j.cviu.2015.12.007 (cit. on pp. 19, 46, 108).

[161] Zhang, Z., Hou, C., Shen, L., and Yang, J. “An Objective Evaluation for Disparity Map Based on the Disparity Gradient and Disparity Acceleration”. In: *International Conference on Information Technology and Computer Science*. 2009, pp. 452–455. DOI: 10.1109/ITCS.2009.98 (cit. on pp. 24, 34).