

# **Progressive Refinement Imaging by Variable-Resolution Image and Range Fusion**

DISSERTATION

ZUR ERLANGUNG DES GRADES EINES  
DOKTORS DER INGENIEURSWISSENSCHAFTEN (DR.-ING.)

vorgelegt von

**MARKUS KLUGE, M. SC.**

eingereicht bei der Naturwissenschaftlich-Technischen Fakultät  
der Universität Siegen

Siegen, 2024



Betreuer und erster Gutachter

**Prof. Dr. Andreas Kolb**  
**Universität Siegen**

Zweiter Gutachter

**Prof. Dr. Tim Weyrich**  
**Friedrich-Alexander-Universität Erlangen-Nürnberg**

Tag der mündlichen Prüfung

17. Juli 2024



# Abstract

In the past years, various algorithmic approaches have been proposed that address the fusion of multiple camera observations, enabling the acquisition of scenes that cannot be captured with a single photograph. Despite various improvements in seamless image blending, a key challenge to creating a convincing composite remains in compensating for geometric and photometric discrepancies (due to, for example, changes in viewpoint and illumination conditions). While previous methods mitigate these inconsistencies mainly through global optimization, any kind of computationally intensive post-processing prevents an acquisition in an interactive, online fashion.

In this thesis, novel methods for fusing a stream of camera observations into a *progressively refined*, consistent image representation are proposed. By enriching a low-resolution image with high-resolution details from close-ups, the user is allowed to interactively increase resolution locally where added image detail is desired.

First, a method is proposed to fuse an RGB image sequence with substantial geometric and photometric discrepancies into a single consistent output image. It can handle large sets of images, acquired from a nearly planar or far-distant scene at variable object-space resolutions and under varying local or global illumination conditions. At its core, a dynamically extendable multi-scale representation allows for *variable-resolution* image fusion. Details from the incoming image data are selectively merged in a way that removes artifacts such as lens distortions, lighting changes, or varying exposure and color balance.

Second, by bridging between 2D and 3D approaches, a *disparity-corrected* method is proposed that allows adaptive image refinement for general 3D scenes, even in the presence of silhouettes and strong scene parallax. It features the fusion of handheld RGB-D camera streams into a high-quality, *variable-resolution* 2.5-D reconstruction (color and range data). This is enabled by a parallax-aware image warping, assisted by adaptively refined depth values to compensate for parallax effects due to depth disparities. All pipeline modules are designed for resilience against low-resolution, artifact-prone depth readings while refining the high-resolution color data.



# Zusammenfassung

In den letzten Jahren wurden verschiedene algorithmische Ansätze zur Fusion von Kameraaufnahmen vorgestellt, welche die Akquisition von Szenen ermöglichen, die nicht mit einer einzigen Photographie erfasst werden können. Um eine überzeugende Bildkomposition zu erreichen, besteht, trotz zahlreicher Fortschritte in der Erzeugung nahtloser Bildübergänge, weiterhin die primäre Herausforderung, geometrische und photometrische Diskrepanzen auszugleichen (z. B. aufgrund von Änderungen des Blickpunktes oder der Beleuchtungsbedingungen). Während bisherige Methoden Inkonsistenzen hauptsächlich durch eine globale Optimierung beheben, verhindert jede Art von rechenintensiver Nachbearbeitung eine interaktive Bildakquisition.

In dieser Dissertation werden neuartige Methoden vorgestellt, welche eine Zusammenführung von Kameraaufnahmen in eine *progressiv verfeinerte*, konsistente Bildrepräsentation ermöglichen. Durch die Anreicherung niedrig aufgelösten Bildmaterials mit hochaufgelösten Details aus Nahaufnahmen kann der Benutzer interaktiv die Auflösung lokal dort erhöhen, wo zusätzliche Bilddetails gewünscht sind.

Zunächst wird eine Methode präsentiert, welche eine Sequenz von Farbaufnahmen mit erheblichen geometrischen und photometrischen Diskrepanzen zu einem einzigen, konsistenten Ausgangsbild fusioniert. Sie unterstützt die Verarbeitung großer Mengen an Bilddaten, welche von einer nahezu ebenen oder weit entfernten Szene und unter variierenden lokalen oder globalen Beleuchtungsbedingungen aufgenommen wurden. Als Schlüsselkomponente ermöglicht eine dynamisch erweiterbare Multiskalenrepräsentation die Fusion von Bildmaterial mit variabler Auflösung in der Objektdomäne. Details aus den eingehenden Bilddaten werden selektiv derart zusammengeführt, so dass Artefakte wie Linsenverzerrungen, Beleuchtungsänderungen oder unterschiedliche Belichtungen und Farbbalancen entfernt werden.

Anschließend wird durch die Kombination von 2D- und 3D-Ansätzen eine Methode vorgestellt, welche die adaptive Bildverfeinerung für allgemeine 3D-Szenen ermöglicht – selbst bei Vorhandensein von Silhouetten und stark ausgeprägter Parallaxe. Dabei werden die Datenströme einer handgeführten RGB-D-Kamera zu einer hochwertigen 2,5-D-Rekonstruktion (Farb- und Entfernungsdaten) mit *variabler Auflösung* fusioniert. Ermöglicht wird dies durch

eine geometrische Bildtransformation (engl. warping), welche, von adaptiv verfeinerten Tiefenwerten unterstützt, die durch Tiefendisparitäten induzierten Parallaxeneffekte berücksichtigt und korrigiert. Alle Pipeline-Module sind so konzipiert, dass sie gegen Artefakt anfällige Tiefenwerte mit geringer Auflösung resistent sind, während zugleich die hochauflösenden Farbdaten verfeinert werden.



# Acknowledgments

**F**irst and foremost, I would like to thank my advisor, *Prof. Dr. Andreas Kolb*, for giving me the opportunity to be part of his research group, for working until 2 a.m. before a paper deadline, and for constantly supporting me with his guidance and knowledge throughout the entire process. His enthusiasm has been inspiring. This thesis would not have been possible without him.

I am also very thankful to my co-advisor, *Prof. Dr. Tim Weyrich*, for investing his time and effort over the years. His scientific expertise, precise thinking, and thought-provoking feedback have been invaluable and greatly enriched the quality of my work.

Many thanks go to my friends and colleagues (mostly) at the Computer Graphics Group: *Andreas Görlitz, Hamed Sarbolandi, Jonas Geiping, Ulrich Schipper, Hendrik Hochstetter, Dmitri Presnov, Tak Ming Wong, Rustam Akhunov, Hendrik Sommerhoff, Hartmut Bauermeister, Willi Gräfrath, and Sarah Wagener.*

Lastly, I want to express my deepest gratitude to my family: *Lutz, Beate, Stefanie, and Katharina Kluge.*



# Contents

<b>Acronyms and Abbreviations</b>	<b>xv</b>
<b>List of Figures</b>	<b>xviii</b>
<b>List of Tables</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Challenges . . . . .	2
1.3 Contributions . . . . .	4
1.4 Overview . . . . .	7
<b>2 Foundations</b>	<b>9</b>
2.1 Digital Imaging . . . . .	9
2.1.1 Camera Models . . . . .	9
2.1.2 Depth Imaging . . . . .	16
2.1.3 RGB-D Camera System . . . . .	20
2.1.4 Camera Calibration . . . . .	22
2.2 Multiresolution Image Representations . . . . .	24
2.2.1 Image Pyramid . . . . .	24
2.2.2 Wavelet Decomposition . . . . .	26
2.3 Image Registration . . . . .	28
2.3.1 Projective Registration . . . . .	28
2.3.2 Deformable Registration . . . . .	30
2.4 Image and Range Fusion . . . . .	33
2.4.1 Image Fusion . . . . .	33
2.4.2 3D Reconstruction with RGB-D Cameras . . . . .	36
<b>3 Progressive Refinement Imaging for Quasi-Planar Scenes</b>	<b>39</b>
3.1 Image Refinement by Variable-Resolution Image Compositing . . . . .	42
3.2 Pipeline Overview . . . . .	43

3.3	Adaptive Model Representation . . . . .	46
3.3.1	Initialization . . . . .	46
3.3.2	Adaptivity . . . . .	47
3.3.3	Level-of-Refinement Maps . . . . .	48
3.4	Progressive Refinement Imaging . . . . .	48
3.4.1	Image Registration . . . . .	48
3.4.2	Decomposing the Observation . . . . .	50
3.4.3	Outlier Removal . . . . .	51
3.4.4	Merging of the Model and Observation . . . . .	53
3.4.5	Refinement Guidance . . . . .	54
3.5	Results . . . . .	54
3.5.1	Refinement Using Different Sources of Imagery . . . . .	58
3.5.2	Inconsistent Illumination . . . . .	58
3.5.3	Inconsistent Scene Geometry . . . . .	62
3.5.4	Ablation Study . . . . .	63
3.5.5	Comparison of Required Resources . . . . .	67
3.5.6	Limitations and Discussion . . . . .	67
3.6	Summary . . . . .	68
<b>4</b>	<b>Depth-Assisted Progressive Refinement Imaging for 3D Scenes</b>	<b>71</b>
4.1	Photometric Scene Reconstruction with Parallax Compensation . . . . .	74
4.2	Pipeline Overview . . . . .	76
4.3	Adaptive RGB-D Model Representation . . . . .	79
4.4	Depth-Assisted Progressive Refinement Imaging . . . . .	80
4.4.1	Pre-processing . . . . .	80
4.4.2	Camera Pose Estimation . . . . .	81
4.4.3	Model Correspondence . . . . .	83
4.4.4	Parallax-Aware Warping . . . . .	85
4.4.5	Local Color Consistency . . . . .	86
4.4.6	Fusion . . . . .	88
4.4.7	Final Output . . . . .	91
4.5	Implementation . . . . .	92
4.6	Results . . . . .	93
4.6.1	Data Sets . . . . .	93
4.6.2	Ablation Study . . . . .	96
4.6.3	Qualitative Comparisons . . . . .	98

---

4.6.4	Robustness against Self-Localization Drift . . . . .	104
4.6.5	Quantitative Comparison . . . . .	105
4.6.6	Performance . . . . .	109
4.6.7	Limitations . . . . .	109
4.7	Summary . . . . .	111
<b>5</b>	<b>Conclusions</b>	<b>113</b>
5.1	Summary . . . . .	113
5.2	Future Work . . . . .	114
	<b>Bibliography</b>	<b>115</b>



# Acronyms and Abbreviations

CCD	Charge-Coupled Device
CMOS	Complementary Metal-Oxide-Semiconductor
CoC	Circle of Confusion
DoF	Degrees of Freedom
DoF	Depth of Field
DoG	Difference of Gaussians
DoH	Determinant of Hessian
DWT	Discrete Wavelet Transform
FFD	Freeform Deformation
HDR	High Dynamic Range
ICP	Iterative Closest Point
IR	Infrared
LoG	Laplacian of Gaussian
LPIPS	Learned Perceptual Image Patch Similarity
MAE	Mean Absolute Error
MMSE	Minimum Mean Square Error
MTF	Modulation Transfer Function
NIR	Near-Infrared
PSNR	Peak Signal-to-Noise Ratio
RANSAC	Random Sample Consensus
RBF	Radial Basis Function
RGB	Red Green Blue
RGB-D	Red Green Blue-Depth
RMSE	Root Mean Square Error
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SPD	Spectral Power Distribution
SSIM	Structural Similarity Index Measure
SURF	Speeded Up Robust Features
SVD	Singular Value Decomposition
ToF	Time-of-Flight
TSDf	Truncated Signed Distance Function





# List of Figures

1.1	<i>The Two Ways of Life</i> by Oscar Gustav Rejlander . . . . .	2
1.2	Sample result of progressive refinement imaging . . . . .	5
2.1	Perspective projection using a pinhole . . . . .	10
2.2	Perspective projection using a thin lens . . . . .	12
2.3	Focus and out-of-focus . . . . .	12
2.4	Spherical aberration . . . . .	13
2.5	Types of optical distortion effects . . . . .	14
2.6	Calculating the depth from triangulation . . . . .	17
2.7	Calculating the depth by measuring the phase shift . . . . .	18
2.8	Points related by a homography . . . . .	23
2.9	Gaussian pyramid generation in 1D . . . . .	25
2.10	Equivalent Gaussian and Laplacian kernels . . . . .	26
2.11	2D wavelet decomposition . . . . .	27
2.12	Laplacian of Gaussian for blob detection . . . . .	28
2.13	Super-resolution from a set of low-resolution images . . . . .	36
3.1	Results for the data set <i>House of Neptune and Amphitrite mosaic</i> . . . . .	41
3.2	Progressive refinement imaging pipeline for planar scenes . . . . .	44
3.3	Adaptive Laplacian model pyramid . . . . .	47
3.4	Positioning of the observation within the model pyramid . . . . .	49
3.5	Rendering of the level-of-refinement map . . . . .	54
3.6	Results for the data set <i>Panorama at different daytimes</i> . . . . .	60
3.7	Results for the data set <i>Wall painting at different daytimes</i> . . . . .	60
3.8	Results for the data set <i>Glossy poster</i> . . . . .	61

3.9	Results for the data set <i>Deësis mosaic</i> . . . . .	61
3.10	Results for the data set <i>Moving cars</i> . . . . .	62
3.11	Results for the data set <i>Streetart fisheye</i> . . . . .	63
3.12	Influence of locally refined image registration . . . . .	64
3.13	Influence of per-pixel outlier removal . . . . .	65
3.14	Image versus Laplacian levels merging . . . . .	65
3.15	Blending versus replacing . . . . .	66
4.1	Progressive refinement imaging pipeline for 3D scenes . . . . .	76
4.2	Layout of the model representation . . . . .	78
4.3	1D graphic representation of the level-of-refinement map . . . . .	84
4.4	Outlier removal scheme . . . . .	87
4.5	Fusion of erroneous pixels at depth discontinuities . . . . .	91
4.6	Voting scheme for depth fusion . . . . .	92
4.7	Unrefined reference images of the data sets . . . . .	94
4.8	Level-of-refinement maps . . . . .	95
4.9	Ablation study for color reconstruction . . . . .	97
4.10	Ablation study for depth reconstruction . . . . .	98
4.11	Comparison with 2D progressive refinement imaging . . . . .	100
4.12	Comparison with online scene reconstruction methods (1) . . . . .	101
4.13	Comparison with online scene reconstruction methods (2) . . . . .	102
4.14	Comparison with an offline, post-processing approach . . . . .	104
4.15	Comparison of reconstructed depths . . . . .	105
4.16	Robustness against self-localization drift . . . . .	105
4.17	Results (color) for the data set <i>BunnySynth</i> . . . . .	107
4.18	Error maps (color) for the data set <i>BunnySynth</i> . . . . .	107
4.19	Results (depth) for the data set <i>BunnySynth</i> . . . . .	108
4.20	Absolute distance error (depth) for the data set <i>BunnySynth</i> . . . . .	108

# List of Tables

3.1	List of conventions . . . . .	45
3.2	Comparison to 2D imaging methods . . . . .	57
3.3	Resources required for the complete refinement process . . . . .	68
4.1	List of conventions . . . . .	77
4.2	Data set specifications . . . . .	96
4.3	Voxel sizes used by the competing methods . . . . .	100
4.4	Quantitative evaluation (color) for the data set <i>BunnySynth</i> . . .	106
4.5	Quantitative evaluation (depth) for the data set <i>BunnySynth</i> . . .	106
4.6	Required resources . . . . .	110



# 1

## Introduction

*This chapter provides a brief introduction to the background and the main objectives of image data fusion. It describes the challenges involved and outlines the main contributions made in this thesis. The chapter closes with an overview of the structure of this dissertation.*

---

### 1.1 Motivation

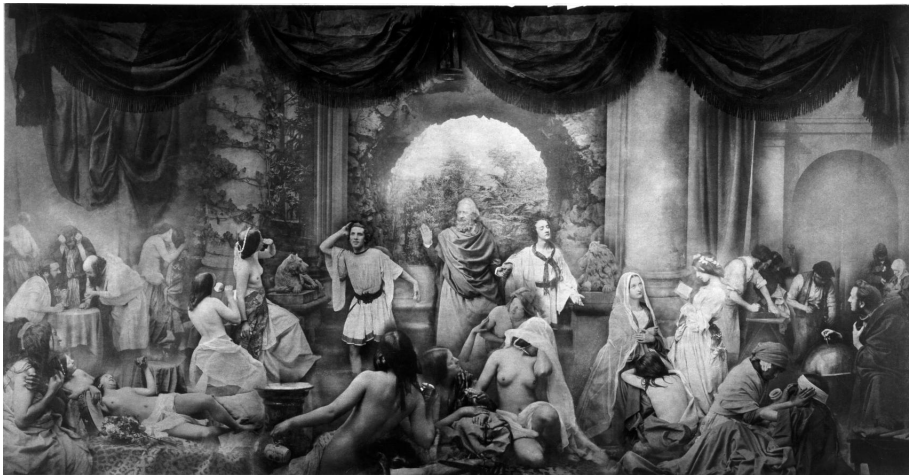
In the mid-19th century, photo montage evolved as a photographic art form. [Image compositing](#) Rejlander [Rej57], for example, composed the allegorical photo “The Two Ways of Life”, a photomontage of 32 carefully composed and feathered pictures (see Figure 1.1). Robinson [Rob69] discussed principles on how to arrange form, light, and shadow to create the perfect photo composition in the context of the aesthetics ideal of the “Picturesque”, a concept popularized in the mid-18th century.

Today, applications of photo montage have gone well beyond the artistic medium, and digital workflows employ modern-day equivalents that build upon works such as digital photomontage [ADA\*04] and image mosaicing [Mil75]. Various computational methods for image recombination and fusion have been developed to acquire scenes or objects that cannot be captured with a single photograph. For example, panoramic photography extends an image laterally, by creating a wide-angle mosaic from a set of images with narrower field of view. Moreover, several algorithmic approaches have been proposed to overcome the resolution limits of digital imaging, creating a higher-resolution image by fusing detail information from multiple sensor observations. [Digital imaging](#)

In doing so, the modality used is not limited to color photography but also includes satellite imaging, microscopy, computed tomography, and range imaging, to name a few. The latter, in particular, has gained much attention in the Computer Graphics and Vision community, where range scanners are used to merge partial range scans (known as “2.5-D” or depth images) into an entire 3D model [CL96].

Regardless of the intended purpose, all approaches follow the common goal of *creating a convincing composite by combining multiple observations into a single, consistent representation*. To achieve this, global optimization techniques are generally used to process all input images in a post-process, after capture. However, the past decade has seen an emergence of scene digitization systems that progressively fuse a stream of observations. The key benefits of “online” systems over offline approaches are the interactive user guidance, due to its immediate availability of intermediate results, and the continuous elimination of redundancy, thus reducing computing resources and taking the burden of efficiency-conscious view planning from the user. This principle is now prominently used both for 2D imaging (e.g., panorama mode in mobile phone camera applications) and 3D model reconstruction [RHHL02], popularized through the introduction of consumer RGB-D (color and depth) cameras.

Online processing



**Figure 1.1:** “The Two Ways of Life” by Oscar Gustav Rejlander, a pioneering image fusion from 1857. After photographing the background and each figure separately, Rejlander combined 32 individual negatives into a seamlessly mounted composite. (From [Uni57], by courtesy of the University of Michigan Library.)

## 1.2 Challenges

In digital image fusion, the key challenge is to combine image data into a consistent composite without leaving visible traces. The reason why this task is far from trivial is due to a discrepancy between the individual images – they are inconsistent with each other. In general, this inconsistency can be categorized into two types: geometric and photometric inconsistency.

Geometric inconsistencies occur when geometric structures of the individual images differ in scale and shape, preventing a convincing composite. The main reason for this is a variable vantage point from which the images have been acquired, thus changing the size and shape of the imaged objects. In addition, the projection of the three-dimensional scene onto the two-dimensional image plane results in a number of discrepancies. Varying the camera's distance from the scene changes the relative scale of near and far objects, resulting in differing images due to perspective distortion (extension and compression distortion<sup>1</sup>). Parallax effects occur as an apparent change in the position of objects when observing the scene from different viewpoints, resulting in a shift of the object's image against the more distant background. Furthermore, the lens of the optical system causes several optical aberrations, including radially symmetric distortions where straight lines bend inwards or outwards in the image (pincushion and barrel distortion).

Geometric  
inconsistency

Photometric inconsistencies, however, are intensity-related discrepancies in the appearance of the source imagery. The main reason for this can be found in variable illumination conditions, i.e., the amount of light, the type of light (ambient or directional), the color temperature, or the position and direction of the light sources, causing global as well as local brightness and color variations (e.g., shadows). Additionally, the camera system may introduce changes in intensity due to different exposure settings, which further control the amount of light entering the lens. The color appearance may also be influenced by the camera's color balance, a global adjustment of the color intensities acquired by the sensor.

Photometric  
inconsistency

Both geometric and photometric inconsistencies might arise when capturing a dynamic scene, where the image content changes over time, for example, due to moving objects or alteration of materials (e.g., photodegradation and (bio-)deterioration in the context of cultural heritage). Lastly, various artifacts and errors can occur, some specific to other modalities. For example, the entire image or only parts thereof may be out of focus (though the "bokeh" effect may be intentional). In range imaging, for instance, an interfering signal can lead to a misinterpretation of the actual distance if the active light travels multiple indirect paths due to reflections.

Besides that, changing the distance to the scene, zooming, or using varying image sensors leads to a different kind of discrepancy: a variable resolution in the object space (i.e., the ability to reproduce object details). In this case, storing and combining variable-resolution images without losing high-detail information or redundantly storing low-resolution data becomes necessary. This raises the question of an efficient data representation that is suitable for manip-

Efficient image  
representation

---

<sup>1</sup>At short distances, objects in the foreground appear distorted and abnormally large, unlike at long distances where light rays are almost parallel.

ulating, merging, storing, and accessing image data at multiple scales. Since the achievable lateral resolution is usually unlimited (especially in panoramic photography), it should also allow for scalability up to the gigapixel range, which makes compression, navigation, and viewing further essential challenges.

Global and  
progressive fusion

If we now want to create a consistent fusion from the inconsistent data, two different approaches can generally be taken. Either all pre-captured images are made globally consistent with each other and fused afterward (offline), or the processing alternates progressively between optimization and fusion with each new image fed into the pipeline (online). Both of these approaches face their own challenges. Global optimization and fusion require that all images (or at least batches of them) are processed jointly without exceeding the capacity of memory and processing time. Using a progressive approach, on the other hand, the so-far accumulated fusion is only optimized with respect to the current source image. Consequently, valuable information for reaching consistency is missing, as even with a progressive approach, all images combined should ultimately produce a globally consistent result.

## 1.3 Contributions

This thesis presents *progressive refinement imaging*, an innovative approach to fuse a stream of camera observations into a progressively refined scene representation. Unlike the usual lateral scan in panoramic photography or 3D reconstruction, *progressive refinement imaging* deliberately aims at zooming in or at a “walking closer to the scene”-like camera path (known amongst photographers as “zooming with one’s feet”). The overall assumption here is that by approaching the scene, subsequent frames provide novel geometric and photometric details to increase resolution locally where more detail is required (see Figure 1.2). In order to achieve this, a processing pipeline for variable-resolution image and range fusion is proposed.

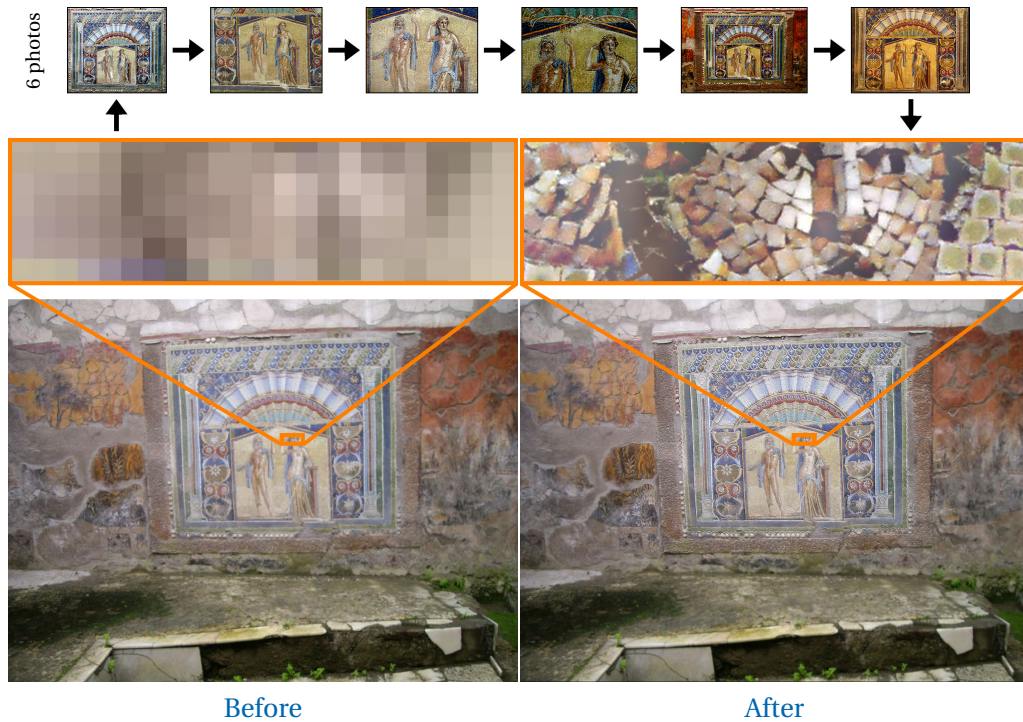
The following contributions have been made in this thesis.

Progressive  
refinement imaging

**Progressive refinement imaging** With *progressive refinement imaging*, a sequence of RGB input images with substantial geometric and photometric discrepancies can be fused into a single consistent and *refined* output image (see Figure 1.2). In order to achieve a globally consistent result despite a progressive approach, a low-resolution reference image is used to guide the refinement process. That is, the user first takes an overview shot before walking into the scene or zooming in to take higher-resolution close-ups where added image detail is desired. It enables:



- Adaptive image refinement using image sequences from different sources and viewpoints, without requiring calibration, pre-alignment, external tracking, lighting adjustment, or other intervention.
- The consistent fusion of images acquired from a nearly planar or far-distant scene at variable object-space resolutions and under varying local or global illumination conditions.
- The fusion of hundreds of images by continuously eliminating redundancy.
- Interactive user guidance for casual capture and dynamic refinement, as the method's *progressive* nature provides intermediate results at any moment during the refinement process.



**Figure 1.2:** A sample result (*right*) of the proposed *progressive refinement imaging* approach. The initial image (*left*) is refined by progressively fusing 6 additional photos (*top row*) taken closer to the scene. Even though the source images exhibit strong photometric inconsistencies (e.g., different color balance), the fused and refined result is consistent with the initial image. (Source images are from [Dav07, ALM06, Amp16, HK13, Ras17, Cra14, Rie09], all with permission to reuse with modification.)

At its core, a dynamically extendable multi-scale representation allows for *variable-resolution* image fusion. To reach geometric consistency, a coarse-to-fine alignment strategy is first applied to compensate for different viewpoints, perspective distortion as well as local lens distortions. A frequency-oriented color merging then fuses color differences while retaining the base color of the reference image, achieving photometric consistency without requiring local or global optimization for color harmonization. Finally, local artifacts inconsistent with the reference image (e.g., moving objects) are compensated using a per-pixel outlier removal.

### **Progressive refinement imaging with depth-assisted disparity correction**

While 2D imaging approaches like *progressive refinement imaging* support a variety of input imagery, they are, however, prone to parallax-induced artifacts and, thus, strictly limited to scenes with minimal depth disparity. While a full 3D reconstruction may offer a more comprehensive capture of the scene, 2D imaging remains the most popular modality in the mainstream, mainly due to most output devices being 2D. Thus, the proposed approach aims at overcoming this restriction by progressively reconstructing an auxiliary depth map alongside a *refined* image reconstruction. By bridging between 2D and 3D approaches, the proposed method offers:

Depth-assisted  
disparity correction

- *Disparity-corrected* image refinement for general 3D scenes, even in the presence of silhouettes and strong scene parallax, while retaining photometric consistency.
- The *progressive* fusion of handheld RGB-D camera streams into a high-quality, *variable-resolution* 2.5-D reconstruction (color and depth).

This is enabled by a parallax-aware image warping, assisted by adaptively refined depth values to guide the camera's self-localization and compensate for parallax effects due to depth disparities. The pipeline modules are designed for resilience against low-resolution, artifact-prone depth readings while refining the high-resolution color data. This is further achieved by introducing a hierarchical color and depth representation that strictly decouples color data from the coarse and potentially incomplete geometry.

**List of publications** The following provides a list of all publications achieved during this research. The contributions made in this thesis have been presented in [KWK20] and [KWK23] while being inspired by the research done in 3D reconstruction [LKS\*17].

- [LKS\*17] DAMIEN LEFLOCH, MARKUS KLUGE, HAMED SARBOLANDI, TIM WEYRICH, ANDREAS KOLB. *Comprehensive Use of Curvature for Robust and Accurate Online Surface Reconstruction*. In IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 39, no. 12, pp. 2349-2365, 2017.
- [KWK20] MARKUS KLUGE, TIM WEYRICH, ANDREAS KOLB. Progressive Refinement Imaging. In *Computer Graphics Forum (CGF)*, vol. 39, no. 1, pp. 360-374, 2020.
- [KWK23] MARKUS KLUGE, TIM WEYRICH, ANDREAS KOLB. Progressive Refinement Imaging with Depth-Assisted Disparity Correction. In *Computers & Graphics*, vol. 115, pp. 446-460, 2023.

## 1.4 Overview

*Chapter 2* introduces the theoretical foundations of this thesis, including models and error sources in the acquisition of color and range images, as well as their representation, registration, and fusion. *Chapter 3* presents the *progressive refinement imaging* approach for quasi-planar scenes, the online integration of uncalibrated RGB image sequences with substantial geometric and/or photometric discrepancies into a single, geometrically and photometrically consistent image. *Chapter 4* extends this idea to general 3D scenes, enabling progressive refinement of both colors and depths by ingesting RGB-D images from a handheld depth camera. Finally, *Chapter 5* summarizes this thesis and provides possible directions of future work.



# 2

## Foundations

*This chapter introduces the fundamental concepts for this thesis. Therefore, models and principles in digital image acquisition are presented, including depth imaging, their error sources, and the resulting effects (see Section 2.1). It follows an introduction to adaptive and hierarchical representations of the acquired images (see Section 2.2). In Section 2.3, an overview of methods for registering multiple images is given, while their fusion into a single representation is presented in Section 2.4.*

---

### 2.1 Digital Imaging

This section describes the image acquisition process by introducing fundamental camera models and principles. A strong focus is given to characteristic error sources and their visual effects, which the image processing pipeline proposed in this thesis has to cope with.

#### 2.1.1 Camera Models

##### Pinhole Camera Model

The pinhole camera model describes a simple camera system using an image plane and a barrier with a *pinhole* as an aperture (see Figure 2.1). This pinhole restricts light rays, which are reflecting from objects in the scene, from traveling through the camera system and reaching the image plane. In theory, the pinhole is reduced to a size that allows only one ray from each scene point to pass the hole [FP02].

Formally, this imaging process describes a perspective projection from 3D coordinates of points in space onto the image plane, depicted in Figure 2.1. Let  $(\mathbf{O}, \mathbf{I}, \mathbf{J}, \mathbf{K})$  be a coordinate system centered at the pinhole at  $\mathbf{O} = (0, 0, 0)^T \in \mathbb{R}^3$ , with the  $\mathbf{K}$ -axis as the *optical axis*, the  $\mathbf{J}$ -axis pointing up, and the  $\mathbf{I}$ -axis pointing to the left. Here, the intersection of the optical axis with the image

[Pinhole model](#)

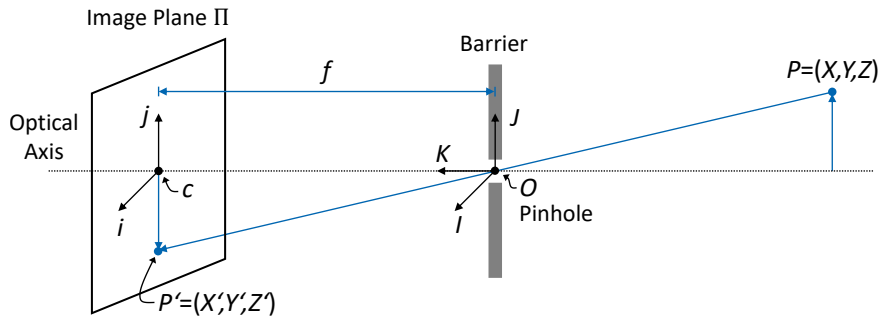


Figure 2.1: Perspective projection using a pinhole.

plane  $\Pi$  is called the *image center* or *principal point*  $c \in \mathbb{R}^2$ , while  $f \in \mathbb{R}$ , the distance between  $O$  and  $\Pi$ , is the effective *focal length* of the pinhole camera model. Furthermore, the ray  $\overrightarrow{PO}$ , originating from 3D scene point  $P = (X, Y, Z)^\top \in \mathbb{R}^3$  and passing through  $O$ , intersects the image plane at  $P' = (X', Y', Z')^\top \in \mathbb{R}^3$ . Because of the collinearity of  $P$ ,  $O$ , and  $P'$ , we have

$$\frac{X'}{f} = \frac{X}{Z} \quad \text{and} \quad \frac{Y'}{f} = \frac{Y}{Z}, \quad (2.1)$$

and thus, scene point  $P = (X, Y, Z)^\top$  is mapped to the image plane by

$$P' = \begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \begin{pmatrix} f \frac{X}{Z} \\ f \frac{Y}{Z} \\ f \end{pmatrix} \in \mathbb{R}^3. \quad (2.2)$$

Camera to image  
coordinates

In general, the projected points relate to their own two-dimensional coordinate system  $(c, i, j)$  in the image plane  $\Pi$ , centered at the principal point  $c$ , with axes  $i \parallel I$  and  $j \parallel J$  (see Figure 2.1). Hence, the projective transformation from camera coordinates in  $\mathbb{R}^3$  to image coordinates in  $\mathbb{R}^2$  is defined by

$$p' = \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} f \frac{X}{Z} \\ f \frac{Y}{Z} \end{pmatrix} \in \mathbb{R}^2. \quad (2.3)$$

### Intrinsic Camera Matrix

The image plane of a digital camera is equipped with a charge-coupled device (CCD) or a complementary metal-oxide semiconductor (CMOS) sensor with a specific image resolution in *pixel* units. In contrast to the image coordinate system  $(c, i, j)$ , the origin of the pixel coordinates is not defined at the center of the image but at the top-left corner. This offset to the principal point, expressed in pixels, leads to a translation by the vector  $(c_u, c_v)^\top$ . Thus, the projective

transformation from camera coordinates in  $\mathbb{R}^3$  to pixel coordinates in  $\mathbb{R}^2$  can be written as Camera to pixel coordinates

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f_u \frac{X}{Z} + c_u \\ f_v \frac{Y}{Z} + c_v \end{pmatrix} \in \mathbb{R}^2, \quad (2.4)$$

$$\text{with } f_u = kf \quad \text{and} \quad f_v = lf,$$

where  $f_u$  and  $f_v$  are the focal length  $f$  expressed in terms of pixels, using the additional scale factors  $k$  and  $l$  in pixel/m, separately for both axes to accommodate for a potentially rectangular pixel size.

By using homogeneous coordinates, Equation (2.4) can be written in matrix form as

$$\begin{pmatrix} u \\ v \end{pmatrix} = \pi \left( \begin{pmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \right) = \pi(\mathbf{K} \mathbf{P}) \in \mathbb{R}^2, \quad (2.5)$$

with  $\mathbf{K}$  being the *intrinsic camera matrix* and the function  $\pi(\tilde{X}, \tilde{Y}, \tilde{Z}) = (\tilde{X}/\tilde{Z}, \tilde{Y}/\tilde{Z})^\top$  the de-homogenization<sup>1</sup>. Note that  $\mathbf{K}$  can be multiplied by a homogeneous 4-vector  $(X, Y, Z, 1)^\top$  by adding an extra column  $(0, 0, 0)^\top$ , i.e.,  $\mathbf{K} = \begin{pmatrix} f_u & 0 & c_u & 0 \\ 0 & f_v & c_v & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ . The camera's intrinsic parameters  $f_u$ ,  $f_v$ , and  $(c_u, c_v)^\top$  are estimated by performing a *camera calibration*, e.g., using Zhang's approach [Zha00]. See Section 2.1.4 for further details. Intrinsic parameters

## Camera Lens

Today's cameras are equipped with lenses to focus light to avoid problems occurring with small pinhole sizes; in particular: (1) the amount of light reaching the image plane decreases, resulting in dark images, and (2) a pinhole size comparable to the wavelength of the incoming light causes the image to blur due to *diffraction* effects (bending of waves around the pinhole edges; for more details, see, e.g., [Hec12]).

Figure 2.2 shows a simple thin lens model to focus rays of light emitting from scene point  $\mathbf{P}$ , which are immediately refracted when entering the lens and eventually converging to point  $\mathbf{P}'$ . All rays parallel to the optical axis, such as the blue-colored ray, are focused on the *focal point*  $\mathbf{F}'$ . Using such a lens with a negligible thickness (much less than the radii of curvature), an approximation for calculating its focal length  $f$  is given by the *thin lens equation* Thin lens equation

$$\frac{1}{|Z'|} + \frac{1}{|Z|} = \frac{1}{f}, \quad (2.6)$$

<sup>1</sup>The triple  $(kx, ky, k)^\top$  with  $k \neq 0$  represents a set of homogeneous coordinates for the same (finite) point  $(x, y)^\top$  of  $\mathbb{R}^2$ , which can be recovered by dividing by  $k$ .

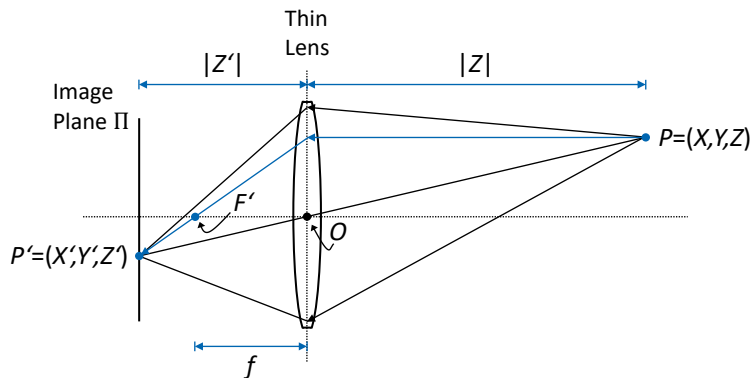


Figure 2.2: Perspective projection using a thin lens.

with  $|Z'|$  being the image distance,  $|Z|$  the object distance and  $f$  the lens's focal length. If the object distance is  $|Z| = \infty$ , then the image distance  $|Z'|$  equals the lens's focal length  $f$ , i.e.,  $|Z'| = f$ , and in this case, Equations (2.2) to (2.5) for the pinhole model apply.

Focus and  
out-of-focus

With a CCD/CMOS sensor erected at image plane  $\Pi$ , scene point  $P$  is in sharp focus. However, as depicted in Figure 2.3, if the object distance is changed, rays emitting from scene point  $Q$  converge in point  $Q'$  behind  $\Pi$ , creating the blur circle  $b$  on  $\Pi$  (*circle of confusion, CoC*); the object is *out of focus*. The effective range of object distances for which objects are in acceptably sharp focus<sup>2</sup> is called the *depth of field (DoF)*. By using an adjustable aperture to limit the cone of lights entering the lens (with diameter  $d$  in Figure 2.3), the depth of field can be increased (the narrower the spread of the cone, the less the image blurs). Furthermore, the focus of a camera system is adjusted by either moving the entire lens or lens elements in relation to each other (*internal focus*).

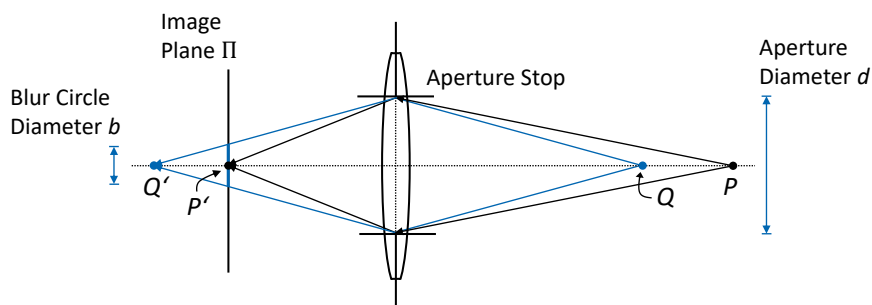


Figure 2.3: Focus and out-of-focus.

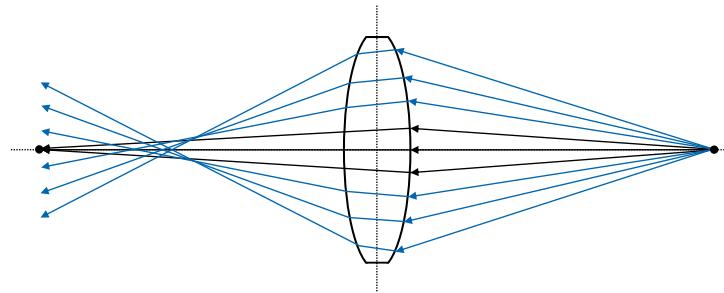
<sup>2</sup>The acceptable diameter of the circle of confusion for depth of field calculations can be computed, e.g., using the so-called *Zeiss formula*; see, e.g., [Hec12].



### Primary Lens Aberrations

A simple lens model, such as the *thin lens* model, assumes the so-called *paraxial approximation*; that is, for each ray, its angle  $\alpha$  to the optical axis is assumed to be small, i.e.,  $\sin \alpha \approx \alpha$ . Thus, *Snell's Law*, according to which the sine of the angle of refraction is directly proportional to the sine of the angle of incidence, can be approximated. However, the difference between the first-order approximation and third-order optics leads to *optical aberrations*, causing the image to be blurred or distorted. There are five monochromatic, *primary aberrations* (*Seidel aberrations*): *Spherical aberration*, *coma*, *astigmatism*, *field curvature*, and *distortion*. The first four types of aberrations have in common that light rays are not focused on the same point, causing the image to blur. This effect increases with rays refracted at lens positions further from the optical axis (non-paraxial rays), as seen in Figure 2.4 for spherical aberration.

Spherical aberration



**Figure 2.4:** Spherical aberration occurs when light rays pass the periphery of a lens with spherical surfaces. Rays intersecting the lens further away from its center converge at different positions on the optical axis.

Coma, astigmatism, and field curvature occur when incident rays from an *off-axis* scene point (offset from the optical axis) enter the lens at an angle, causing: (1) a characteristic comet-shaped image of a point source, due to different effective focal lengths the further away a ray hits the lens from its center (coma), or (2) two different foci from ray bundles in the sagittal and tangential plane, entering the lens asymmetrically (astigmatism), or (3) a single point of focus but on a curved image “plane”, preventing a uniformly sharp image when using a flat image sensor (field curvature). For more details, see, e.g., [Hec12].

Coma, astigmatism,  
field curvature

Radial distortion, however, alters the overall shape of the image because of different focal lengths in different areas of a lens and, hence, different lateral magnifications. Distortion causes straight lines (see Figure 2.5a) to be curved outward if the magnification increases with distance from the axis (pincushion distortion, see Figure 2.5b), or inward with decreasing magnification (barrel

Radial distortion

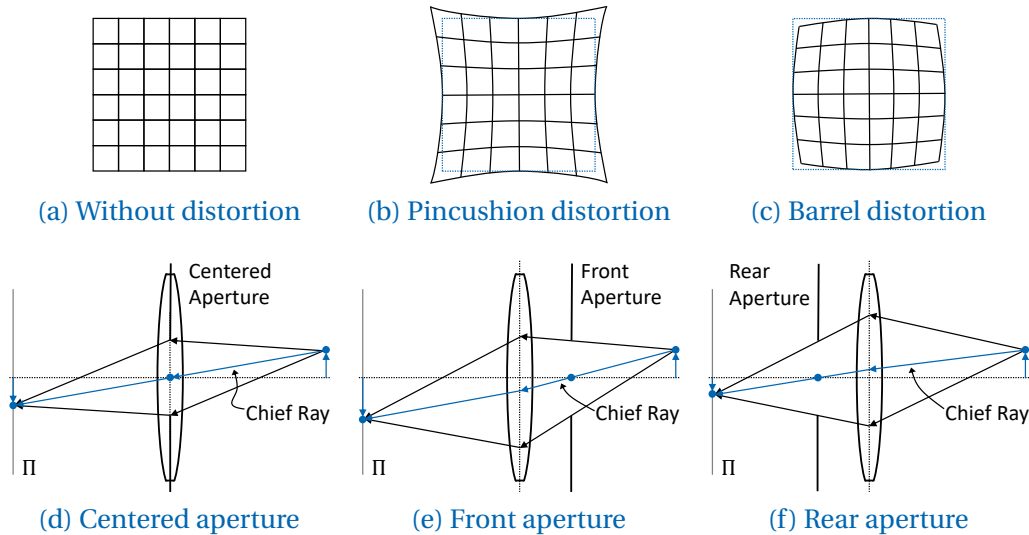
distortion, see Figure 2.5c). Furthermore, the position of the aperture stop introduces and influences distortion effects. For an off-axis object point, the stop causes the bundle of rays to travel asymmetrically through the optical system and, thus, changes the *chief ray's*<sup>3</sup> angle of incidence. With a stop in front of the lens, the object-to-image distance increases; hence, the magnification decreases, as seen in Figure 2.5e. With a stop placed behind the lens, the opposite effect occurs (see Figure 2.5f).

The radial displacement from the ideal image coordinates  $(x', y')^T$  to the distorted coordinates  $(x'_{\text{dist}}, y'_{\text{dist}})^T$  can be modeled by [Zha00],

$$\begin{pmatrix} x'_{\text{dist}} \\ y'_{\text{dist}} \end{pmatrix} = \begin{pmatrix} x' + x'(k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y' + y'(k_1 r^2 + k_2 r^4 + k_3 r^6) \end{pmatrix}, \quad (2.7)$$

with  $r = \sqrt{x'^2 + y'^2}$ ,

where  $k_1, k_2, k_3, \dots$  are the radial distortion coefficients and  $r$  is the Euclidean distance of  $(x', y')^T$  to the distortion center, which is assumed to be the principal point  $c$ . Typically, two coefficients  $k_1, k_2$  are sufficient, and  $k_3$  is included for severe distortions.



**Figure 2.5:** (a) - (c): Types of optical distortion effects. (d) - (f): The position of the aperture stop affects the incident angle of the ray bundle and, thus, the magnification.

<sup>3</sup>The chief ray is the ray that passes through the center of the aperture stop.

Other error sources are:

- *Tangential distortion*, which is not caused by the lens design but occurs when the image plane is not aligned perfectly parallel to the lens. Further error sources
- *Vignetting*, which is a brightness drop-off toward the periphery of the image. For off-axis scene points, the cone of light passing through the optical system is partially blocked if two or more stops are used, e.g., an aperture or boundary of a lens. Moreover, it is caused by the natural light fall-off (rays further away from the optical axis have to travel a longer distance).
- *Chromatic aberration* appears when using non-monochromatic light. A lens's ability to bend light (*refractive index*) varies with wavelength and, thus, its effective focal length. As a consequence, different colors of light focus on different positions, causing a separation of focus along the optical axis, i.e., only one color plane is in sharp focus (axial chromatic aberration) and a difference of magnification in the color planes (lateral chromatic aberration).

### Camera Exposure

By modifying the amount of light exposed to the image sensor, the brightness (intensity) of the photographic result can be controlled. *Exposure* is a combination of illuminance and exposure time, expressed by Exposure

$$H = Et, \quad (2.8)$$

where  $E$  is the image plane's illuminance, measured in *lux* (lx), and  $t$  is the exposure time in s. While  $E$  is manipulated by adjusting the aperture size, the *shutter speed* controls the exposure time  $t$ .

The shutter speed is a precise amount of time the camera's shutter is open and light enters the optical system. Its downside is potentially generating *motion blur* caused by a camera or object movement within this period.

The relative aperture of a camera lens is expressed by the *f-number*

$$N = \frac{f}{d}, \quad (2.9)$$

with  $f$  being the focal length and  $d$  the aperture's diameter. By increasing the f-number, the aperture stop's size is narrowed (*stopping down*), and hence, the amount of light entering the lens is reduced. This affects the depth of field and may introduce blur.

The visual effect of improper exposure is a loss of detail in the bright (*overexposure*) or dark areas (*underexposure*).

## White Balance

Visible light is usually a mixture of various wavelengths. Depending on the distribution of energy at each wavelength (*spectral power distribution, SPD*), different light sources produce different photographic results in terms of color appearance [JRAA00].

**Color temperature** To describe the color appearance of a light source, the term *color temperature* is commonly used. It indicates a similar SPD to that of a *Planckian radiator*<sup>4</sup> (*black body*) at a specific temperature in *Kelvin* (K). With increasing temperature, the color changes from red (1000 K) through yellowish (2700–3000 K) to bluish (over 5000 K).

**White balance** *White balance* refers to adjusting the colors acquired by the sensor so that a white object appears white for a particular color temperature of the scene illuminant. This is commonly achieved by applying a color correction matrix to the three color channels. *Auto white balance* refers to estimating the scene illuminant by using a *color constancy* algorithm, e.g., by searching for the lightest patch to use as a white reference (*white patch Retinex* algorithm based on the Retinex theory [LM71]) or by assuming natural color statistics across the image pixels. However, capturing a scene with mixed lighting conditions or under artificial light may lead to an inconsistent color appearance.

### 2.1.2 Depth Imaging

Depth imaging refers to methods and techniques that acquire depth maps (or range images), i.e., the depth at which the ray corresponding to a pixel intersects the surface of a scene. As this does not provide full 3D information of the scene, the term *2.5-D* is used for this kind of representation. Principles for depth imaging can be divided into triangulation-based and *Time-of-Flight (ToF)* methods.

#### Depth Imaging Principles

**Stereovision** *Stereovision* is a passive depth imaging method that involves two 2D cameras observing the scene. Using *triangulation*, the depth of a scene point can be calculated from its *disparity*, that is, the difference in image location of the same scene point, projected to both cameras. As illustrated in Figure 2.6, in case the displacement  $b$  (*baseline*) of the two horizontally displaced camera centers  $O_1 \in \mathbb{R}^3$  and  $O_2 \in \mathbb{R}^3$  is known, the depth  $|Z|$  of scene point  $P = (X, Y, Z) \in$

<sup>4</sup>A Planckian radiator is an idealized light source according to *Planck's law*, which emits light whose SPD depends only on its temperature.

$\mathbb{R}^3$  can be calculated by exploiting the similarity of triangles ( $\Delta PO_1O_2$  and  $\Delta Pp'_1p'_2$ ), where

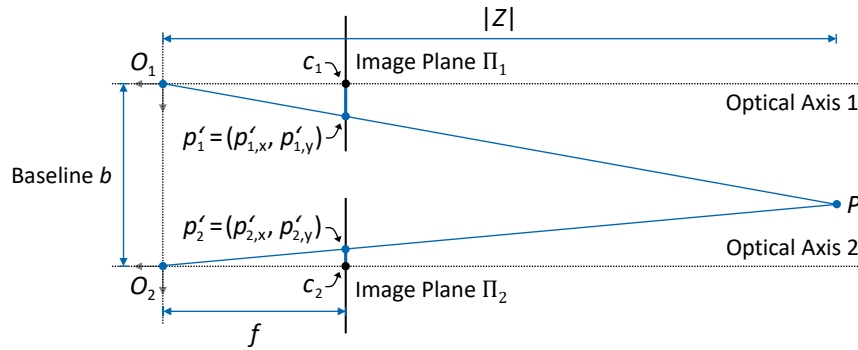
$$\frac{b}{|Z|} = \frac{b - d_{\text{disp}}}{|Z| - f}, \quad (2.10)$$

$$\text{with } d_{\text{disp}} = p'_{1,x} - p'_{2,x},$$

and thus,

$$|Z| = \frac{fb}{d_{\text{disp}}}, \quad (2.11)$$

with  $d_{\text{disp}}$  being the disparity of both projections  $p'_1 = (p'_{1,x}, p'_{1,y}) \in \mathbb{R}^2$ ,  $p'_2 = (p'_{2,x}, p'_{2,y}) \in \mathbb{R}^2$  and  $f$  being the focal length.



**Figure 2.6:** Calculating the depth  $|Z|$  of a scene point  $P$  from triangulation by forming the triangles  $\Delta PO_1O_2$  and  $\Delta Pp'_1p'_2$ . Triangulation is based on  $P$ 's projections  $p'_1, p'_2$  onto the (virtual) image planes  $\Pi_1, \Pi_2$  of two horizontally displaced cameras with optical centers  $O_1, O_2$  and principal points  $c_1, c_2$ .

*Structured light*, on the other hand, uses a single 2D camera together with an active illumination unit. For example, with Microsoft's *Kinect* or Asus' *Xtion Pro Live*, a near-infrared (NIR) emitter projects a fixed, dot-based pattern onto the scene. Depending on the depths of the objects in the scene, the pattern is deformed and captured by a horizontally displaced, passive IR camera. By comparing the observed pattern with the known reference pattern, i.e., calculating its local disparity, the depth can be estimated by triangulation. Correspondences between both patterns are extracted using, e.g., a sliding correlation window with  $9 \times 7$  px or  $9 \times 9$  px [KP15].

Structured light

Another active method for depth imaging is based on the Time-of-Flight principle, which is used, e.g., in Microsoft's *Kinect v2*. To estimate the depth, the time required for the emission of a light signal to the scene and its return to the sensor is measured. The most common approach to this is *Continuous Wave Intensity Modulation*, where an intensity-modulated light with modulation frequency  $f_m$  is emitted. Depending on the distance the wave traveled, the

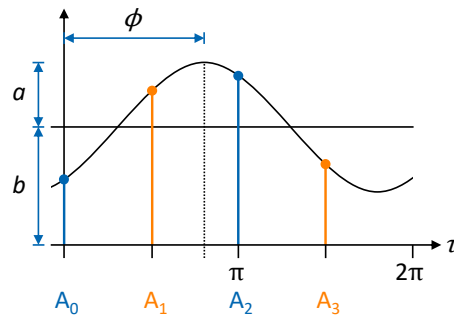
Time-of-Flight

phase between the emitted and the received signal will be shifted. This phase shift is estimated by measuring the similarity between both signals using the cross-correlation

$$C(\tau) = s \otimes g = \lim_{T \rightarrow \infty} \int_{-T/2}^{T/2} s(t) \cdot g(t + \tau) dt, \quad (2.12)$$

between the emitted signal  $g(t)$  and the received signal  $s(t)$ , with  $\tau$  being the offset parameter. For an emitted sinusoidal signal  $g(t) = \cos(2\pi f_m t)$  with modulation frequency  $f_m$ , the received signal can be expressed by  $g(t) = b + a \cos(2\pi f_m t + \phi)$ . Here,  $a$  is the amplitude of the received signal (depending on the object's reflectivity),  $b$  is an offset due to ambient illumination, and  $\phi$  is the phase shift; see Figure 2.7. Some basic trigonometric calculus reveals

$$C(\tau) = \frac{a}{2} \cos(f_m \tau + \phi) + b. \quad (2.13)$$



**Figure 2.7:** Measuring the phase shift  $\phi$  for calculating the depth by sampling the correlation function at four equally spaced intervals.  $a$  is the amplitude and  $b$  is an offset due to ambient illumination.

The phase offset can then be obtained by sampling the correlation function  $C(\tau)$  at four equally spaced intervals  $\tau_0 = 0, \tau_1 = \frac{\pi}{2}, \tau_2 = \pi, \tau_3 = \frac{3\pi}{4}$  (see Figure 2.7):

$$\begin{aligned} \phi &= \arctan2(A_3 - A_1, A_0 - A_2) \\ &= \arctan2(C(\tau_3) - C(\tau_1), C(\tau_0) - C(\tau_2)), \end{aligned} \quad (2.14)$$

that is, the ratio of  $A_3 - A_1$  and  $A_0 - A_2$  is equal to the tangent of the phase angle. From the measured phase shift  $\phi$ , the distance  $d_{\text{dist}}$  the light has traveled can be calculated by

$$d_{\text{dist}} = \frac{c \phi}{2\pi f_m}, \quad (2.15)$$

with  $c = 3 \cdot 10^8$  m/s being the speed of light. Considering that the light traveled the distance twice, the depth between the sensor and the object is  $Z = 1/2 d_{\text{dist}}$ .

Therefore,  $f_m = c/\lambda$  determines the appropriate modulation frequency to avoid ambiguity if the wavelength  $\lambda$  is set to twice the desired maximum possible range, e.g., for a 7.5 m operating range,  $f_m = c/15\text{ m} = 20\text{ MHz}$ .

### Error Sources and Their Effects

A systematic distance error occurs in depth cameras utilizing the structured light approach, mainly due to inaccurate measurements of disparities because of restrictions in resolution and quantization [KE12, SLK15]. For the Time-of-Flight technology, however, a systematic error is caused by the assumption of a perfect sinusoidal signal, which, in practice, is not met due to hardware limitations or design choices [KP15, Rap07]. Instead, both the emitted and the received signal contain higher-order harmonics, which are not accounted for in Equation (2.14). This causes an error in the phase measurements and, thus, leads to the so-called *wiggling error* in the depth estimations, which is periodic to the real depth. Systematic distance error

Inhomogeneous depth values may occur at depth discontinuities. Due to the large displacement of the projector and the IR camera in structured light devices (e.g., a 7.5 cm baseline for Kinect), the projected light cannot reach occluded regions visible for the IR camera; thus, shadow regions with invalid/unknown depths are created close to object silhouettes. For Time-of-Flight devices, light reflecting from the fore- and background may lead to a superimposed signal at depth discontinuities [KP15]. This results in a false depth estimation, with a value between both distances (*flying pixels*). Inhomogeneous depth values

*Multi-path effects* refer to a source of error where the active light travels multiple, indirect paths because of reflections between objects or semi-transparent surfaces [IKL\*10]. For the structured light approach, this may lead to a projection of the pattern onto other objects, e.g., due to highly reflective material [SLK15]. For the ToF principle, multiple returns of the emitted light from additional, indirect paths cause a superimposed signal. Multi-path effects

An intensity-related distance error can be observed for ToF cameras, manifesting in a depth bias for darker object surfaces with low NIR reflectivity [LK07], i.e., a low amount of incident light to the camera sensor. It is assumed to be caused by non-linearities of the semiconductor [Lin10] or a multi-path effect [SLK15]. Intensity-related distance error

Another source of error is camera or object motion during the sequential acquisition of the phase images  $A_0, A_1, A_2, A_3$ , resulting in erroneous depth measurements.

### 2.1.3 RGB-D Camera System

**RGB-D** RGB-D sensor systems combine an RGB camera and a depth imaging technique into a single device, acquiring both a color image  $\mathcal{I} \in \mathbb{R}^3$  with RGB intensities and a depth map  $\mathcal{D} \in \mathbb{R}$  with camera-to-surface distances (usually in meters). Each optical system is related to a common *world coordinate system* through its *extrinsic matrix*. This matrix describes the world coordinate system relative to a camera's own coordinate system.

#### Extrinsic Camera Matrix

The extrinsic camera matrix  $\mathbf{T} \in \mathbb{SE}^3$  is a transformation matrix defined as

$$\mathbf{T} = (\mathbf{R} \mid \mathbf{t}) = \left( \begin{array}{ccc|c} r_{1,1} & r_{1,2} & r_{1,3} & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & t_3 \end{array} \right), \quad (2.16)$$

World to camera  
coordinates

with translation vector  $\mathbf{t} \in \mathbb{R}^3$  and 3D rotation matrix  $\mathbf{R} \in \mathbb{SO}^3$ .  $\mathbf{T}$  defines a rigid transformation  $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ , which converts world coordinates to camera coordinates and, hence, depends on the camera's *pose*, i.e., its location and orientation in world space. Note that  $\mathbf{T}$  can be augmented by an extra row, i.e.,  $\mathbf{T} = \left( \begin{array}{c|c} \mathbf{R} & \mathbf{t} \\ \hline \mathbf{0}^\top & 1 \end{array} \right)$ , to make the matrix square and to preserve the w-component of a homogeneous 4-vector.

Extrinsic parameters

Let  $\mathbf{C} \in \mathbb{R}^3$  be the location of the camera center in world coordinates and  $\mathbf{R}_C \in \mathbb{SO}^3$  the camera's orientation, i.e., the directions of the camera axes in world coordinates. Then, the relationship between the camera pose  $[\mathbf{R}_C \mid \mathbf{C}] \in \mathbb{SE}^3$  and the extrinsic matrix  $[\mathbf{R} \mid \mathbf{t}]$  is

$$\begin{aligned} \left( \begin{array}{c|c} \mathbf{R} & \mathbf{t} \\ \hline \mathbf{0}^\top & 1 \end{array} \right) &= \left( \begin{array}{c|c} \mathbf{R}_C & \mathbf{C} \\ \hline \mathbf{0}^\top & 1 \end{array} \right)^{-1} \\ &= \left( \left( \begin{array}{c|c} \mathbf{I} & \mathbf{C} \\ \hline \mathbf{0}^\top & 1 \end{array} \right) \left( \begin{array}{c|c} \mathbf{R}_C & \mathbf{0} \\ \hline \mathbf{0}^\top & 1 \end{array} \right) \right)^{-1} \\ &= \left( \begin{array}{c|c} \mathbf{R}_C^\top & \mathbf{0} \\ \hline \mathbf{0}^\top & 1 \end{array} \right) \left( \begin{array}{c|c} \mathbf{I} & -\mathbf{C} \\ \hline \mathbf{0}^\top & 1 \end{array} \right) \\ &= \left( \begin{array}{c|c} \mathbf{R}_C^\top & -\mathbf{R}_C^\top \mathbf{C} \\ \hline \mathbf{0}^\top & 1 \end{array} \right), \end{aligned} \quad (2.17)$$

and therefore,

$$\begin{aligned} \mathbf{R} &= \mathbf{R}_C^\top \\ \mathbf{t} &= -\mathbf{R}_C \mathbf{C}. \end{aligned} \quad (2.18)$$



### RGB-D Registration

The 3D scene point  $\mathbf{P}_D = (X_D, Y_D, Z_D)^\top \in \mathbb{R}^3$  in the *camera coordinate system* of the depth imager is first transformed into world coordinates and then mapped to RGB camera coordinates by RGB-D registration

$$\mathbf{P}_{\text{RGB}} = \begin{pmatrix} X_{\text{RGB}} \\ Y_{\text{RGB}} \\ Z_{\text{RGB}} \end{pmatrix} = \mathbf{T}_{\text{RGB}} \mathbf{T}_D^{-1} \mathbf{P}_D \in \mathbb{R}^3, \quad (2.19)$$

with  $\mathbf{T}_{\text{RGB}}$  and  $\mathbf{T}_D$  being the extrinsic matrix of the RGB and depth camera, respectively<sup>5</sup>. Note that by setting the coordinate system of the RGB camera as the world coordinate system, i.e.,  $\mathbf{T}_{\text{RGB}} = \mathbf{I}$ , the mapping reduces to  $\mathbf{P}_{\text{RGB}} = \mathbf{T}_D^{-1} \mathbf{P}_D$ , with  $\mathbf{T}_D^{-1}$  now being the absolute pose of the depth camera.

To establish a per-pixel correspondence between both optical systems, a projection from one *pixel coordinate system* to the other is performed, utilizing their extrinsic camera matrices as well as their intrinsic camera matrices  $\mathbf{K}_{\text{RGB}}$  and  $\mathbf{K}_D$  (see Section 2.1.1). Therefore, 2D pixel coordinates  $\mathbf{p}_D = (x_D, y_D)^\top \in \mathbb{N}^2$  of the depth sensor are mapped to RGB pixel coordinates by

$$\mathbf{p}_{\text{RGB}} = \begin{pmatrix} x_{\text{RGB}} \\ y_{\text{RGB}} \end{pmatrix} = \pi \left( \mathbf{K}_{\text{RGB}} \mathbf{T}_{\text{RGB}} \mathbf{T}_D^{-1} \underbrace{Z_D \mathbf{K}_D^{-1} (\mathbf{p}_D, 1)^\top}_{\text{back-projection}} \right) \in \mathbb{R}^2, \quad (2.20)$$

using the *back-projection*

Back-projection

$$\mathbf{P}_D = \begin{pmatrix} X_D \\ Y_D \\ Z_D \end{pmatrix} = Z_D \mathbf{K}_D^{-1} (\mathbf{p}_D, 1)^\top \in \mathbb{R}^3 \quad (2.21)$$

from 2D pixel coordinates  $\mathbf{p}_D$  to 3D camera coordinates  $\mathbf{P}_D$ , where  $Z_D = \mathcal{D}(\mathbf{p}_D) \in \mathbb{R}$  is the corresponding depth value in the depth map  $\mathcal{D} \in \mathbb{R}$  in Cartesian coordinates.

By applying Equation (2.20) to each pixel of the depth map  $\mathcal{D}$ , the pixel mapping  $\mathcal{W}_{D \rightarrow \text{RGB}}(x, y) = \pi \left( \mathbf{K}_{\text{RGB}} \mathbf{T}_{\text{RGB}} \mathbf{T}_D^{-1} \mathcal{D}(x, y) \mathbf{K}_D^{-1} (x, y, 1)^\top \right)$  is calculated. That is, each regular lattice grid position  $(x, y) \in [0, \dots, x_{\text{max}}] \times [0, \dots, y_{\text{max}}]$  within  $\mathcal{D}$  is mapped to an irregular sub-pixel coordinate. The color image  $\mathcal{I}$  can then be projected onto the depth camera's image plane by performing the *backward remapping*

$$\mathcal{I}_{\text{RGB} \rightarrow D}(x, y) = \mathcal{I}(\mathcal{W}_{D \rightarrow \text{RGB}}(x, y)), \quad (2.22)$$

<sup>5</sup>Note that in Equations (2.19) and (2.20), a conversion between 3-vectors and homogeneous 4-vectors (e.g., for multiplication with  $\mathbf{T}$ ) is omitted to simplify notation.

i.e., a resampling of  $\mathcal{I}$  at sub-pixel positions  $\mathcal{W}_{D \rightarrow \text{RGB}} \in \mathbb{R}^2$  using bi-linear interpolation. However, transforming the depth image into the viewpoint of the RGB camera involves triangulation and rendering, as depth values are missing for calculating the required forward mapping of RGB pixel coordinates to depth pixel coordinates (see Chapter 4 for further details).

## 2.1.4 Camera Calibration

The camera's intrinsic parameters  $f_u$ ,  $f_v$ , and  $(c_u, c_v)^\top$  can be estimated by performing a camera calibration, e.g., using Zhang's approach [Zha00] based on a *homography* estimation.

### Homography

Homography

Assuming the pinhole camera model, two planes are related by the 2D homography  $\mathbf{H}$ , a  $3 \times 3$  *projective transformation* matrix, describing the (invertible) projective mapping<sup>6</sup>  $\mathbb{P}^2 \rightarrow \mathbb{P}^2$ , i.e.,

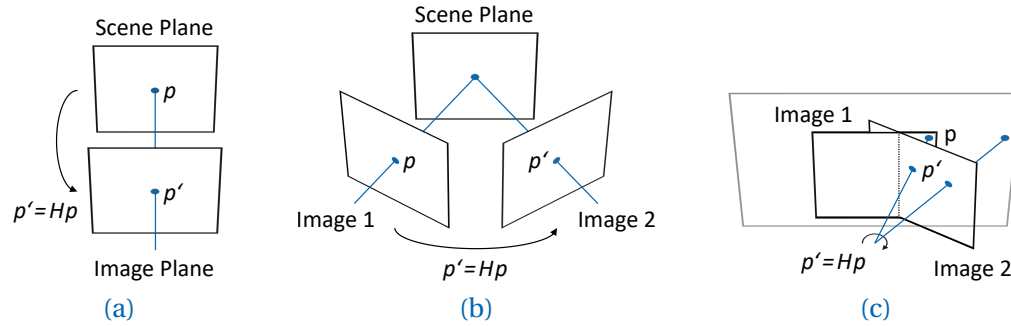
$$\mathbf{p}' = \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} \\ h_{2,1} & h_{2,2} & h_{2,3} \\ h_{3,1} & h_{3,2} & h_{3,3} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \mathbf{H} \mathbf{p} \in \mathbb{P}^2, \quad (2.23)$$

for a homogeneous vector  $\mathbf{p} \in \mathbb{P}^2$ . As illustrated in Figure 2.8, points  $\mathbf{p}$  and  $\mathbf{p}'$  are related by  $\mathbf{H}$  if they belong to: (a) either a planar surface and the image plane or (b) two images of a planar scene viewed from different camera poses. Additionally, Figure 2.8c shows a special case considering two images of an arbitrary scene geometry but viewed from a camera purely rotating about its center.

By keeping the scene plane stationary at the  $XY$  plane of the world coordinate system, i.e.,  $Z = 0$ , each 3D scene point  $\mathbf{P} = (X, Y, Z)^\top = (X, Y, 0)^\top \in \mathbb{R}^3$  can be represented by  $\mathbf{p}$  in the projective space  $\mathbb{P}^2$  using the homogeneous 3-vector  $\mathbf{p} = (X, Y, 1)^\top$ . Its linear mapping to the 2D image point  $\mathbf{p}'$  is then defined (up to scale) by

$$\mathbf{p}' = s \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \mathbf{K} \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \mathbf{K} \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{pmatrix} \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix}, \quad (2.24)$$

<sup>6</sup>The *projective space*  $\mathbb{P}^2$  is an extension of the Euclidean space  $\mathbb{R}^2$  using homogeneous coordinates  $(kx, ky, k)^\top$  and includes points at infinity, i.e., points with homogeneous coordinates  $(x, y, 0)^\top$ . For more information, see, e.g., [HZ03].



**Figure 2.8:** Points  $p$  and  $p'$  are related by the homography  $H$  if they belong to (a) a planar surface and the image plane, (b) two images of a planar scene, or (c) two images of an arbitrary scene viewed from a purely rotating camera, projecting it onto a shared plane.

with  $K = \begin{pmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{pmatrix}$  being the intrinsic matrix,  $R = (\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3) \in \mathbb{SO}^3$ ,  $\mathbf{t} \in \mathbb{R}^3$  the extrinsic parameters and, thus, a 2D homography defined by

$$H = \lambda K (\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}), \quad (2.25)$$

up to an arbitrary scale factor  $\lambda \in \mathbb{R}^+$  (see [Zha00]). Therefore, as a 2D projective transformation, the homography  $H$  has eight degrees of freedom ( $H$  can be divided by one of the nine matrix entries without changing the transformation). Thus, as each point-to-point correspondence accounts for two constraints (corresponding to the  $x$  and  $y$  components), at least four correspondences are needed to find a solution for  $H$ , e.g., by using *singular value decomposition* (SVD).

### Intrinsic and Extrinsic Parameters Estimation

By observing a planar pattern as a calibration object with  $i$  known visual features (such as the corners of a checker-board), 3D object points  $P_i$  and 2D image points  $p'_i$  are known, and, hence, a corresponding homography can be estimated. Further, by acquiring  $n$  images from a moving camera, the intrinsic matrix  $K$  and the extrinsic parameters  $R_n, \mathbf{t}_n$  can be estimated by minimizing the *reprojection error* over all  $n$  sets of  $i$  point-to-point correspondences, i.e., the sum of errors between image points  $p'_{ni}$  and the re-projected object points  $\hat{p}_{ni} = K (R_n \mathbf{t}_n) P_{ni}$ . To solve this non-linear minimization problem (e.g., using a Levenberg-Marquardt optimization), an analytical solution of  $K$  and  $R_n, \mathbf{t}_n$  can be found as initialization by exploiting constraints on the intrinsic parameters, obtained from each homography. As these constraints are derived from the rotational vectors  $\mathbf{r}_1, \mathbf{r}_2$  (by exploiting that they are orthonormal, see Camera calibration

[Zha00]), the movement of either the camera or the calibration pattern requires a rotation to be involved.

## 2.2 Multiresolution Image Representations

Fundamental tasks in image processing and computer vision involve analysis (e.g., pattern matching), manipulation, storage, and viewing of images at multiple scales. Therefore, hierarchical data structures for representing images are introduced in the following.

### 2.2.1 Image Pyramid

An image pyramid is a multiresolution data structure representing image information localized in both the spatial domain and the spatial-frequency domain. While the *Gaussian pyramid* consists of a set of low-pass filtered and subsampled copies of the original image, the *Laplacian pyramid* comprises subsampled, band-pass filtered versions that represent details at different spatial scales.

#### Gaussian Pyramid

Gaussian pyramid

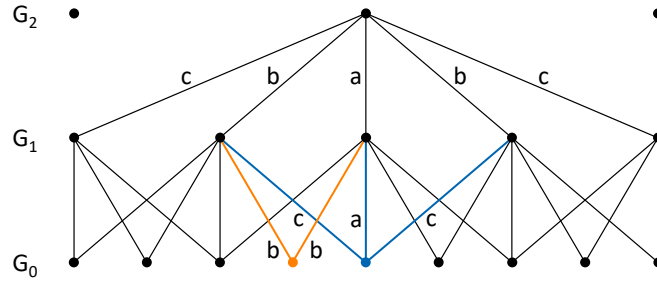
An image  $\mathcal{G}_0 \in \mathbb{R}$  (gray-scale intensities or an individual color channel of RGB) is decomposed into a Gaussian pyramid with levels  $\mathcal{G}_l$  and level indices  $l \in \mathbb{N}$  by recursively low-pass-filtering and subsampling a level in one-octave steps [AAB\*84, Bur81]. To combine the bandwidth reduction and sample rate decimation (i.e., keeping only every 2<sup>nd</sup> sample) in a single step, the down-sampling is defined in one dimension by computing only every 2<sup>nd</sup> output sample:

$$\mathcal{G}_l(x) = \sum_{i=-2}^2 w(i) \mathcal{G}_{l-1}(\underbrace{2x + i}_{\text{every 2}^{\text{nd}} \text{ output}}) \quad \text{for } l \geq 1, \quad (2.26)$$

where the discrete weighting function  $w(i)$  is a symmetric, unimodal kernel  $[c \ b \ a \ b \ c]$  with kernel elements  $a, b, c \in \mathbb{R}$ , radius  $m = 2$  and normalization  $\sum_{i=-2}^2 w(i) = 1$ . To ensure that all samples of level  $l-1$  contribute equally to the next level  $l$ , the weights  $a, b, c$  must satisfy the additional constraint  $a + 2c = 2b$  (see Figure 2.9) and, therefore,  $b = 1/4$  and  $c = 1/4 - a/2$  [Bur81]. Setting  $a = 0.375$ , the resulting kernel  $w = [0.0625 \ 0.25 \ 0.375 \ 0.25 \ 0.0625]$  approximates a Gaussian distribution; see Figure 2.10a (left).

This recursive computation is equivalent to convolving the original image  $\mathcal{G}_0$  with weighting functions  $h_l(j)$ , where, instead, the kernel's radius

Equivalent weighting function



**Figure 2.9:** Gaussian pyramid generation in 1D using the kernel  $[c \ b \ a \ b \ c]$  with a subsampling by a factor of two. The constraint  $a + 2c = 2b$  ensures that all samples of one level contribute equally to the next level, e.g., the total weight ( $c + a + c = a + 2c$ ) of the blue-colored sample is equal to the total weight ( $b + b = 2b$ ) of the orange-colored sample.

increases from level to level (see Figure 2.10a).  $h_l(j)$  for level  $l$  is defined recursively [Bur81]:

$$h_0(j) = \begin{cases} 1, & \text{if } j = 0, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$h_l(j) = \sum_{i=-2}^2 w(i) h_{l-1}(j + \underbrace{2^{l-1} i}_{\text{sample distance}}) \quad \text{for } l \geq 1, \quad (2.27)$$

where the sample distance increases in octave steps (to only include those samples of the previous level  $l-1$  that would not have been decimated). Instead of applying Equation (2.26) on subsequent levels, a specific pyramid level  $\mathcal{G}_l$  can now be obtained directly from  $\mathcal{G}_0$  by computing every  $2^l$ -th output:

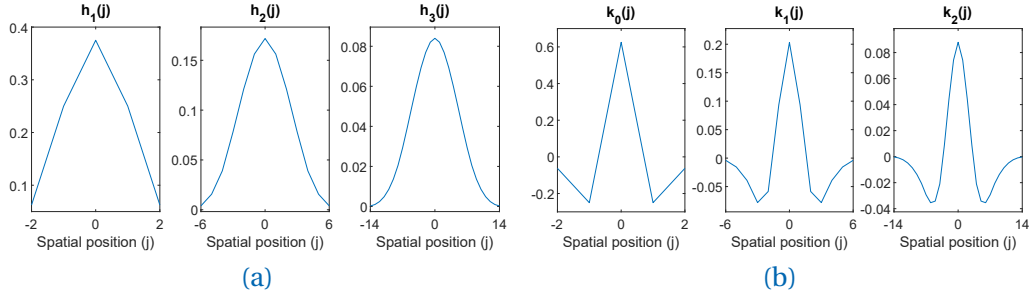
$$\mathcal{G}_l(x) = \sum_{j=-m_l}^{m_l} h_l(j) \mathcal{G}_0(2^l x + j) \quad \text{for } l \geq 1, \quad (2.28)$$

with  $m_l = \frac{s_l - 1}{2}$ ,

where  $s_l$  is the kernel's width, i.e., the number of nonzero values of  $h_l(j)$ , and  $m_l$  the kernel's radius.

### Laplacian Pyramid

The image  $\mathcal{G}_0$  is decomposed into a Laplacian pyramid [BA83a], comprising [Laplacian pyramid](#) band-pass filtered levels  $\mathcal{L}_l \in \mathbb{R}$ , by computing the differences between two



**Figure 2.10:** (a): Equivalent Gaussian (low-pass) kernels  $h_1(j)$ ,  $h_2(j)$ , and  $h_3(j)$  for generating Gaussian pyramid levels  $\mathcal{G}_1$ ,  $\mathcal{G}_2$ , and  $\mathcal{G}_3$  from the original image  $\mathcal{G}_0$ . (b): Equivalent Laplacian (band-pass) kernels  $k_0(j)$ ,  $k_1(j)$ , and  $k_2(j)$  for generating Laplacian pyramid levels  $\mathcal{L}_0$ ,  $\mathcal{L}_1$ , and  $\mathcal{L}_2$  from  $\mathcal{G}_0$ .

successive (low-pass filtered) levels  $\mathcal{G}_l$  and  $\mathcal{G}_{l+1}$  of the Gaussian pyramid:

$$\begin{aligned} \mathcal{L}_l &= \mathcal{G}_l - [\mathcal{G}_{l+1}]_{\uparrow 2} \quad \text{for } 0 \leq l < N \\ \text{and} \\ \mathcal{L}_N &= \mathcal{G}_N, \end{aligned} \tag{2.29}$$

where  $[\dots]_{\uparrow 2}$  indicates an up-sampling by one octave, with  $N+1$  being the total number of pyramid levels and  $\mathcal{L}_N \in \mathbb{R}$  the low-frequency residual equal to the top Gaussian pyramid level  $\mathcal{G}_N$ . Alternatively,  $\mathcal{L}_l$  can be generated directly from  $\mathcal{G}_0$  using equivalent Laplacian kernels  $k_l(j)$  that can be obtained by subtracting two weighting functions  $k_l(j) = h_l(j) - h_{l+1}(j)$  (see Figure 2.10b).

By reversing the steps of Equation (2.29), the original image  $\mathcal{G}_0$  can be recovered without loss of information by re-composing the Laplacian pyramid, i.e., summing all levels  $\mathcal{L}_l$  by applying

$$\mathcal{G}_l = \mathcal{L}_l + [\mathcal{G}_{l+1}]_{\uparrow 2}, \tag{2.30}$$

recursively until  $\mathcal{G}_0$  is obtained.

## 2.2.2 Wavelet Decomposition

Discrete wavelet transform

Similar to an image pyramid, an image can be decomposed into frequency bands using the *discrete wavelet transform* (DWT). In contrast to the *Fourier transform*, the wavelet transform allows a localization not only in frequency but also in space, as its basis functions have a compact support.

An arbitrary signal  $f(x)$  can be represented by the series

$$f(x) = \sum_j \sum_k d_{j,k} \psi_{j,k}(x), \tag{2.31}$$

with wavelet coefficients  $d_{j,k}$  and child wavelets

$$\psi_{j,k}(x) = \frac{1}{\sqrt{a^j}} \psi\left(\frac{x - ka^j}{a^j}\right), \tag{2.32}$$

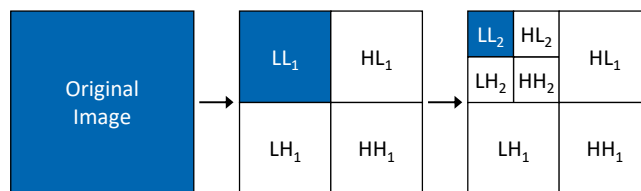
for a given *mother wavelet*  $\psi(x)$ . Here,  $\psi_{j,k}(x)$  acts as a band-pass filter, where  $j \in \mathbb{Z}$  and  $k \in \mathbb{Z}$  are parameters for discrete scaling and shifting, respectively, and  $1/\sqrt{a^j}$  is the energy normalization factor. With  $a$  usually set to  $a = 2$ ,  $\psi(x)$  is scaled and shifted by powers of two, so that halving the spatial resolution doubles the frequency resolution (as the bandwidth is halved). However, as repeatedly halving the bandwidth (ad infinitum) requires a lower bound, *multiresolution analysis* [Mal89] introduced the scaling function  $\phi$  (also called *father wavelet*), acting as a low-pass filter to cover the remaining parts of the frequency spectrum at the coarsest scale  $j = M$ :

Multiresolution analysis

$$f(x) = \underbrace{\sum_k c_{M,k} \phi_{M,k}(x)}_{\text{low-frequency residual}} + \sum_{j=1}^M \sum_k d_{j,k} \psi_{j,k}(x), \tag{2.33}$$

$$\text{with } \phi_{j,k}(x) = \frac{1}{\sqrt{a^j}} \phi\left(\frac{x - ka^j}{a^j}\right),$$

where  $c_{j,k}$  and  $d_{j,k}$  are the approximation and detail coefficients, respectively. This can be efficiently computed by recursively decomposing the low-pass band at scale  $j-1$  into a low-pass (using the scaling function  $\phi$ ) and a high-pass band (using the wavelet function  $\psi$ ) at scale  $j$  (see Figure 2.11 for the 2D case).



**Figure 2.11:** An example of the 2D wavelet decomposition. The original image (at scale 0) is first decomposed into a low-pass filtered approximation image  $LL_1$  and detail images  $LH_1$ ,  $HL_1$ , and  $HH_1$ , each downsampled to scale 1. The low-pass filtered image  $LL_1$  is then further decomposed into  $LL_2$ ,  $LH_2$ ,  $HL_2$ , and  $HH_2$ . Low-pass and high-pass filtering is applied row/column-wise, representing horizontal (LH), vertical (HL), and diagonal details (HH).

## 2.3 Image Registration

Comparing and fusing image data as part of, e.g., computer vision or medical imaging applications, requires transforming the data into a common coordinate system. Hence, this section introduces image registration as the process of finding and applying this transformation. For a broader overview of fundamental image registration methods, the reader is referred to the survey by Zitová and Flusser [ZF03].

### 2.3.1 Projective Registration

Projective registration

Using a projective registration, two images,  $\mathcal{I}_T \in \mathbb{R}$  and  $\mathcal{I}_S \in \mathbb{R}$ , of a planar or far-distant scene are aligned geometrically by finding and applying the homography  $\mathbf{H}_{S \rightarrow T}$  (see Section 2.1.4). As the required point-to-point correspondences are (usually) unknown, sparse *keypoints* are detected based on local features.

#### Feature Detection, Extraction, and Matching

Feature detection

*Feature detection* is the process of converting a 2D image  $\mathcal{I}(x, y) \in \mathbb{R}$  into a set of keypoints, also called *interest points* or landmarks. To find those interest points, structures like edges (e.g., using gradient and Laplacian edge detection or the Canny edge detector [Can86]), corners (e.g., using the Harris corner detector [HS\*88]), or distinctive blobs (patches with local appearance, e.g., using the *Laplacian of Gaussian*<sup>7</sup>, see Figure 2.12) are detected [TM\*08].

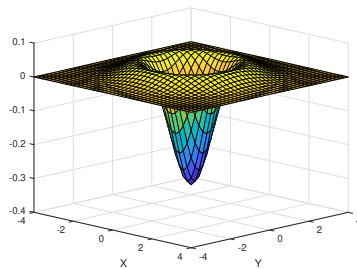


Figure 2.12: Laplacian of Gaussian (LoG) for blob detection ( $\sigma = 1$ ).

The popular SIFT (scale-invariant feature transform) algorithm [Low04], as one of the most accurate approaches [TS18], is based on the *Difference*

<sup>7</sup>The Laplacian of Gaussian (LoG) is the sum of second-order partial derivatives of the 2D Gaussian  $G_\sigma(x, y)$ , i.e.,  $\nabla^2 G_\sigma(x, y) = \frac{\partial^2 G_\sigma(x, y)}{\partial x^2} + \frac{\partial^2 G_\sigma(x, y)}{\partial y^2}$ , with  $G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$



of Gaussians<sup>8</sup> (DoG) for blob detection as a fast LoG approximation. Here, keypoints are detected by searching for local extrema in the DoG images across multiple scales, comparing each pixel to its local  $3 \times 3$  neighborhood of the current and adjacent scales. Further, the dominant orientation is assigned to each keypoint in addition to its location and scale. SURF (Speeded Up Robust Features) [BTVG06], however, accelerates SIFT by choosing points that maximize the determinant of the Hessian (DoH). Many other methods have been proposed, such as BRISK [LCS11] (corners), ORB [RRKB11] (corners), and KAZE [ABD12] (blobs), to name a few; see [MJF\*21] for a recent survey.

*Feature descriptor extraction* is computing a compact representation of the local region around each detected keypoint to compare it with other features. The SIFT descriptor, for example, is computed by accumulating the image gradient magnitudes of the local region into a  $4 \times 4$  grid of orientation histograms, each with eight orientation bins, concatenated into a 128-dimensional feature vector. To achieve invariance to image rotation, the gradients are rotated relative to the keypoint's dominant orientation. SURF, however, accumulates the horizontal and vertical Haar wavelet [Haa10] responses ( $\sum dx$ ,  $\sum |dx|$ ,  $\sum dy$ ,  $\sum |dy|$ ) of  $4 \times 4$  sub-regions, thus resulting in a 64-dimensional feature vector.

Feature descriptor extraction

*Feature matching* finds point-to-point correspondences between two images by comparing their descriptors, i.e., feature vectors. For each descriptor in the first set of features, the closest descriptor in the second set is determined using, e.g., the L2 norm as a metric. Potential feature matches are then pruned by filtering matches with a distance greater than a threshold and, e.g., by discarding ambiguous matches (Lowe's ratio test [Low04]).

Feature matching

### Image Resampling

Even after pruning potential mismatches, the detected point-to-point correspondences are affected by outliers. To robustly estimate the homography  $\mathbf{H}_{S \rightarrow T}$ , with  $\mathbf{H}_{S \leftarrow T} = \mathbf{H}_{S \rightarrow T}^{-1}$ , RANSAC (Random Sample Consensus) [FB81] is used: By repeatedly selecting a random subset of point correspondences and estimating a homography for each of these subsets, the homography is chosen that yields the most inliers among all point pairs based on the reprojection error. Image  $\mathcal{I}_S$  is then projected onto image  $\mathcal{I}_T$  by applying the backward resampling  $\mathcal{I}_{S \rightarrow T}(x, y) = \mathcal{I}_S(\pi(\mathbf{H}_{S \leftarrow T}(x, y, 1)^T))$  using bi-linear interpolation.

RANSAC

<sup>8</sup>The Difference of Gaussians is defined by  $\mathcal{D}_\sigma(x, y) = (G_{k\sigma}(x, y) - G_\sigma(x, y)) * \mathcal{I}(x, y)$ , separated in scale by  $k \in \mathbb{N}$

### 2.3.2 Deformable Registration

Deformable registration allows local alignment of two images by estimating non-rigid transformations, e.g., for registering images that are affected by optical distortions or part of a motion sequence. For a comprehensive introduction to non-rigid registration, see, e.g., [Rue01].

#### Spline-Based Deformation

In order to deform an image, the displacement of each pixel is needed. Rather than deforming an image directly on a per-pixel basis, spline-based registration techniques locally transform an image by manipulating a sparse set of control points distributed across the image. Parameterized by these control points, two-dimensional splines define a smoothly varying displacement field to map each pixel to its new location. For non-rigid transformations, two types of splines are mainly used: *B-splines* and *thin-plate splines*.

Freeform deformation

*Freeform deformation* [SP86, RSH\*99] (FFD) computes a dense displacement field from manipulating a coarse, regular mesh of control points using B-splines, i.e., piecewise polynomial curves. In the 2D case, this is expressed by the tensor-product of 1D B-splines, the affine combination of control points  $c_{i,j}$  on an  $(m_x+1) \times (m_y+1)$  lattice:

$$\mathbf{s}(x, y) = \sum_{i=0}^{m_x} \sum_{j=0}^{m_y} c_{i,j} N_i^n(x) N_j^n(y), \quad (2.34)$$

where  $N_i^n(x)$  and  $N_j^n(y)$  are B-spline basis functions of degree  $n$  [PBP02]. Since  $N_i^n(x)$ ,  $N_j^n(y)$  have minimal support, each function is nonzero in only a local neighborhood of  $(x, y)^\top$ , and thus, only some of the control points influence  $\mathbf{s}(x, y)$  at  $(x, y)^\top$ . In the cubic case ( $n=3$ ), for example,  $\mathbf{s}(x, y)$  relates to sixteen control points ( $n+1$  along each direction).

Radial basis function

Another method for non-rigid transformation is based on *radial basis functions*, particularly thin-plate<sup>9</sup> splines [Duc77]. In general, radial basis functions (RBFs) are linearly independent functions  $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$  that satisfy

$$\phi(\mathbf{x}, \mathbf{c}) = \hat{\phi}(r), \quad r = \|\mathbf{x} - \mathbf{c}\|, \quad (2.35)$$

i.e., the function value depends only on the distance  $r \in \mathbb{R}$  between an input point  $\mathbf{x} \in \mathbb{R}^n$  and the center  $\mathbf{c} \in \mathbb{R}^n$ . Thus, for different  $\mathbf{c}_i$ , functions  $\phi(\mathbf{x}, \mathbf{c}_i)$  are radially symmetric functions with the same shape but shifted to  $\mathbf{c}_i$ . In

<sup>9</sup>The name comes from the analogy of an infinite, thin metal plate that is deformed by adding point loads to the plate.

the context of data interpolation, this is used to represent a thin-plate spline, [Thin-plate spline](#) defined in two dimensions for  $\mathbf{x} = (x, y)^\top \in \mathbb{R}^2$  by

$$f(\mathbf{x}) = \underbrace{a_0 + a_1x + a_2y}_{\text{affine part}} + \underbrace{\sum_{i=1}^n b_i \phi(\|\mathbf{x} - \mathbf{c}_i\|)}_{\text{non-affine part}}, \quad (2.36)$$

with  $\phi(r) = r^2 \ln r$ ,

comprising an affine part, represented by a polynomial with coefficients  $a$ , and a linear combination of  $n$  RBFs  $\phi$  with coefficients  $b_i$ , representing a non-affine part [HD72, Boo89]. With  $f(\mathbf{x})$  being the height at  $\mathbf{x}$ , Equation (2.36) can be interpreted as a smooth surface, where the non-affine part produces deformations at  $\mathbf{c}_i$ , while at infinity, the spline approaches a flat plane (represented by the linear part).

To register two images with  $n$  corresponding landmarks,  $\mathbf{c}_i = (c_{i,x}, c_{i,y})^\top$  and  $\mathbf{c}'_i = (c'_{i,x}, c'_{i,y})^\top$ , in the source and target images, respectively, each dimension ( $x$  and  $y$ ) is considered separately. By solving for  $a$  and  $b_i$  and with  $\phi(\|\mathbf{x} - \mathbf{c}_i\|)$  centered at the source landmarks  $\mathbf{c}_i$ , two separate thin-plate splines,  $x' = f_x(\mathbf{x})$  and  $y' = f_y(\mathbf{x})$ , are determined such that a mapping to the target landmarks  $\mathbf{c}'_i$  is interpolated (i.e., fulfilling the constraints  $c'_{i,x} = f_x(\mathbf{c}_i)$  and  $c'_{i,y} = f_y(\mathbf{c}_i)$ ). That is, two smooth surfaces are constructed, one of which passes through points  $(c_{i,x}, c_{i,y}, c'_{i,x})^\top \in \mathbb{R}^3$  and the other through  $(c_{i,x}, c_{i,y}, c'_{i,y})^\top \in \mathbb{R}^3$ . The resulting function  $\mathbf{f}(x, y) = (f_x(x, y), f_y(x, y))$  provides the desired two-dimensional transformation [Gos88, Boo89].

In contrast to freeform deformations with B-splines, thin-plate splines have a global influence on the transformation. However, they allow the interpolation of arbitrary configurations of control points.

### Optical Flow

The *optical flow* estimates the per-pixel motion between two frames  $\mathcal{I}(x, y, t - 1)$  and  $\mathcal{I}(x, y, t)$ , at times  $t - 1$  and  $t$ , resulting in a 2D flow field that contains the displacement vectors  $\mathbf{d} = (u, v)^\top$  of corresponding pixels: [Optical flow](#)

$$\mathcal{I}(x, y, t - 1) = \mathcal{I}(x + u(x, y), y + v(x, y), t), \quad (2.37)$$

assuming *brightness constancy*, i.e., only the location of a pixel changes, not its intensity. By also assuming a *small motion*, the right side can be linearized

using a Taylor series expansion and truncating the higher order terms, yielding

$$\mathcal{I}(x + u(x, y), y + v(x, y), t) \approx \mathcal{I}(x, y, t - 1) + \underbrace{\mathcal{I}_x \cdot u(x, y) + \mathcal{I}_y \cdot v(x, y)}_{\approx 0} + \mathcal{I}_t, \quad (2.38)$$

with the spatial derivatives  $\mathcal{I}_x = \frac{\partial \mathcal{I}}{\partial x}$  and  $\mathcal{I}_y = \frac{\partial \mathcal{I}}{\partial y}$  and the temporal derivative  $\mathcal{I}_t = \frac{\partial \mathcal{I}}{\partial t}$ , and hence,

$$\mathcal{I}_x \cdot u + \mathcal{I}_y \cdot v + \mathcal{I}_t \approx 0. \quad (2.39)$$

Optical flow  
constraint equation

However, with two unknowns, the solution cannot be determined uniquely (so-called *aperture problem*), which is why various optical flow methods introduce additional constraints. To resolve this ambiguity, the Lucas–Kanade method [LK81] assumes a constant displacement for all pixels within a local neighborhood (resulting in a least squares problem) and, thus, estimates a sparse flow field unsuitable for per-pixel registration problems. The Horn–Schunck method [HS81] computes the displacement for each pixel but enforces a smooth flow field where neighboring pixels have a similar motion, i.e.,  $\nabla u(x, y) \approx 0$  and  $\nabla v(x, y) \approx 0$ . This results in the objective function  $E = E_d + \alpha E_s$ , with  $E_d = \iint (\mathcal{I}_x u + \mathcal{I}_y v + \mathcal{I}_t)^2 dx dy$  being the brightness constancy term,  $E_s = \iint \left( \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial u}{\partial y} \right)^2 + \left( \frac{\partial v}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 \right) dx dy$  representing smoothness, and  $\alpha$  being a scaling factor.

Farneback optical  
flow variant

The dense optical flow variant proposed by Farneback [Far03], which is used in Chapter 3 and Chapter 4, approximates the intensity distribution in the local neighborhood of each pixel with a quadratic polynomial

$$f(x, y) = a_0 x^2 + a_1 x y + a_2 y^2 + a_3 x + a_4 y + a_5, \quad (2.40)$$

which can be written in matrix form as

$$f(x, y) = \begin{pmatrix} x & y \end{pmatrix} \mathbf{A} \begin{pmatrix} x \\ y \end{pmatrix} + \mathbf{b}^\top \begin{pmatrix} x \\ y \end{pmatrix} + c, \quad (2.41)$$

where  $\mathbf{A} = \begin{pmatrix} a_0 & a_1/2 \\ a_1/2 & a_2 \end{pmatrix}$ ,  $\mathbf{b} = \begin{pmatrix} a_3 \\ a_4 \end{pmatrix}$ , and  $c = a_5$ .

Then, by assuming brightness constancy, pixel  $\mathbf{x} = (x, y)^\top$  at time  $t - 1$  corresponds with pixel  $(\mathbf{x} + \mathbf{d}) = (x + u, y + v)^\top$  at time  $t$ , and some basic calculus reveals (see [Far02]):

$$\begin{aligned} f_{t-1}(\mathbf{x}) &= f_t(\mathbf{x} + \mathbf{d}) \\ &= (\mathbf{x} + \mathbf{d})^\top \mathbf{A}_t (\mathbf{x} + \mathbf{d}) + \mathbf{b}_t^\top (\mathbf{x} + \mathbf{d}) + c_t \\ &= \underbrace{\mathbf{x}^\top \mathbf{A}_t \mathbf{x}}_{\mathbf{A}_{t-1}} + \underbrace{(\mathbf{b}_t + 2\mathbf{A}_t \mathbf{d})^\top \mathbf{x}}_{\mathbf{b}_{t-1}^\top} + \underbrace{\mathbf{d}^\top \mathbf{A}_t \mathbf{d} + \mathbf{b}_t^\top \mathbf{d} + c_t}_{c_{t-1}} \\ &= \mathbf{x}^\top \mathbf{A}_{t-1} \mathbf{x} + \mathbf{b}_{t-1}^\top \mathbf{x} + c_{t-1}, \end{aligned} \quad (2.42)$$

hence,

$$\mathbf{A}_{t-1} = \mathbf{A}_t, \quad \mathbf{b}_{t-1} = \mathbf{b}_t + 2\mathbf{A}_t\mathbf{d}, \quad \text{and} \quad c_{t-1} = \mathbf{d}^\top \mathbf{A}_t \mathbf{d} + \mathbf{b}_t^\top \mathbf{d} + c_t, \quad (2.43)$$

and, thus, the displacement  $\mathbf{d} = (u, v)^\top$  can be obtained by

$$\mathbf{d} = \frac{1}{2} \mathbf{A}_t^{-1} (\mathbf{b}_{t-1} - \mathbf{b}_t). \quad (2.44)$$

However, in practice, to increase robustness to noise,  $\mathbf{d}$  is estimated with respect to a pixel's local neighborhood using a Gaussian weighting, assuming that the flow field varies smoothly. Furthermore, to allow for larger displacements, an image pyramid is used, performing a coarse-to-fine estimation.

## 2.4 Image and Range Fusion

While the acquisition, representation, and registration of image data have been introduced, the concepts and techniques used to accumulate the data are yet to be described. Hence, as this thesis aims to bridge 2D imaging (see Section 2.4.1) and 3D model reconstruction (see Section 2.4.2), an overview of both domains is provided in the following.

### 2.4.1 Image Fusion

#### Image Compositing and Stitching

The main technical challenge in digital photo montage is to recombine images without leaving visible traces at the seams where images are composited. Several works explored strategies for visually least disruptive placement of seams [Mil75, EF01, KSE\*03, ADA\*04, LSTS04] and blending operations to obscure image differences across a seam, such as linear alpha blending, multi-band blending [BA83b], and Poisson blending [HLSH17, SUS11, PTX10, ADA\*04].

To blend two images,  $\mathcal{I}_S$  and  $\mathcal{I}_T$ , in overlapping regions, *linear alpha blending* [Linear blending](#) applies the weighted average

$$\mathcal{I}(\mathbf{x}) = \alpha \mathcal{I}_S(\mathbf{x}) + (1 - \alpha) \mathcal{I}_T(\mathbf{x}), \quad (2.45)$$

with  $\alpha \in [0, 1]$ . By varying the weight  $\alpha$  within a blending region, a smooth transition can be obtained, so that, for instance, one image gradually fades out while the other fades in.

## Multi-band blending

Burt and Adelson [BA83b] were the first to fuse images using Laplacian pyramids (see Section 2.2.1) to prevent details from being blurred. Here, *multi-band blending* performs a weighted average on each band (pyramid level) separately, within a transition zone that is scaled proportionally to a band's spatial frequency (low frequencies are blended smoothly over larger distances, while high frequencies are blended over a short range). That is, after decomposing the images  $\mathcal{I}_S$ ,  $\mathcal{I}_T$  and the mask  $\mathcal{I}_M \in [0, 1]$ , containing the weight  $\alpha$  for each pixel, into the Laplacian pyramids  $\mathcal{L}_l^S$ ,  $\mathcal{L}_l^T$  and the Gaussian pyramid  $\mathcal{G}_l^M$ , respectively, the blending is performed for each level  $l$  by

$$\mathcal{L}_l = \mathcal{G}_l^M \mathcal{L}_l^S + (1 - \mathcal{G}_l^M) \mathcal{L}_l^T. \quad (2.46)$$

Burt and Kolczynski [BK93] extend this idea by addressing the objective of combining several pre-aligned source images into a single composite image, retaining specific image regions while discarding other image portions. Related to image fusion, these early approaches assume consistent object resolution and geometric alignment.

## Poisson blending

Rather than creating a transition from one image to the other, *Poisson blending* [PGB03], or gradient-domain blending, reduces color mismatches over the entire blending region to achieve a convincing composite. This is done by creating the composite  $\mathcal{I}$  that retains the gradients of the source image ( $\mathcal{I}_S$ ) within the blending region  $\Omega$  (a subset of the full image domain in a common coordinate system), while matching the color of the target image ( $\mathcal{I}_T$ ) on the boundary  $\partial\Omega$ :

$$\begin{aligned} & \min_{\mathcal{I}} \iint_{\Omega} \|\nabla \mathcal{I}(x, y) - \nabla \mathcal{I}_S(x, y)\|^2 dx dy, \\ \text{subject to} \quad & \mathcal{I}(x, y) = \mathcal{I}_T(x, y) \text{ on } \partial\Omega, \end{aligned} \quad (2.47)$$

whose solution is given by the *Poisson equation*

$$\begin{aligned} & \nabla^2 \mathcal{I}(x, y) = \nabla^2 \mathcal{I}_S(x, y) \text{ over } \Omega, \\ \text{with} \quad & \mathcal{I}(x, y) = \mathcal{I}_T(x, y) \text{ on } \partial\Omega, \end{aligned} \quad (2.48)$$

where  $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})^\top$  is the gradient operator and  $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$  the Laplace operator.

## Panoramic photography

*Panoramic photography* is strongly related to seamless photomontage, as it attempts to combine several images into a consistent, artifact-free image. Geometric registration is facilitated via feature matching, either based on simple landmarks and image translations [Mil75] or on more complex features like SIFT to constrain homographies [BL07] (see Section 2.3.1). To compensate for (global) changes in brightness, *gain compensation* is commonly applied [BL07]

## Gain compensation

to all  $n$  images based on the gain normalized intensity error

$$E_{\text{gain}} = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \sum_{(p,q) \in \Omega_{i,j}} (g_i \mathcal{I}_i(\mathbf{p}) - g_j \mathcal{I}_j(\mathbf{q}))^2, \quad (2.49)$$

for all corresponding pixels in the overlapping region  $\Omega_{i,j}$  between images  $\mathcal{I}_i$  and  $\mathcal{I}_j$ , where  $g_i$  and  $g_j$  are the gains<sup>10</sup>. For the final image composition, blending strategies, including Poisson and multi-band blending, are used [SS97, BL07, PTX10, SUS11, HLSH17].

Kopf et al. [KUDC07] introduced a system to acquire *gigapixel images*, i.e., wide-angle images of extremely high resolution, addressing the specific challenge of capturing panoramic images consisting of billions of pixels. Their source imagery consists of geometrically uncalibrated high dynamic range (HDR) image stacks, captured using an automated camera mount and undistorted through feature matching. Overall geometric consistency is achieved via global alignment, e.g., bundle adjustment [TMHF00]; photometric consistency results from an exposure adjustment utilizing the linear intensity domain of the HDR imagery and a photometric alignment and composition technique [EUS06]. The final composition is achieved by applying a graph-cut optimization, i.e., finding optimal cuts that allow a seamless result [KSE\*03, ADA\*04]. Kazhdan and Hoppe [KH08] proposed methods for further editing gigapixel images. Their out-of-core multi-grid approach allows for gradient-domain image-editing operations involving the solution of Poisson equations that would exceed the main memory capacity in the case of gigapixel images. He et al. [HLSH17] extended the gigapixel approach towards wide-angle, high-resolution looping panoramic video synthesis.

Gigapixel images

### Image Refinement

By fusing information from a set of low-resolution observations, a *super-resolution* image may be estimated [TH84, PPK03]. Relying on the presence of *aliasing*<sup>11</sup> and an in-plane motion between images, this approach exploits sub-pixel shifts to recover the aliased information, i.e., the (desired) high-frequency content that is overlapped with low-frequency components. Hence,

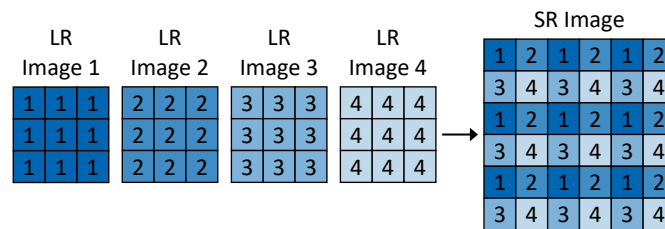
Super-resolution

Aliasing

<sup>10</sup>In practice,  $\mathcal{I}_i(\mathbf{p})$  and  $\mathcal{I}_j(\mathbf{q})$  are each approximated by the mean intensity over all overlapping pixels, and the prior term  $(1 - g_i)^2$  is added to Equation (2.49) to avoid  $g = 0$  as a solution.

<sup>11</sup>Aliasing occurs whenever the signal contains frequencies higher than half the sampling rate  $f_s$  (i.e., the *Nyquist frequency*  $f_s/2$ ). In the frequency domain, frequency components above the Nyquist frequency are folded back into the interval  $[0, f_s/2]$ , overlapping with lower frequency components of the spectrum. The visual manifestations are artifacts in the form of moiré patterns.

in case of an absence of aliasing, either (1) high(er)-frequency content is not present, making super-resolution unnecessary, or (2) a blur-filter/anti-aliasing filter removed the (desired) high-frequency content to prevent aliasing artifacts [LCA18]. While in the frequency domain, the phase shifts are used to separate the overlapping frequency components (see, e.g., [LCA18]), in the spatial domain, the sub-pixel offsets map each pixel of the low-resolution observations to a location on the high-resolution grid, as illustrated in Figure 2.13.



**Figure 2.13:** Super-resolution (right) from a set of low-resolution images, each shifted by a sub-pixel offset (left). Under ideal conditions, the sub-pixel offsets map each low-resolution pixel to an integer position on the high-resolution grid, resulting in a super-resolution image.

In contrast, methods for enriching a low-resolution image with high-resolution details from close-ups have been proposed, which will be discussed in Chapter 3.

## 2.4.2 3D Reconstruction with RGB-D Cameras

While real-time 3D reconstruction has its roots in the work by Rusinkiewicz et al. [RHHL02], enabling interactive object acquisition, the influential KinectFusion [IKH\*11, NIH\*11] system has spawned a class of algorithms for online reconstruction of high-detailed 3D models (geometry and color) using a handheld RGB-D camera. The following provides an introduction to the established pipeline that fuses a stream of RGB-D data into a progressively updated scene representation. For a comprehensive overview of various pipeline enhancements, the reader is referred to the survey paper by Zollhöfer et al. [ZSG\*18].

Depth map  
pre-processing

In the *pre-processing* stage, bilateral filtering [TM98] is commonly applied to the input depth map  $\mathcal{D}_i(\mathbf{x})$ , with  $\mathbf{x} = (x, y)^T$  and frame index  $i$ , to mitigate noise by smoothing homogeneous regions while preserving depth discontinuities:



$$\mathcal{D}_{\text{denoised}}(\mathbf{x}) = \frac{1}{W} \sum_{\mathbf{x}' \in \Omega} \mathcal{D}(\mathbf{x}') \underbrace{G_s(\|\mathbf{x} - \mathbf{x}'\|)}_{\text{spatial kernel}} \underbrace{G_r(\|\mathcal{D}(\mathbf{x}) - \mathcal{D}(\mathbf{x}')\|)}_{\text{range kernel}}, \quad \text{Bilateral filter}$$

where

$$W = \sum_{\mathbf{x}' \in \Omega} G_s(\|\mathbf{x} - \mathbf{x}'\|) G_r(\|\mathcal{D}(\mathbf{x}) - \mathcal{D}(\mathbf{x}')\|), \quad (2.50)$$

$$G_s(\mathbf{x}) = e^{-\|\mathbf{x}\|^2/2\sigma_s^2}, \quad \text{and} \quad G_r(d) = e^{-d^2/2\sigma_r^2},$$

with  $\Omega$  being the window centered in  $\mathbf{x}$ ,  $W \in \mathbb{R}$  the normalization factor,  $G_s$  the spatial Gaussian kernel, and  $G_r$  the range kernel; thus, weighting the local neighborhood depending on the Euclidean distance of pixels and their depth offset. Furthermore, in this pre-processing stage, additional attribute maps are extracted that will be needed in later pipeline stages, e.g., a vertex map  $\mathcal{V}_i \in \mathbb{R}^3$  by back-projecting each pixel (Equation (2.21)) and a normal map  $\mathcal{N}_i \in \mathbb{R}^3$  determined from  $\mathcal{V}_i$  by central differences, i.e.,  $\mathcal{N}(x, y) = (\mathcal{V}(x+1, y) - \mathcal{V}(x-1, y)) \times (\mathcal{V}(x, y+1) - \mathcal{V}(x, y-1))$ , normalized to unit length. If the depth and color frames are not pre-registered, the RGB-D registration (see Section 2.1.3) is also performed in this stage.

To globally align the current frame  $\mathcal{D}_i$  with the so-far accumulated model  $\mathcal{M}$  (frame-to-model tracking), a *camera pose estimation* is performed to compute the rigid camera transformation  $\mathbf{T}_i = [\mathbf{R}_i, \mathbf{t}_i] \in \mathbb{SE}^3$ , with 3D rotation matrix  $\mathbf{R}_i \in \mathbb{SO}^3$  and translation vector  $\mathbf{t}_i \in \mathbb{R}^3$ . As point correspondences are unknown, the iterative-closest-point (ICP) algorithm [BM92, CM92] is used, which alternates iteratively between a data association step and a minimization of the alignment error between (candidate) point pairs. Here, the relative transformation  $\mathbf{T}_{i \rightarrow (i-1)}$ , with  $\mathbf{T}_i = \mathbf{T}_{i-1} \mathbf{T}_{i \rightarrow (i-1)}$ , between the vertex map  $\mathcal{V}_i$  of the current frame and the model's vertex map  $\mathcal{V}_{\mathcal{M}}$  (a *local surface reconstruction* of  $\mathcal{M}$  as seen from the previous frame  $i-1$ ) is estimated based on the point-to-plane error metric

$$E_{\text{ICP}}(\mathbf{T}_{i \rightarrow (i-1)}) = \sum_{(\mathbf{p}, \mathbf{q}) \in \mathcal{R}} \langle \mathbf{T}_{i \rightarrow (i-1)} \mathcal{V}_i(\mathbf{q}) - \mathcal{V}_{\mathcal{M}}(\mathbf{p}), \mathcal{N}_{\mathcal{M}}(\mathbf{p}) \rangle^2, \quad (2.51)$$

i.e., the distance between a point and the tangent plane at its corresponding model point. The correspondence set  $\mathcal{R} = \{(\mathbf{p}, \mathbf{q})\}$  is determined via projective data association, i.e.,  $\mathcal{V}_i(\mathbf{q})$  and  $\mathcal{V}_{\mathcal{M}}(\mathbf{p})$  projected onto the same image coordinates by  $\mathbf{p} = \pi(\mathbf{K} \mathbf{T}_{i \rightarrow (i-1)} \mathcal{V}_i(\mathbf{q}))$ . Advanced error metrics and pairing strategies have been proposed, such as by introducing normals [SG15], contours [ZK15], curvature [LKS\*17], or color [PZK17]; see also [RL01] for an analysis of various ICP design choices.

After a successful registration of the depth map  $\mathcal{D}_i$ , the incremental update of the scene representation is done by a *depth map fusion* of the new observa-

Camera pose estimation

ICP algorithm

Depth map fusion

tion with the model  $\mathcal{M}$  based on [CL96]. That is, an input point's position is blended with its corresponding model point by the cumulative average

$$\mathcal{V}_{\mathcal{M}} \leftarrow \frac{w_{\mathcal{M}}\mathcal{V}_{\mathcal{M}} + \mathcal{V}_i}{w_{\mathcal{M}} + 1}, \quad w_{\mathcal{M}} \leftarrow w_{\mathcal{M}} + 1, \quad (2.52)$$

with  $w_{\mathcal{M}}$  being a weight stored per model point, which is incremented with each new observation.

3D scene  
representations

Mainly, two types of scene representations are used to accumulate the incoming depth observations: volumetric or point-based representations. Volumetric fusion [CL96, NIH\*11] uses the TSDF (truncated signed distance function) over a uniform grid of voxels that stores the closest distance to the surface (truncated to a maximum distance), with interior and exterior voxels encoded by negative or positive distances, respectively, and the zero-crossing defining the surface itself. As this requires conversion between input points and the implicit voxel-based representation, point-based fusion [KLL\*13] uses an unordered set of oriented points (surfels), each defined by a 3D position, normal, and radius.

In order to fuse the color information of each input frame, RGB colors are accumulated per voxel/surfel analogously to Equation (2.52). However, as this inhibits the reconstruction of high-fidelity textures, methods for texture optimization have recently been proposed, which will be discussed in Chapter 4.

# 3

## Progressive Refinement Imaging for Quasi-Planar Scenes

*This chapter presents a novel technique for progressive online integration of uncalibrated image sequences with substantial geometric and/or photometric discrepancies into a single, geometrically and photometrically consistent image. It can handle large sets of images, acquired from a nearly planar or far-distant scene at variable object-space resolutions and under varying local or global illumination conditions. It allows for efficient user guidance as its progressive nature provides a valid and consistent reconstruction at any moment during the online refinement process.*

*The proposed approach avoids global optimization techniques, as commonly used in the field of image refinement, and progressively incorporates new imagery into a dynamically extendable and memory-efficient Laplacian pyramid. The image registration process includes a coarse homography and a local refinement stage using optical flow. Photometric consistency is achieved by retaining the photometric intensities given in a reference image while it is being refined. Globally blurred imagery and local geometric inconsistencies due to, e.g., dynamic objects are detected and removed prior to image fusion.*

*The quality and robustness of the proposed approach are demonstrated using several image and video sequences, including handheld acquisition with mobile phones and zooming sequences with consumer cameras.*

*The method described in this chapter has been published [KWK20] in Computer Graphics Forum, Vol. 39(1), 2020.*

---

**T**he visual appearance of real-world objects and scenarios spans multiple scales, and yet, despite an impressive rise in sensor resolution, photographic imaging hardware is hardly able to simultaneously capture visual details across all of these scales. Several algorithmic approaches have been proposed to overcome the resolution limits of digital imaging, creating higher-resolution images by fusing information from multiple observations.

Super-resolution techniques obtain a high-resolution image from multiple low-resolution images [PPK03], exploiting sub-pixel shifts between the individual images. These approaches commonly require a large mutual overlap of the observations, a nearly in-plane motion between images, and strongly rely

Super-resolution

on sufficient aliasing to be present in the imager (see Section 2.4.1). Not the least due to these many constraints, practical applications are limited to specialized domains where the imaging process meets hard physical limits, such as satellite imaging, microscopy, or computed tomography [NM14]. Moreover, the achievable increase in resolution is limited, typically well below an order of magnitude.

Alternatively, computational methods for image recombination and fusion have been developed that address the acquisition of scenes or objects that cannot be captured with a single photograph (see Section 2.4.1). Panoramic photography extends an image laterally, by creating a wide-angle mosaic from a set of images with narrower field of view and small overlapping regions. Both alignment and stitching are usually formulated as global optimization problems, commonly further constrained by assuming that the camera's focal point is fixed, i.e., all images share the same viewpoint. The achievable panorama size is generally unlimited, which gave rise to the popular concept of gigapixel images [KUDC07]; however, the object-space resolution is fixed and defined by the resolution and focal length of the camera used to capture the individual panorama tiles. In contrast, a low-resolution reference image that completely covers a scene of interest can be enriched with high-resolution details from close-ups [EESM10]. The work presented in this chapter takes a similar approach to increase resolution locally where more detail is required.

All methods mentioned above have in common that they process images in batch mode, after capture. Inspired by progressive acquisition approaches in 3D scene reconstruction [ZSG\*18], the proposed method deliberately aims at a progressive framework that allows continuous addition of observations without the need for repeated global optimization. As we will see, this results in a lightweight and robust image acquisition pipeline that enables (1) reconstruction of *variable-resolution* images from different sources, such as hand-held video streams, or mixed-field-of-view images from different viewpoints, without requiring calibration, pre-alignment, external tracking, lighting adjustment, or other intervention; (2) online user guidance for casual capture and dynamic refinement, even in fleeting situations, due to its immediate availability of intermediate results; and (3) fusing hundreds of images by continuously eliminating redundancy, thus taking the burden of efficiency-conscious view planning from the user.

At the core of the proposed method is an adaptive and expandable Laplacian pyramid representation that is used to accumulate observations. Image pyramids are a popular multi-scale representation known for their ability to edit or recombine details from multiple image sources while consistently blending or preserving coarse-scale characteristics (see Section 2.2.1 and Section 2.4.1).

Panoramic  
photography

Image refinement

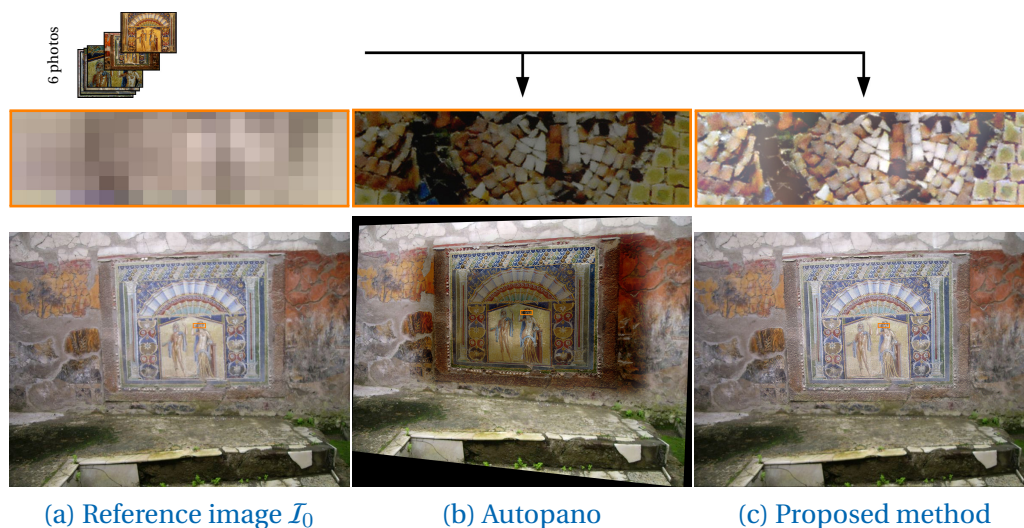
Progressive  
framework

Adaptive multi-scale  
representation

For maximum flexibility, the proposed pyramid representation does not have a fixed size but can grow both laterally and vertically (see Section 3.4.2), due to its sparse, tile-based design and the ability to append levels as needed. With its progressive nature and low costs of decoding, this representation provides a valid and consistent adaptive-resolution reconstruction at any moment during the progressive imaging process.

Similar to conventional panoramic imaging, the proposed method assumes the absence of strong parallax in the input images. However, it allows for general camera viewpoints spanning a wide range of resolutions and imagery with strongly varying lens characteristics. The described system features exceptional robustness against geometric and photometric inconsistencies. A coarse-to-fine alignment strategy compensates for lens distortions or small amounts of parallax by employing optical flow [Far03]. Details from the aligned image data are then selectively merged into the pyramidal reconstruction in a way that removes low-frequency photometric artifacts, such as lens vignetting, changes in ambient lighting conditions, or varying auto exposure and color balance.

In summary, this work proposes a simple, yet effective approach to progressively integrate an open set of images into a single geometrically and photo-



**Figure 3.1:** A sample result of the *progressive refinement imaging* pipeline applied to the *House of Neptune and Amphitrite mosaic* data set comprising one reference image  $I_0$  that is refined using six additional images captured with six different cameras over the period of 10 years. Compared to prior work, the proposed method successfully generates photometrically and geometrically consistent results in an online and memory-efficient fashion without global optimization.

metrically consistent image of a near-planar scenery. Unique strengths and contributions are

- the ability to robustly process uncalibrated, potentially unsharp, geometrically and photometrically inconsistent images at different levels of object resolution and from different viewpoints,
- the continuous local resolution adjustment to meet the resolution and extent of the incoming images, and
- the scalability into gigapixel range while maintaining near-constant update times upon incoming images.

### 3.1 Image Refinement by Variable-Resolution Image Compositing

Conceptually, panoramic stitching [SS97] approaches combine several images into a single photograph by solving the problem of image registration, i.e., *geometric consistency*, and image recombination, i.e., *photometric consistency*. However, very specific conditions usually have to be met, and applying these kinds of methods to imagery with highly variable object-space resolution and significant geometric and photometric discrepancies usually leads to failure. This is mainly due to enforcing a panoramic mosaicing scenario with constant resolution in the object domain, resulting in unsuccessful matching of the input frames or unsuccessful integration of variable-resolution images (see Table 3.2). However, feature-based panorama stitching approaches for unordered data sets using, for instance, SIFT feature matching and multi-band blending [BL07] can solve the challenging data characteristics as demonstrated in methods like AutoStitch [Bro18] and AutoPano [Kol18], but require global post-optimization for aligning the imagery and reducing texture inconsistencies.

A similar goal to the work presented in this chapter is pursued by Eismann et al.'s Photo Zoom [EESM10], which automatically constructs a high-resolution image from an unordered set of zoomed-in photos but, again, requires global, post-capture processing. Furthermore, they (1) tackle color inconsistencies using a recursive gradient domain fusion approach that cannot handle strong local variations such as reflections, (2) only apply homographies to register images and mask out regions with inconsistent content, (3) expect all input images to be focused, and (4) only fuse a comparable small number of images. Licorish et al. [LFS21], published after the work presented here, address adaptive compositing of different-resolution images by computing

variable-resolution seams. However, the proposed offline method assumes pre-registered images at different resolutions captured with a single camera with optical zoom and within a short period of time, thus mitigating photometric inconsistencies.

The research presented in this chapter advances the state of the art by proposing a method that enables the compositing of *variable-resolution* input imagery under inconsistent (local and global) illumination conditions, yet in an interactive *progressive* fashion.

## 3.2 Pipeline Overview

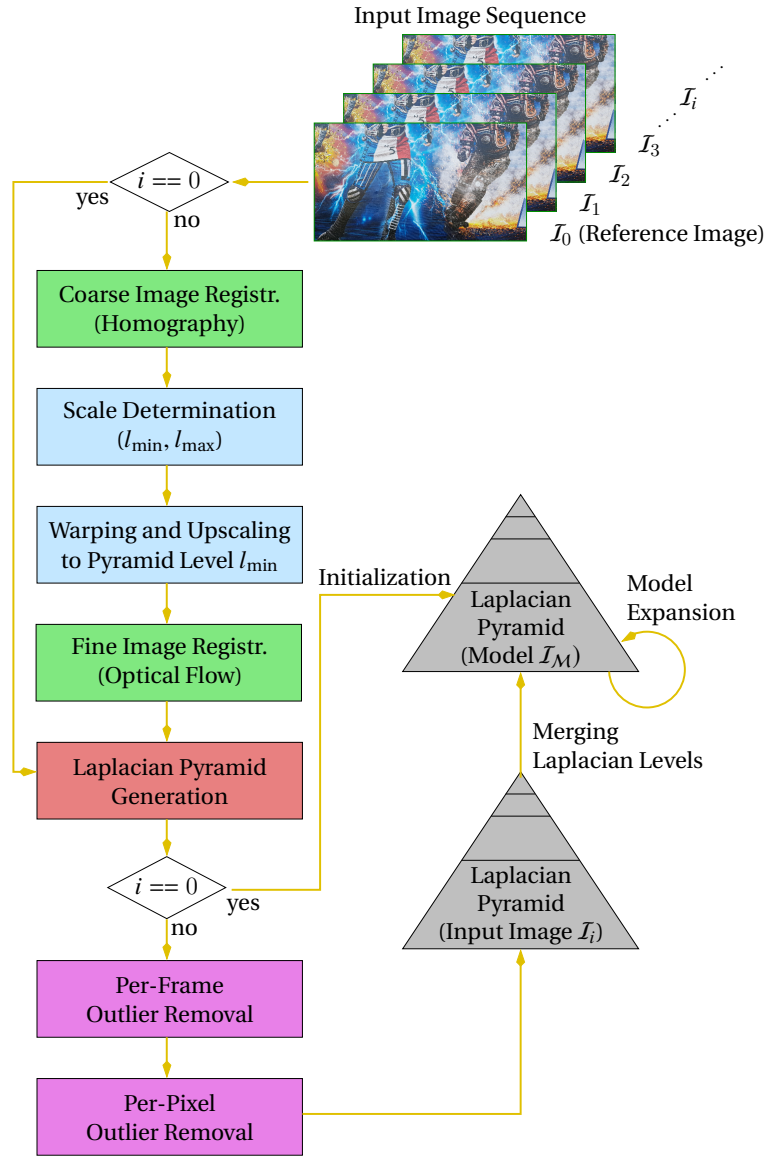
The proposed refinement pipeline comprises several processing stages, as depicted in Figure 3.2. The input to this pipeline is an open set of images  $\mathcal{I}_i \in \mathbb{R}^3$  for frame indices  $i$ , comprising RGB intensities in the sRGB color space. The first input image  $\mathcal{I}_0$  fed into the pipeline is expected to be a *reference image*, covering the region of interest for all following input images  $\mathcal{I}_i$ ,  $i > 0$ . Within this region initialized by  $\mathcal{I}_0$ , a refined and geometrically as well as photometrically consistent image representation is produced by progressively fusing the incoming observations. In the following, this representation is called the *model*  $\mathcal{I}_M$ . The overall assumption here is that by zooming in or moving closer to the scene, subsequent input images provide further information in terms of finer details or new lateral image regions.

The main stages of the proposed pipeline can be summarized as follows (see Table 3.1 for a complete list of conventions used):

[Pipeline overview](#)

*Image Registration:* While the viewing direction of the reference image  $\mathcal{I}_0$  defines the default view for the refinement process, further observations  $\mathcal{I}_i$ ,  $i > 0$ , can be acquired from different positions and viewing angles. To match the model’s pixel grid, the observation of the current pipeline iteration,  $\mathcal{I}_{\text{curr}} = \mathcal{I}_i$ , is registered with the so-far accumulated model  $\mathcal{I}_M$ . This is done by first aligning the observation globally using a homography estimated with the help of local features. Afterward, the registration is locally fine-corrected based on an estimated flow field, resulting in the warped observation  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$  (see Section 3.4.1).

*Laplacian Pyramid Generation:* In this pipeline stage, the registered observation  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$  is decomposed into Laplacian pyramid levels  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l$ , with level indices  $l$ , by differences of low-pass filtered and downsampled versions of  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$  (see Section 2.2.1). Thus, after the decomposition, each level of  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l$  contains the frequencies of a specific band. Depending



**Figure 3.2:** The proposed *progressive refinement imaging* pipeline for planar scenes.

on the observation's viewing direction and position, the Laplacian levels  $I_{\text{curr} \rightarrow \mathcal{M}}^l$  may contribute to the corresponding model levels  $I_M^l$  by adding new information in the following ways. They can provide (1) higher frequency band(s) not present in the model so far, (2) high frequencies already present but with less precision, and/or (3) new spatial coverage not observed so far (see Section 3.4.2).

**Outlier Removal:** As the warped observation  $I_{\text{curr} \rightarrow \mathcal{M}}$  may have different deficiencies, a two-stage outlier removal is conducted. First, a global reliabil-



**Table 3.1:** List of conventions.

$\mathcal{I}_i$	$i$ th input image, where $\mathcal{I}_0$ is the reference image and $\mathcal{I}_i$ , $i > 0$ , an observation	List of conventions
$\mathcal{I}_{\text{curr}}$	Input image (observation) of the current iteration	
$\mathcal{I}_{\mathcal{M}}$	Model (refined reference image), consisting of pyramid levels $\mathcal{I}_{\mathcal{M}}^l$ with level indices $l \in [l_{\min}^{\mathcal{M}}, l_{\max}^{\mathcal{M}}]$	
$\mathcal{F}_{\text{curr}}, \mathcal{F}_{\mathcal{M}}$	Local feature set in $\mathcal{I}_{\text{curr}}$ and $\mathcal{I}_{\mathcal{M}}$	
$\mathcal{H}_{\text{curr} \rightarrow \mathcal{M}}$	Homography warping $\mathcal{I}_{\text{curr}}$ to $\mathcal{I}_{\mathcal{M}}$	
$\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$	$\mathcal{I}_{\text{curr}}$ warped to $\mathcal{I}_{\mathcal{M}}$ 's image space	
$\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l$	$\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$ decomposed into Laplacian levels with level indices $l \in [l_{\min}^{\text{curr} \rightarrow \mathcal{M}}, l_{\max}^{\text{curr} \rightarrow \mathcal{M}}]$	
$T_{\text{curr} \rightarrow \mathcal{M}}^{(p,q),l}, T_{\mathcal{M}}^{(p,q),l}$	$\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l$ and $\mathcal{I}_{\mathcal{M}}^l$ split into tiles with 2D array position $(p, q)$	
$l_{\min}, l_{\max}$	Finest and coarsest corresponding level index of the warped observation within model pyramid $\mathcal{I}_{\mathcal{M}}$	
$\mathcal{L}_{\text{curr} \rightarrow \mathcal{M}}$	Level-of-refinement map of $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$ storing real-valued level indices per pixel with respect to the model pyramid	
$\mathcal{L}_{\text{curr} \rightarrow \mathcal{M}}^l, \mathcal{L}_{\mathcal{M}}^l$	Pyramidal representations of the level of refinement, representing confidences for $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l$ and $\mathcal{I}_{\mathcal{M}}^l$	

ity check is applied to ensure that  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$  provides valuable frequency information consistent with the so-far accumulated model  $\mathcal{I}_{\mathcal{M}}$ , or if it is out of focus, e.g., due to an incorrect autofocus or motion blur. On the second outlier removal stage, a per-pixel error on the Laplacian levels is computed to recognize local registration errors due to, e.g., inaccuracies in the optical flow estimation (Section 3.4.3).

*Model Expansion:* The accumulation of observations into the model is not restricted in terms of scale, resolution, or spatial coverage in the object domain. The proposed model representation is an adaptive Laplacian pyramid that can be expanded both laterally and vertically to incorporate novel information. This is facilitated through a sparse, tile-based representation of pyramid levels in which tiles are allocated and levels are added on demand during progressive enhancement (see Sections 3.3 and 3.4.2).

*Merging of Laplacian Levels:* At the core of the proposed technique lies the merging of specific Laplacian levels  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l$  and  $\mathcal{I}_{\mathcal{M}}^l$  of the current observation and the model, respectively. Depending on the object-space resolution and/or lateral information provided by  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l$ , the obser-

vation’s potential to refine the model is estimated by determining per-pixel level-of-refinement values  $\mathcal{L}_{\text{curr} \rightarrow \mathcal{M}}^l(x, y)$ . A comparison with the model’s accumulated level of refinement  $\mathcal{L}_{\mathcal{M}}^l(x, y)$  is then used to decide which pixels are capable of refining the model and how the observation and the model pixel values of the Laplacian levels are combined (see Section 3.4.4). Note that the top Gaussian levels of the model and the observation pyramid are never merged, but only Laplacian levels, thus retaining photometric consistency.

*Refinement Guidance:* Optionally, a visualization can be rendered to steer the user towards image areas that need further refinement according to his or her needs and interests (see Section 3.4.5).

### 3.3 Adaptive Model Representation

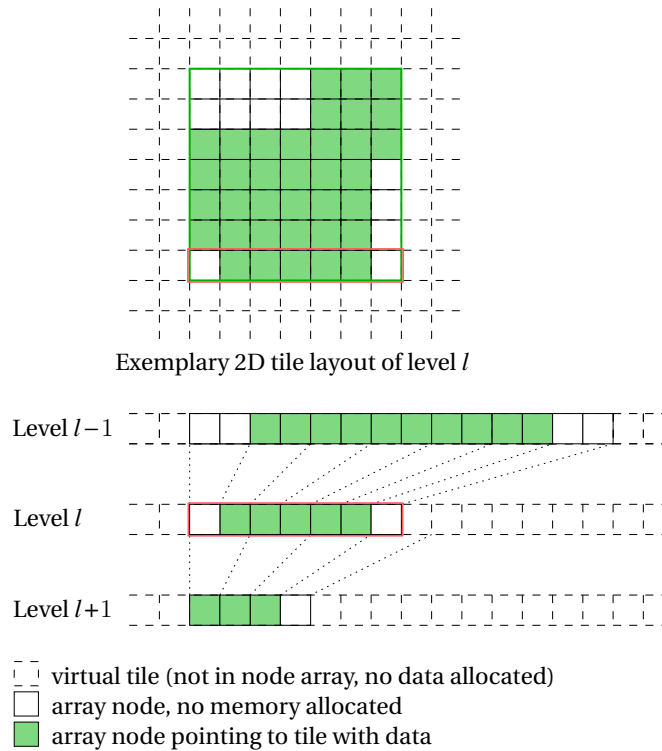
The preliminary goal is to progressively refine a given model  $\mathcal{I}_{\mathcal{M}}$  with new input images  $\mathcal{I}_i$  (observations) that can be taken at different scales or resolutions in the object domain and cover potentially different regions. To enable this *variable-resolution* refinement, the model is represented by an adaptive Laplacian pyramid instead of a flat image representation. This adaptive Laplacian pyramid efficiently stores the model  $\mathcal{I}_{\mathcal{M}}$  by means of localized detail information at different resolutions. Assuming that two images (the observation and the model in this case) are properly registered, Laplacian pyramids offer the advantage of directly comparing and manipulating detail information on corresponding resolution levels without the computational burden of an explicit frequency analysis; see Burt et al. [BA83b] and Chapter 2 for further technical details.

#### 3.3.1 Initialization

Laplacian pyramid  
representation

The model  $\mathcal{I}_{\mathcal{M}}$  is a multi-scale representation consisting of Laplacian pyramid levels  $\mathcal{I}_{\mathcal{M}}^l$ , where the level index  $l \in [l_{\min}^{\mathcal{M}}, l_{\max}^{\mathcal{M}}]$  decreases with finer resolution and, thus,  $l_{\min}^{\mathcal{M}}$  and  $l_{\max}^{\mathcal{M}}$  refer to its finest and coarsest pyramid level for which data has been accumulated so far.  $\mathcal{I}_{\mathcal{M}}$  is initialized by the reference image  $\mathcal{I}_0$  and serves as a reference view of the scene. Over time, new, finer Laplacian levels  $\mathcal{I}_{\mathcal{M}}^{l < 0}$  are appended to the bottom of the pyramid, refining the initial reference image as novel details are added from subsequent observations. Hence, all pyramid levels  $\mathcal{I}_{\mathcal{M}}^l$  refer to a specific scale factor with respect to  $\mathcal{I}_0$ , i.e., level  $\mathcal{I}_{\mathcal{M}}^{l=0}$  refers to the full resolution of  $\mathcal{I}_0$ , whereas levels  $\mathcal{I}_{\mathcal{M}}^{l < 0}$  and  $\mathcal{I}_{\mathcal{M}}^{l > 0}$  contain finer and coarser image resolutions, respectively (see Figure 3.3).

From level  $\mathcal{I}_M^l$  to  $\mathcal{I}_M^{l+1}$ , the resolution decreases by one octave, i.e., if level  $\mathcal{I}_M^{l=0}$  is defined as sampling distance 1, level  $\mathcal{I}_M^l$  has the sampling distance  $2^l$ . All further incoming observations that are potentially acquired from different positions under different view directions are warped appropriately to match this reference view.



**Figure 3.3:** Adaptive Laplacian model pyramid. Top: on each pyramid level, a virtually infinite tile array is set up. The nodes in the array form the bounding box (green box) of potential tiles (white squares) and, if required, allocated tiles (green squares). Bottom: corresponding tiles related to the tile row marked in orange on different pyramid levels (as 1D layout), where two neighboring tiles are downsampled to a single tile.

### 3.3.2 Adaptivity

As the model has to be dynamically expanded to represent so far unobserved content, i.e., coarser or finer Laplacian levels or new lateral regions, a tile-based variant of the Laplacian pyramid is used. As storing a complete Laplacian pyramid would be extremely memory inefficient, a simple regular grid per pyramid level is set up, with a 2D node array covering the bounding box of the tiles. While tiles with data are stored in an unordered list, the 2D node array stores the actual layout of the tiles, forming a pyramid level of the model  $\mathcal{I}_M$ .

Model adaptivity

A node points either to the allocated data of its tile or stores  $-1$  if no memory has been allocated so far. This 2D node array can be extended in the lateral direction and new levels can easily be added to represent new resolution levels (see Figure 3.3). If required, new tiles get allocated and assigned to the virtual nodes on demand. In all experiments, a tile size of  $512 \times 512$  px is used.

### 3.3.3 Level-of-Refinement Maps

The confidence of the accumulated model pixels  $\mathcal{I}_{\mathcal{M}}^l(x, y)$  is represented by storing per-pixel level-of-refinement values  $\mathcal{L}_{\mathcal{M}}^l(x, y)$  for each Laplacian level. Together with the level of refinement  $\mathcal{L}_{\text{curr} \rightarrow \mathcal{M}}^l(x, y)$  computed for the current observation  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l$ , these values guide the merging process (see Section 3.4.4 for more details).

## 3.4 Progressive Refinement Imaging

The proposed refinement imaging approach is based on an input sequence of color images  $\mathcal{I}_i$  that are progressively fused into the model  $\mathcal{I}_{\mathcal{M}}$  (see Section 3.3). With each new input image, the *current observation*  $\mathcal{I}_{\text{curr}} = \mathcal{I}_i$  is passed to the following pipeline stages.

### 3.4.1 Image Registration

Homography estimation

As the current observation  $\mathcal{I}_{\text{curr}}$  is expected to be captured with a different focal length and/or from a different camera pose than the reference view of the model  $\mathcal{I}_{\mathcal{M}}$ , the homography between  $\mathcal{I}_{\text{curr}}$  and  $\mathcal{I}_{\mathcal{M}}$  is estimated first. Therefore, a set of local features  $\mathcal{F}_{\text{curr}}$  is detected in  $\mathcal{I}_{\text{curr}}$  at multiple scales (using four octaves with four scales in each octave) and matched to the so-far accumulated model features  $\mathcal{F}_{\mathcal{M}}$ , detected in previous observations. Here, each set  $\mathcal{F} = \{(x_k, y_k, f_k) \mid k = 1, \dots, n\}$  of  $n$  detected features is defined by its position  $x_k, y_k$  and its scale-invariant descriptor  $f_k$  (see Section 2.3.1). To allow for fast and robust detection, speeded-up robust features (SURF) [BTVG06] are used, while a RANSAC matching [FB81] is applied to the feature sets  $\mathcal{F}_{\text{curr}}$  and  $\mathcal{F}_{\mathcal{M}}$  to estimate the homography  $\mathcal{H}_{\text{curr} \rightarrow \mathcal{M}}$  (see Section 2.3.1). As some spatial coherence between consecutive input images can be assumed, which is especially true in the case of video sequences, the homography of the previous frame is used as initialization. Furthermore, to accumulate features for later usage without having to reconstruct the model pyramid, all features  $\mathcal{F}_{\mathcal{M}}$  positioned within the currently observed area are replaced with new features  $\mathcal{F}_{\text{curr} \rightarrow \mathcal{M}}$  (if

the observation passes the per-frame outlier check in Section 3.4.3). Since all positions  $(x_k, y_k)$  of  $\mathcal{F}_M$  are related to the finest model level  $l_{\min}^M$ , the positions of  $\mathcal{F}_{\text{curr} \rightarrow M}$  are transformed accordingly. Note that this re-positioning is also performed on  $\mathcal{F}_M$  after the model is expanded to finer levels.

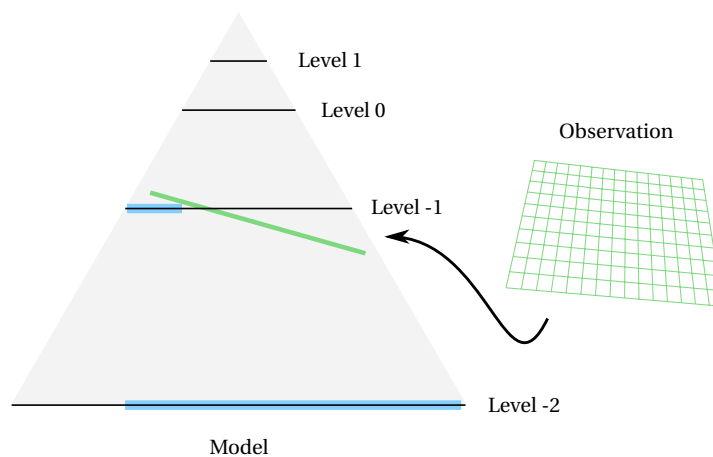
### Homography-Based Image Warping

Using the homography  $\mathcal{H}_{\text{curr} \rightarrow M}$ , the observation  $\mathcal{I}_{\text{curr}}$  is now positioned laterally and with respect to its object-space resolution within the model (see Figure 3.4). This yields the minimum and maximum level indices  $l_{\min}$  and  $l_{\max}$  in the model pyramid that bound the scale of  $\mathcal{I}_{\text{curr}}$ . As an information loss due to a downscaling should be avoided, the observation is upsampled to the finest corresponding level  $l_{\min}$  (e.g., level  $l = -2$  in Figure 3.4). However, to maintain the original level positioning, a per-pixel *level-of-refinement map*  $\mathcal{L}_{\text{curr} \rightarrow M}(x, y)$  is determined by storing the real-valued level indices (see also Section 3.4.4). The warping of  $\mathcal{I}_{\text{curr}}$  to  $\mathcal{I}_M$ 's image space is then performed by applying a resampling of  $\mathcal{I}_{\text{curr}}$  according to  $\mathcal{H}_{\text{curr} \rightarrow M}$  using bi-linear interpolation, resulting in  $\mathcal{I}_{\text{curr} \rightarrow M}$ .

Model  
correspondence

### Local Fine-Correction Using Optical Flow

As uncalibrated observations are taken as input, mismatches can be expected, especially in border and corner regions, if only the homography is applied. To



**Figure 3.4:** An observation is positioned within the adaptive Laplacian model pyramid, contributing high frequencies to the Laplacian model levels  $l = -1$  and  $l = -2$  (for the areas highlighted in blue). To avoid downsampling, the observation is warped to the finest corresponding level ( $l = -2$ ) to match its pixel grid.

Dense optical flow

reduce this mismatch to a minimum, the registration is locally fine-corrected in a second stage. This is achieved by computing the displacement for each pixel of  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$  so that the photometric consistency between  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$  and  $\mathcal{I}_{\mathcal{M}}$  is as high as possible. As a dense optical flow [HS81, LK81] estimates the pixel-wise motion between two frames, the resulting 2D flow field contains the required displacement vectors. Therefore, a backward optical flow is computed between grayscale variants of level  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$  and  $\mathcal{I}_{\mathcal{M}}^{\text{comp}}$ , where  $\mathcal{I}_{\mathcal{M}}^{\text{comp}}$  is produced by re-composing the Laplacian model pyramid for the currently observed area. In order to reduce computing resources, the optical flow is estimated at the scale of the finest corresponding level for which model data exists, i.e., level  $l = \max(l_{\min}, l_{\min}^{\mathcal{M}})$ . After potentially upscaling the flow field to the resolution of level  $l_{\min}$ , the observation  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$  is resampled accordingly. For the computation of the dense optical flow, Farneback's optical flow variant [Far03] is used.

### 3.4.2 Decomposing the Observation

Let  $\mathcal{I}_{\mathcal{M}}^l$ , with level indices  $l \in [l_{\min}^{\mathcal{M}}, l_{\max}^{\mathcal{M}}]$ , be the Laplacian pyramid of the model, where  $l_{\min}^{\mathcal{M}}$  and  $l_{\max}^{\mathcal{M}}$  denote its finest and coarsest pyramid level for which image data has been accumulated so far. Furthermore, model tiles  $T_{\mathcal{M}}^{(p,q),l}$  have been allocated, where  $(p, q)$  is the tile's position in the 2D tile array.

Laplacian decomposition

In order to establish correspondence with the model, the warped observation  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$  is decomposed into corresponding Laplacian pyramid levels  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l$ , with  $l \in [l_{\min}^{\text{curr} \rightarrow \mathcal{M}}, l_{\max}^{\text{curr} \rightarrow \mathcal{M}}]$ , where  $l_{\min}^{\text{curr} \rightarrow \mathcal{M}} := l_{\min}$  refers to the observation's finest corresponding level and  $l_{\max}^{\text{curr} \rightarrow \mathcal{M}} := \max(l_{\max}, l_{\max}^{\mathcal{M}})$  to the coarsest pyramid level observed so far. Afterward, each level is split into corresponding observation tiles  $T_{\text{curr} \rightarrow \mathcal{M}}^{(p,q),l}$ .

When capturing the scene from different positions, an observation can contribute content for merging into the model considering three cases:

Moving closer

**Contributing finer image information** The new observation shows the scene captured from a closer distance, e.g., after moving the camera toward the scene or zooming in. If the currently observed level ( $l_{\min}$ ) is beyond the level boundaries of  $\mathcal{I}_{\mathcal{M}}^l$ , i.e.,  $l_{\min}^{\text{curr} \rightarrow \mathcal{M}} < l_{\min}^{\mathcal{M}}$ , the model will be expanded by appending a new level of unallocated tiles to the bottom of the pyramid. This allows all observation tiles of the new frequency band(s), i.e.,  $T_{\text{curr} \rightarrow \mathcal{M}}^{(p,q),l}$ , with  $l < l_{\min}^{\mathcal{M}}$ , to be added to the model pyramid. However, an observation can also contribute finer information to already existing model levels as long as the real-valued

level index  $\mathcal{L}_{\text{curr} \rightarrow \mathcal{M}}(x, y)$  implies superior confidence over the so-far accumulated model data (see Section 3.4.4). In this case, existing model data will be refined by merging the model and the observation.

**Contributing new scene areas** Moving the camera laterally provides new areas outside the current image boundaries, which allows more of the scene to be included in the reconstruction. In this case, all observation levels up to the (top) Gaussian level  $l_{\text{max}}^{\text{curr} \rightarrow \mathcal{M}}$  will be used for incorporation into the model. Therefore, areas not yet present in the model will be added; already observed pixels will be merged (see Section 3.4.4). Note that in this situation, photometric inconsistencies may occur on the Gaussian level of the model pyramid outside of the region defined by the reference image  $\mathcal{I}_0$ . Lateral movement

**Contributing coarser image information** Similar to the prior case, moving the camera further away or zooming out of the reference image reveals new regions beyond the current boundaries. However, in this scenario, the observation's resolution is lower than the existing model data, i.e.,  $l_{\text{max}}^{\text{curr} \rightarrow \mathcal{M}} > l_{\text{max}}^{\mathcal{M}}$ . Consequently, the observation does not provide information to refine the model but only to extend it laterally. Thus, novel areas will be added to the model, but no existing model data will be merged. To incorporate data on coarser pyramid levels, the model's Laplacian pyramid is expanded to the same level as the observation, i.e., to  $l_{\text{max}}^{\text{curr} \rightarrow \mathcal{M}}$ , by decomposing the (top) Gaussian level  $l_{\text{max}}^{\mathcal{M}}$  into further Laplacian levels. Again, as in the prior case, photometric inconsistencies may occur in areas not covered by the reference image  $\mathcal{I}_0$ . Moving further away

### 3.4.3 Outlier Removal

Before fusing the Laplacian levels  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l$  of the current observation into the model pyramid, an outlier removal is applied in a per-frame and a per-pixel stage. Here, outlier refers to image details of the observation that are inconsistent with the so-far accumulated model  $\mathcal{I}_{\mathcal{M}}$  and, thus, should not be merged into the model. The main reasons for global inconsistencies are out-of-focus or motion-blurred images that should be rejected completely and local inconsistencies due to inaccurate flow estimations or dynamic scene parts (see Section 3.4.1).

#### Per-Frame Outlier Removal

A check for global consistency is performed by comparing the warped observation ( $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l$ ) and the model ( $\mathcal{I}_{\mathcal{M}}^l$ ) on the finest Laplacian level that is occupied Global outlier removal

by image data in both pyramids, i.e., on level  $l = \max(l_{\min}^{\text{curr} \rightarrow \mathcal{M}}, l_{\min}^{\mathcal{M}})$ . Here, a simple rule is applied, assuming that the novel observation contains at least as many fine details as the model. Therefore, the standard deviation of  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l$  and  $\mathcal{I}_{\mathcal{M}}^l$  is computed for the currently observed area, i.e.,  $\sigma_{\text{curr} \rightarrow \mathcal{M}}$  and  $\sigma_{\mathcal{M}}$ . If the standard deviation of the observed Laplacian level is smaller than the model value, i.e.,  $\sigma_{\text{curr} \rightarrow \mathcal{M}} < \sigma_{\mathcal{M}}$ , it can be concluded that the observation does not provide any new image details, and  $\mathcal{I}_{\text{curr}}$  is dropped.

### Per-Pixel Outlier Removal

If the observation has passed the per-frame outlier check, the next stage computes a per-pixel matching error that accounts for imperfect local warps due to flow estimation insufficiencies or dynamic scene parts. As a local error metric, the per-pixel error

Local outlier removal

$$E(x, y) = \sum_{l \in [l_{\min}^{\text{OR}}, l_{\max}^{\text{OR}})} \frac{|\mathcal{I}_{\mathcal{M}}^l(x, y) - \mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l(x, y)|}{\min(|\mathcal{I}_{\mathcal{M}}^l(x, y)|, |\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l(x, y)|)}, \quad (3.1)$$

is computed on pyramid levels  $l \in [l_{\min}^{\text{OR}}, l_{\max}^{\text{OR}})$ , with  $l_{\min}^{\text{OR}} := \max(l_{\min}^{\text{curr} \rightarrow \mathcal{M}}, l_{\min}^{\mathcal{M}})$  and  $l_{\max}^{\text{OR}} := l_{\max}^{\text{curr} \rightarrow \mathcal{M}}$  being the finest and coarsest Laplacian levels for which image data at  $(x, y)$  exist in both pyramids. Note that the top Gaussian level  $l_{\max}^{\text{OR}}$  is excluded from the comparison due to its susceptibility to false positives if low-frequency photometric inconsistencies (e.g., local illumination changes) exist between  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$  and  $\mathcal{I}_{\mathcal{M}}$ . Moreover, to reduce the effect of misclassifying novel incoming details as outliers, high-frequency levels that are only present in  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l$  are not included, i.e., if  $l_{\min}^{\text{curr} \rightarrow \mathcal{M}} < l_{\min}^{\mathcal{M}}$ .

The idea behind the metric given in Equation (3.1) is that the model contains consistent detail information across all Laplacian levels. Thus, the error will become large if the observation adds specifically high values in areas where the model contains only very small values, or vice versa. This is a clear indication of a local geometric inconsistency in the observation. For reasons of noise removal and filling in gaps, the resulting mask is then post-processed by a morphological opening followed by a closing. For these operations, a disk-shaped structuring element is used with radius  $r = 3$  px and  $r = 4$  px, respectively. If the observation contributes new image regions, and thus, the model does not contain data on any level, the novel content is always added.

In all experiments, observation pixels are discarded if the error exceeds  $E(x, y) > 10$  in the case of low geometric distortions and  $E(x, y) > 1$  in the case of strong geometric distortions, e.g., for the data sets *Moving cars* in Figure 3.10 and *Streetart fisheye* in Figure 3.11.



### 3.4.4 Merging of the Model and Observation

In case the observation's pyramid levels have triggered an expansion of the model (see Section 3.4.2), new empty levels are now appended to the bottom of the model pyramid and memory is allocated for the currently observed area. Image data not yet present in the model is added to  $\mathcal{I}_{\mathcal{M}}^l$ , while the merging of already existing data is described in the following.

Model expansion

Let  $\mathcal{I}_{\mathcal{M}}^l(x, y)$  be individual pixels in the Laplacian model pyramid on level  $l$  for which the observation provides valid pixels  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l(x, y)$  that need to be merged, i.e., the pixels have passed the outlier test (see Section 3.4.3). Furthermore, let  $\mathcal{L}_{\text{curr} \rightarrow \mathcal{M}}$  be the level-of-refinement map containing the real-valued level indices of the pixels of  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$  with respect to the model pyramid levels (see Section 3.4.1).

Inspired by online 3D scene reconstruction [ZSG\*18], confidence values are determined, which refer to the pixels' reliability on each Laplacian level  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l$ . In the case of image fusion, the confidence can be related to the contrast in a focused image, which can be measured using the *modulation transfer function*<sup>1</sup> (MTF) of a camera; see, for example, Williams and Becklund [WB89]. Independent of the specific camera used, the MTF states that the imaging system's ability to transfer contrast decreases at higher spatial frequencies. Consequently, any observation acquired closer to the imaged object should be superior to other observations taken from farther distances. Hence, the per-pixel level of refinement is used to represent the observation's confidence  $\mathcal{L}_{\text{curr} \rightarrow \mathcal{M}}^l(x, y)$  on each pyramid level  $l$ ; that is,  $\mathcal{L}_{\text{curr} \rightarrow \mathcal{M}}^l$  denotes the Gaussian decomposition of the level-of-refinement map  $\mathcal{L}_{\text{curr} \rightarrow \mathcal{M}}$ .

Confidence values

Finally, all corresponding model pixels are replaced by observation pixels with superior confidence:

Merging of model and observation

$$\begin{aligned} \mathcal{I}_{\mathcal{M}}^l(x, y) &\leftarrow \begin{cases} \mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l(x, y), & \text{if } \mathcal{L}_{\text{curr} \rightarrow \mathcal{M}}^l(x, y) < \mathcal{L}_{\mathcal{M}}^l(x, y), \\ \mathcal{I}_{\mathcal{M}}^l(x, y), & \text{otherwise,} \end{cases} \\ \mathcal{L}_{\mathcal{M}}^l(x, y) &\leftarrow \min(\mathcal{L}_{\text{curr} \rightarrow \mathcal{M}}^l(x, y), \mathcal{L}_{\mathcal{M}}^l(x, y)), \end{aligned} \quad (3.2)$$

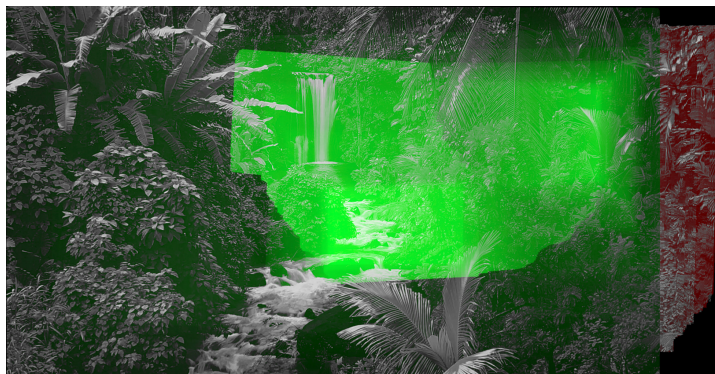
for all corresponding Laplacian levels  $l \in [l_{\min}^{\text{curr} \rightarrow \mathcal{M}}, l_{\max}^{\text{curr} \rightarrow \mathcal{M}})$ . This operation ensures that the model stores the observation acquired closest to the scene on a per-pixel basis, i.e., the model contains a single and reliable observation with maximal contrast. As the model frequencies are also replaced on coarser Laplacian levels (while retaining the color of the Gaussian level), a photometrically

<sup>1</sup>The MTF measures the imaging system's ability to transfer contrast from an object to the image plane at a given spatial frequency (percentage of transferred contrast relative to low frequencies). It is defined as the magnitude of the complex-valued *optical transfer function* (OTF), which is the Fourier transform of the camera's *point-spread function* (PSF).

and geometrically consistent reconstruction is produced without any further post-processing.

### 3.4.5 Refinement Guidance

After the refinement, the model’s level-of-refinement map is rendered to make the user aware of the current model composition in terms of accumulated image detail. Figure 3.5 shows such a visualization for an example refinement. Using this visual guidance, the user can steer the acquisition process according to his or her needs and interests. By also visualizing areas where the initial scene area, defined by the reference image  $\mathcal{I}_0$ , has been extended by further observations, the user is made aware of regions where photometric consistency is not guaranteed (red areas in Figure 3.5).




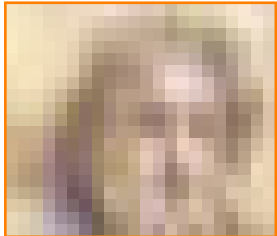



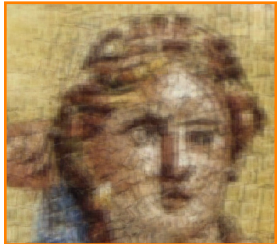






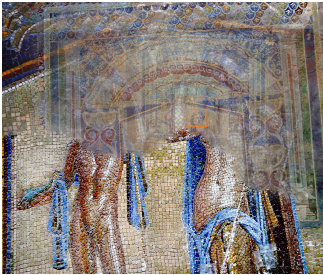


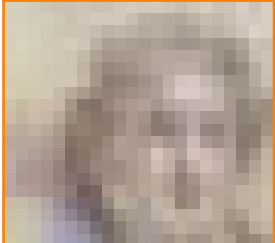

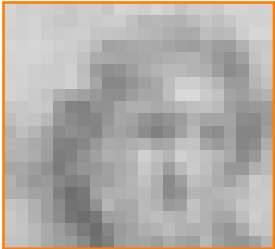


**Figure 3.5:** Rendering the level-of-refinement map shows the so-far refined areas (green). The brighter the green color, the finer the available geometric detail. Red areas indicate regions with potential photometric inconsistencies.









## 3.5 Results

The quality and robustness of the proposed *progressive refinement imaging* approach are evaluated using eight data sets, consisting of photos as well as videos captured with seven different camera models (plus five unknown cameras). For each record, the reference image  $\mathcal{I}_0$  is locally refined by fusing additional images of the same scene but taken closer to the object or by zooming in.

Table 3.2 reports the results of 14 state-of-the-art 2D imaging methods for an example data set (*House of Neptune and Amphitrite mosaic*), using a sequence of input photos captured with different camera-to-object distances. Experiments revealed that most of these methods fail to process the data sets properly, and the following behaviors can be observed. The method either

	Reference image $\mathcal{I}_0$	Additional input image (close-up)
		
	Result	Result (close-up)
ADG Panorama Tools Pro [Alb]		
Autopano [Kol18]		
AutoStitch [Bro18]		
GigaPan Stitch [Gig]		

	Reference image $\mathcal{I}_0$	Additional input image (close-up)
		
	Result	Result (close-up)
Hugin [d'A]		
Image Composite Editor [Mic]		
Panorama Composer 3 [Fir]		
Panorama Studio 3 Pro [tsh]		

	Reference image $\mathcal{I}_0$	Additional input image (close-up)
		
	Result	Result (close-up)
Photoshop CC 20.0.1 [Ado]		
PTGui Pro [New]		
Proposed method		

**Table 3.2:** Comparison to 2D imaging methods. The reference image  $\mathcal{I}_0$  is merged with additional input images, of which the one with the shortest camera-to-object distance is shown for comparison. Except for Autopano [Kol18], AutoStitch [Bro18], and the proposed method, none of the approaches achieve a convincing result, i.e., the merged output image does not contain the fine details provided by the additional input images. Moreover, the methods Panoweaver Professional [Eas], PhotoStitcher [Teo], PTAssembler [Taw], and Stitcher 4 [3DV] reported that no matching of the input frames is possible, resulting in no output.

(1) reported that no matching of the input frames is possible or (2) did not achieve any refinement, i.e., the merged image did not contain the fine details provided by the input images, or (3) enforced a typical panorama scenario, resulting in a merged image where the input photos are aligned horizontally.

AutoStitch [Bro18, BL07] and Kolor Autopano [Kol18], which is using the AutoStitch technology, were the only systems able to reach a refinement. Unfortunately, AutoStitch crashes if the resolution of the merged image exceeds 30 942 px in one dimension. Furthermore, no access to Eisemann et al.'s Photo Zoom [EESM10] was possible, which precludes experimental comparison.

In the following, the proposed method is compared to the unrefined input and the result of Autopano [Kol18].

### 3.5.1 Refinement Using Different Sources of Imagery

For this experiment, photos captured from different sources on different dates using different cameras from various unknown positions are used. All photos are publicly available, e.g., from Flickr or Wikimedia Commons, unedited and labeled for reuse with modification by the author.

*House of Neptune and Amphitrite mosaic:* A photo of the mosaic at the *House of Neptune and Amphitrite* in Herculaneum captured with a Pentax Optio S7 by Johnboy Davidson [Dav07] is refined using six additional close-up photos [AlM06, Amp16, HK13, Ras17, Cra14, Rie09] captured with six different cameras (FUJIFILM FinePix F900EXR, Panasonic DMC-ZS6, Nikon D7100, 3 unknown cameras) in the years 2007, 2006, 2014, 2011, 2017, 2014, and 2009, respectively (see Figure 3.1).

This data set comprises challenging photometric inconsistencies, e.g., due to different camera hardware, exposure, and color balance used in the acquisition. As shown in Figure 3.1, feeding this data set into Autopano results in a geometrically consistent but photometrically inconsistent image, as Autopano tries to generate smooth transitions between the individual photos. In contrast, the proposed method yields photometric and geometric consistency.

### 3.5.2 Inconsistent Illumination

In this section, the robustness against illumination changes is evaluated using the following four data sets:

*Panorama at different daytimes:* A panorama shot is refined using nine additional zoomed-in photos that were taken at different daytimes with approximately 1h time difference in the afternoon, showing the same scene with decreasing sunlight, locally changing shadows and clouds, and with a fixed camera position (see Figure 3.6). All photos were captured with a Panasonic DMC-FZ28 ( $3648 \times 2736$  px mode).

*Wall painting at different daytimes:* A photo of an outside wall painting is refined using 38 additional photos that were taken at different daytimes during a single day, showing the same scene with varying sunlight and locally changing shadows on the wall from strongly varying camera poses (see Figure 3.7). All photos were captured with a Samsung Galaxy S8 built-in camera ( $4032 \times 1960$  px mode).

*Glossy poster:* The first frame of a video sequence capturing a glossy poster is refined using the remaining 847 frames that were captured closer to the scene (every other frame of a 57s video clip). This sequence comprises frames with very strong photometric inconsistencies in terms of reflections. It was acquired with a Samsung Galaxy S8 built-in camera in 1080p mode (see Figure 3.8).

*Deësis mosaic:* An overview photo of the Mosaic of the Deësis in the Hagia Sophia captured by Steven Zucker [Zuc12h] is refined using nine additional close-up photos [Zuc12a, Zuc12d, Zuc12i, Zuc12c, Zuc12j, Zuc12e, Zuc12g, Zuc12f, Zuc12b], where sunlight passes through the windows, resulting in a pattern of differently illuminated areas. All photos were captured with a Sony DSC-RX100 (see Figure 3.9).

### Global Illumination Changes

The first two data sets, i.e., *Panorama at different daytimes* (Figure 3.6) and *Wall painting at different daytimes* (Figure 3.7), contain major changes in global illumination, while *Panorama at different daytimes* additionally contains geometric inconsistencies due to changes in cloudiness. While Autopano has major difficulties in handling the illumination changes, geometric variations (*Panorama at different daytimes*), and different camera poses (*Wall painting at different daytimes*), the proposed approach is able to combine both data sets into a photometric and geometric consistent image. The close-ups of the refined images depicted in the comparisons demonstrate the proper handling of photometric and geometric information during progressive image refinement.

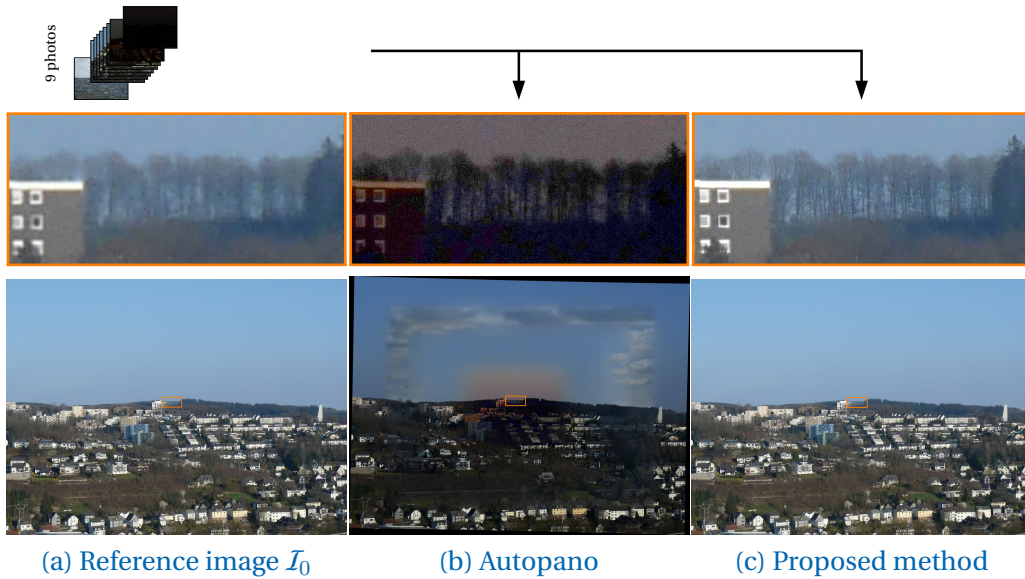


Figure 3.6: Panorama at different daytimes.

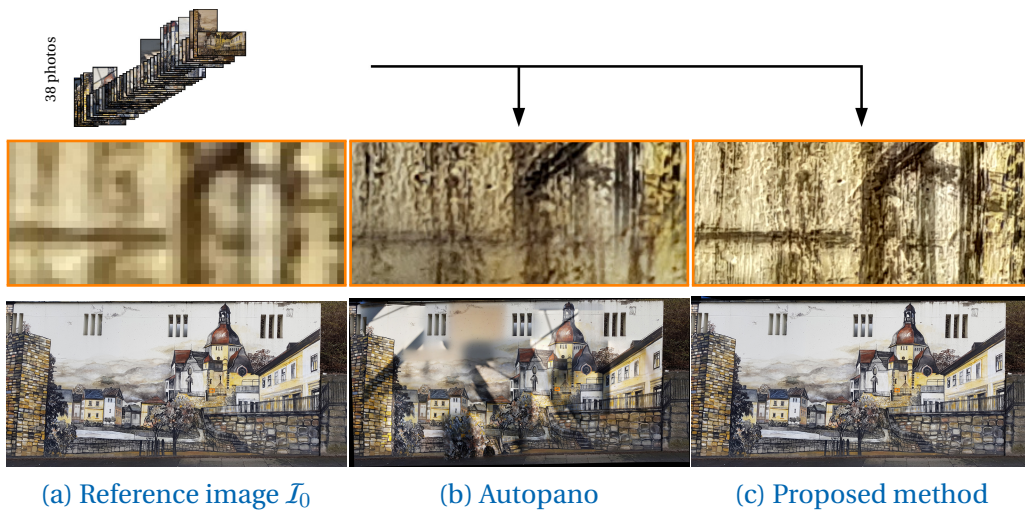


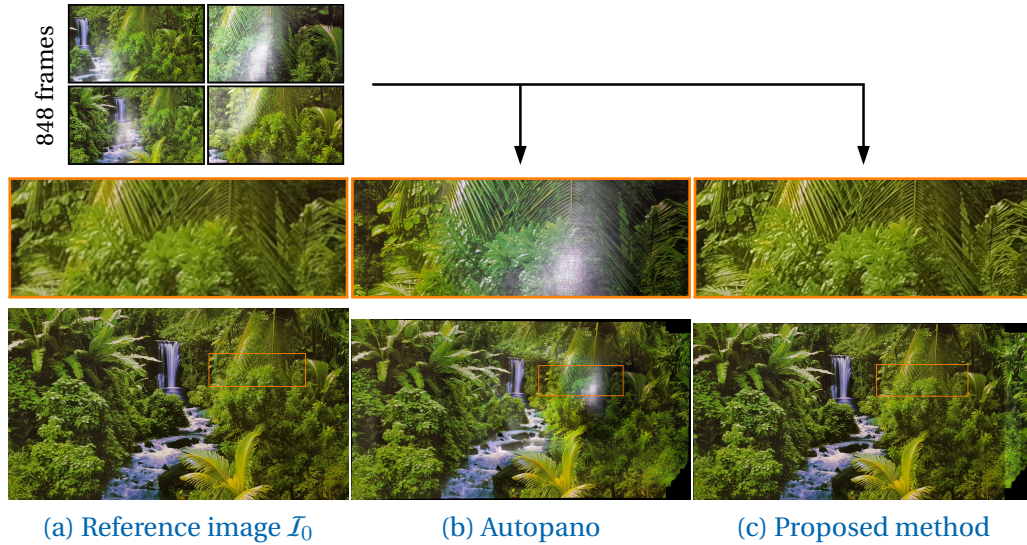
Figure 3.7: Wall painting at different daytimes.

### Local Illumination Changes

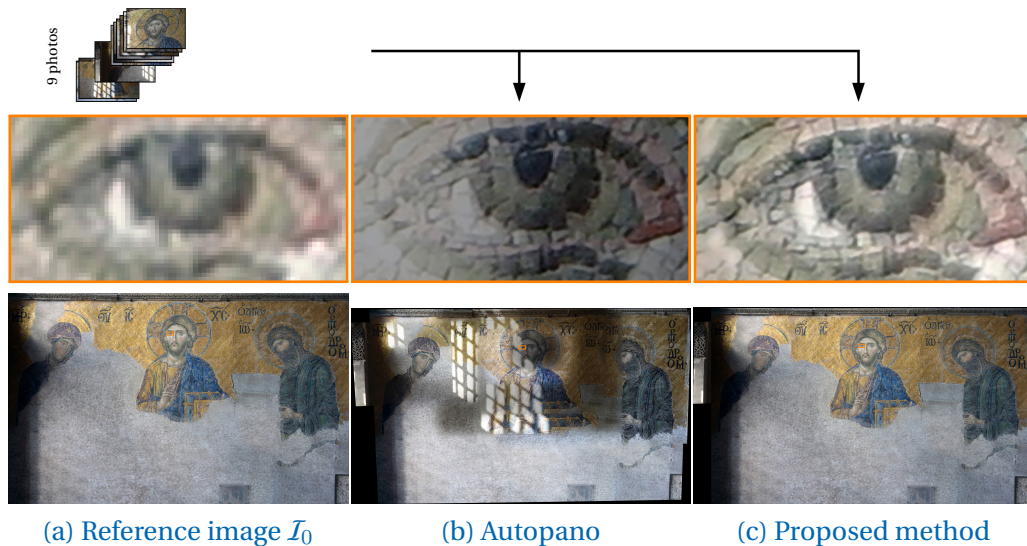
The second two data sets, i.e., *Glossy poster* (Figure 3.8) and *Deësis mosaic* (Figure 3.9), contain strong local illumination variations due to photoflash reflections and shadow casts by a window grating, respectively. In both scenarios, Autopano incorporates local illumination constellations from different close-up images into the reconstruction, resulting in very inconsistent intensity



distributions in the output image. The proposed progressive method is able to generate a photometric consistent result even under these extreme lighting conditions (see also Figure 3.5 for a visualization of the refined areas for the *Glossy poster* data set).



**Figure 3.8:** *Glossy poster*. The four sample frames (top) are part of the input video sequence, showing that the clip contains strong reflections.



**Figure 3.9:** *Deësis mosaic*.

### 3.5.3 Inconsistent Scene Geometry

The robustness against strong geometric variations is evaluated using the following two data sets:

*Moving cars:* A panorama shot showing a freeway is refined using two additional zoomed-in photos where the cars have been moving (see Figure 3.10). All photos were captured with a Panasonic DMC-FZ28 (3648 × 2736 px).

*Streetart fisheye:* An ultra-wide-angle shot of street art graffiti captured with an unknown camera with a fisheye lens by Mike Lambert [Lam14a] is refined using an additional photo [Lam14b] captured with a normal lens (see Figure 3.11).

Additionally, the per-pixel outlier masks are depicted, generated for both data sets; see Figures 3.10 and 3.11.

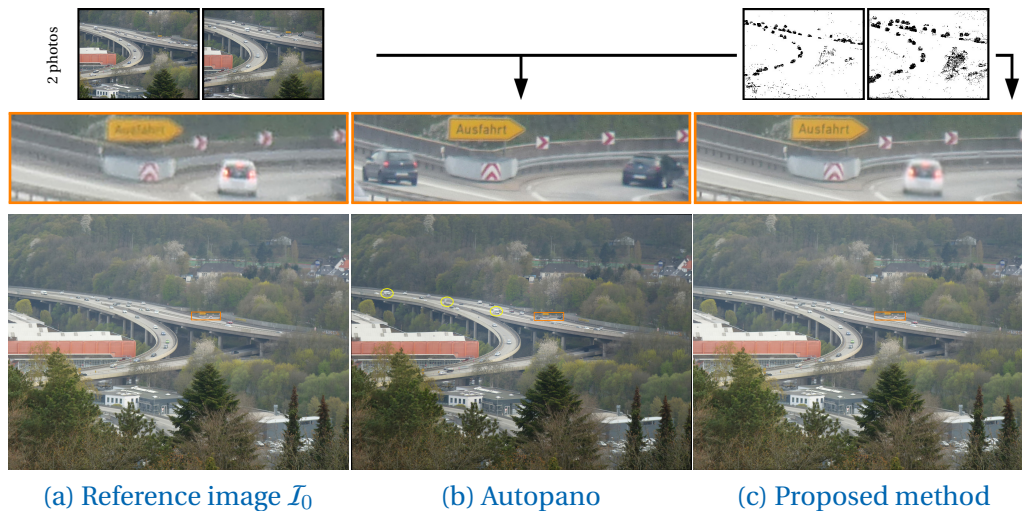
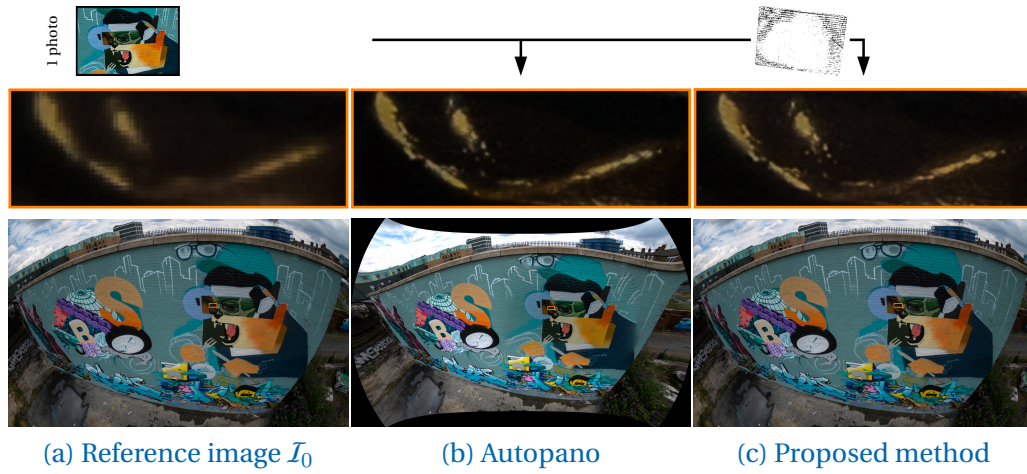


Figure 3.10: *Moving cars.*

Dynamic scenes  
Ultra wide-angle

The main difference between both data sets is the type of geometric inconsistency. While the *Moving cars* data set comprises locally unconstrained geometric variations, the *Streetart fisheye* data set suffers from strong lens distortions that can be seen as globally constrained geometric inconsistencies. Both scenarios exhibit the different approaches taken by Autopano and the proposed method. While Autopano generates visually pleasing output images in both cases, they both contain a mixture of all provided images leading to, e.g., duplications of moving cars (see yellow circles in Figure 3.10b) and a blended, deformed geometry in case of strongly varying lens artifacts (see Figures 3.11



**Figure 3.11:** *Streetart fisheye.*

and 3.13). In contrast, the proposed method takes the initial image as a photometric and geometric reference and adjusts subsequent images to match this reference as closely as possible before adding details. Therefore, it delivers a consistent geometric result, i.e., there are no multiple instances of moving objects or unexpected lens properties. Autopano, however, always selects scene fragments with the longest focal length, whereas the proposed approach does not refine moving objects in the reference image, potentially leaving unsharp objects untouched; see Figure 3.10c. By assessing the depicted outlier masks, the overall quality of the two-stage registration process can be evaluated, as described in Section 3.4.1 (see also the discussion in Section 3.5.4): in the *Moving cars* data set, mainly moving cars and trees are discarded, while in the *Streetart fisheye* data set, scene fragments of the strong lens distortion are removed that cannot be fully compensated by the optical flow stage.

### 3.5.4 Ablation Study

In the following, the influence of essential processing stages of the progressive image refinement pipeline is discussed. For this evaluation, two additional sequences are used:

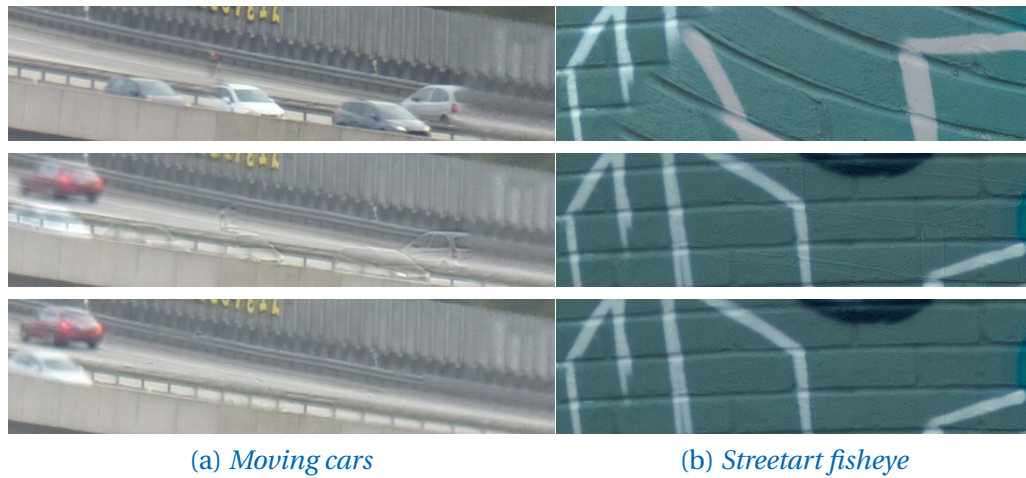
*Starlight:* There are two different captures of this sequence. (1) A sequence of five photos captured free-hand with a Samsung Galaxy S8 built-in camera with  $1920 \times 1080$  px resolution, taken from an advertising poster (see Figure 3.14), and (2) a 477 frames video captured with a Samsung Galaxy S8 built-in camera, downsampled to  $960 \times 540$  px (see Figure 3.15). The second video sequence was generated to evaluate the approach in the case of imagery with robust and noise-reduced maximum frequency.

**Fine image registration** The *fine image registration* stage has a strong impact on the quality of the final result. Figure 3.12 demonstrates the effect of the locally re-aligned image registration using optical flow on the *Starlight (5 images)* data set. Even for the comparable small lens distortion in this data set, the additional optical flow significantly improves the local matching of object details. This becomes even more apparent when images with strong optical distortions, such as the one in the *Streetart fisheye* data set, are considered that cannot be modeled using a homography; see Figure 3.11.



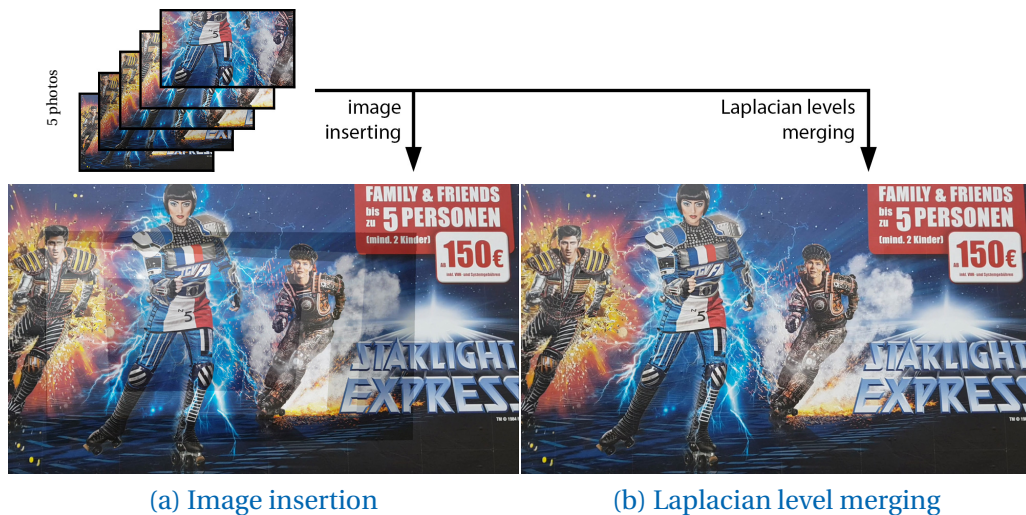
**Figure 3.12:** A close-up comparison of the *Starlight (5 images)* data set without (left) and with locally re-aligned image registration (right).

**Outlier removal** The effect of the *per-frame outlier removal* is demonstrated in the *Panorama at different daytimes* data set; see Figure 3.6. Here, the last input frame, which has been captured in very weak sunlight, has not passed the global consistency check, i.e., it has been discarded from model refinement, since it does not provide additional image details. In comparison, Autopano performs a histogram equalization and incorporates the last frame, overwriting the details of the previous frames, which results in a loss of detail and increased noise in the refined image. For the *Glossy poster* data set, 2.01% of the input frames were rated unable to contribute finer details; hence, only newly observed areas were incorporated into the model if available. The *per-pixel outlier removal*, as described in Section 3.4.3, is evaluated in Figure 3.13, which shows close-ups of the *Moving cars* and *Streetart fisheye* scenarios. Disabling the local outlier removal yields artifacts, manifesting as slight ghosting of cars and mismatching seams in the *Moving cars* and *Streetart fisheye* scenarios, respectively. Both effects vanish nearly completely if the per-pixel outlier removal is enabled.



**Figure 3.13:** Influence of the per-pixel outlier removal. *Top row:* the result of Autopano. *Middle row:* the proposed method without outlier removal. *Bottom row:* the proposed method with outlier removal.

**Laplacian levels merging** For the *Starlight (5 images)* data set, Fig. 3.14 compares a simple insertion of the registered observations (Fig. 3.14a), i.e., a replacement on a flat image representation, with the described replacement strategy on Laplacian levels (Fig. 3.14b), demonstrating its ability to maintain photometric consistency.



**Figure 3.14:** *Starlight (5 images)*: 5 photos captured with a Samsung Galaxy S8 built-in camera were merged. The described method is combined with a simple image insertion (left), i.e., a replacement on a flat image representation, and the proposed replacement strategy on Laplacian levels (right).

**Replacement strategy for merging** The choice of replacing frequencies instead of blending them is mainly motivated by the goal of being able to fuse several hundred images without global optimization. As demonstrated in Figure 3.15, blending a larger set of input images using cumulative average weighting leads to a gradual reduction of image details, even in the case of the *Starlight* (477 frames) scenario with its robust and noise-reduced high frequencies. Due to the non-perfect nature of image registration, blending all observations will wash out geometric details that will never be fully recovered by further blending operations. The replacement strategy, on the other hand, preserves the fine geometric details, as it composites single, reliable observations with maximal contrast.



**Figure 3.15:** *Starlight* (477 frames): Comparison between a blending using cumulative average weighting (left) and the described replacement strategy (right) applied to a 477 frames sequence captured with a Samsung Galaxy S8 built-in camera.

### 3.5.5 Comparison of Required Resources

Table 3.3 shows for each data set a comparison of the peak total RAM usage and the processing time for the complete refinement process for both Autopano and the proposed method. This comparison demonstrates that global optimization significantly increases memory requirements and runtime. This is unavoidable as global optimization methods have to keep all relevant images in memory in order to process them jointly. Especially for the video data set *Glossy poster*, the memory requirements increase severely by a factor of approximately 40, while the processing time increases by a factor of 5. In contrast, the proposed approach of progressively refining the image is much more lightweight and continuously eliminates redundancy, substantially lowering resource requirements.

For the implementation of the proposed pipeline, mainly the adaptive Laplacian pyramid has been optimized as described in Section 3.4, while basic image processing operations, such as feature extraction and optical flow, are taken from OpenCV as is.

### 3.5.6 Limitations and Discussion

The current pipeline is able to maintain photometric consistency only within the region observed by the initially captured overview shot  $\mathcal{I}_0$ . While the system is capable of incorporating images that are partially outside this initial region, at the seam to  $\mathcal{I}_0$ , it yields geometric but no photometric consistency. Since the refined image is always consistent with the reference image, unintended photometric effects in  $\mathcal{I}_0$ , e.g., photoflash reflections, will not be compensated by additional photos. Furthermore, the fine image registration using optical flow cannot correct strong optical distortions or images containing severe photometric inconsistencies; however, the per-pixel outlier removal compensates for these artifacts almost entirely; see Figure 3.13.

The current implementation is not re-entrant, i.e., it does not support the continuation of a previously acquired model image represented in a Laplacian pyramid as described in Section 3.3. While the system is truly progressive, in that information is fed frame-by-frame without any global optimization, the current implementation is interactive but not real-time capable. So far, the pipeline components have not been fully optimized and tightly integrated to achieve optimal load and compute balancing, e.g., by leveraging concurrency. Apparently, faster executions of dense image processing operations, e.g., optical flow, will have a direct impact on the performance (see Table 3.3).

**Table 3.3:** Comparison of the resources required for the complete refinement process, with the number of input photos/pixels in total (using an AMD Ryzen Threadripper 1950X with 128 GB RAM and an NVIDIA GeForce GTX 1080 Ti). For the *Glossy poster* data set, the timings per pipeline stage for the proposed approach are: Image registration: 06:48 (min:s)/Pyramid generation: 03:23/Outlier removal: 01:20/Model expansion: < 00:01/Merging Laplacian levels: 02:43.

	Processing time (min:s)		Peak total RAM usage (GB)	
	Autopano	Proposed method	Autopano	Proposed method
<i>Deësis mosaic</i> (10 photos/0.12 gigapixel)	01:52	00:40	31.16	5.12
<i>Glossy poster</i> (848 frames/1.76 gigapixel)	70:34	14:14	121.53	2.23
<i>House of Neptune and Amphitrite mosaic</i> (7 photos/0.01 gigapixel)	01:25	00:06	17.52	1.45
<i>Moving cars</i> (3 photos/0.03 gigapixel)	00:26	00:05	4.15	2.00
<i>Panorama at different daytimes</i> (10 photos/0.10 gigapixel)	01:18	00:27	14.97	2.57
<i>Streetart fisheye</i> (2 photos/0.03 gigapixel)	00:33	00:05	7.38	2.52
<i>Wall painting at different daytimes</i> (39 photos/0.31 gigapixel)	06:13	02:35	68.73	7.56

## 3.6 Summary

This chapter presented a simple, yet very effective and efficient technique for the progressive incorporation of large image sequences into a single, geometrically and photometrically consistent image. Conceptually, the proposed approach has no restriction to object-space resolution, camera-to-object distance, camera intrinsics, or acquisition setup. Additionally, it does not require global optimization applied to the complete input image set or parts thereof. It achieves geometric registration using a two-stage approach that combines a homography and an additional local re-alignment using a flow field. It can



handle global as well as local illumination changes, yielding photometrically consistent results. Due to its progressive nature, the proposed approach allows for a valid and consistent reconstruction at any moment during the refinement process without any post-processing.



# 4

## Depth-Assisted Progressive Refinement Imaging for 3D Scenes

*The increasing on-board compute power of mobile camera devices gave rise to a class of digitization algorithms that dynamically fuse a stream of camera observations into a progressively updated scene representation. Previous algorithms either obtain general 3D surface representations, often exploiting range maps from a depth camera, such as, Kinect Fusion, etc.; or they reconstruct planar (or distant spherical, respectively) 2D images with respect to a single (perspective or orthographic) reference view, such as, panoramic stitching or aerial mapping. This chapter sets out to combine aspects of both, reconstructing a 2.5-D representation (color and depth) as seen from a fixed viewpoint, at spatially variable resolution. Based on the previous chapter on progressive refinement imaging for planar scenes, this chapter proposes a hierarchical representation that enables progressive refinement of both colors and depths by ingesting RGB-D images from a handheld depth camera that is carried through the scene.*

*The proposed system is evaluated by comparisons against state-of-the-art methods in 2D progressive refinement and 3D scene reconstruction, using high-detail indoor and outdoor data sets comprising medium to large disparities. As this chapter will show, the restriction to 2.5-D from a fixed viewpoint affords added robustness (particularly against self-localization drift, as well as backprojection errors near silhouettes), increased geometric and photometric fidelity, as well as greatly improved storage efficiency, compared to more general 3D reconstructions. The proposed representation has the potential to enable scene exploration with realistic parallax from within a constrained range of vantage points, including stereo pair generation, visual surface inspection, or scene presentation within a fixed VR viewing volume.*

*The approach described in this chapter has been published [KWK23] in *Computers & Graphics*, Vol. 115, 2023.*

---

**T**he past decade has seen an emergence of interactive scene digitization systems that dynamically fuse a stream of sensor observations into a progressively updated scene representation. The key benefit of dynamic (“online”) reconstruction over offline methods (where all data is captured first before a reconstruction happens in a post-process) is the ability to interactively capture more data where the current reconstruction indicates insufficient data

Online  
reconstruction

so far [RHHL02].

#### 2D and 3D approaches

This principle is now prominently used both for 2D imaging (e.g., panorama mode in mobile phone camera applications) and 3D model reconstruction; the latter was popularized through the introduction of affordable color+depth (RGB-D) cameras and immediately spawned the field of online scene digitization from handheld RGB-D cameras, pioneered by KinectFusion [IKH\*11, NIH\*11].

Even though a full 3D reconstruction (geometry and color) has the appeal of capturing more comprehensive aspects of a scene, and despite many modern mobile phones featuring RGB-D sensors, 2D imaging remains the most popular modality in the mainstream. It can be argued that, besides other reasons, that popularity is mainly due to most *output* devices being 2D, due to the tighter control over the output's appearance, but also due to one's ability to take in a 2D scene at a single glance while 3D content requires an interface for navigation and exploration.

#### 2D scene imaging

In recognition of the enduring importance of 2D scene imaging, recent work adapts the concept of online scene capture to the 2D domain, creating a *variable-resolution* RGB image from unstructured image collections. In the last chapter, interactive, *progressive refinement imaging* was introduced to bridge panoramic stitching and handheld “fusion-style” digitization. Similarly, Licorish et al. [LFS21] use adaptive compositing of pre-registered images with variable resolution captured with a single camera with optical zoom. While these methods support the high-quality photometric integration of images across a wide range of object-space resolutions, they are, however, strictly limited either to a perfectly fixed vantage point, or to scenes with minimal depth disparity; in particular, they are prone to parallax-induced misalignment artifacts whenever the camera is moved within the scene to obtain higher-resolution close-ups of objects.

#### Depth-assisted image refinement

The work presented in this chapter aims at overcoming this restriction by progressively reconstructing an auxiliary depth map alongside an image reconstruction. This adaptively refined depth map is used to compensate for parallax due to depth disparities and further assists with self-localization of the camera. In a departure from common approaches for scene reconstruction from RGB-D images, however, and more in line with image-based rendering, the proposed method strictly decouples color data from the coarse and potentially incomplete geometry representation. Thus, the inherent difference in data quality between color and depth sensors is accommodated, which greatly increases robustness of the scene capture.

Just like online 3D scene reconstruction approaches, handheld RGB-D camera streams are taken as input. Similar to Chapter 3, color differences are fused

hierarchically in a sparse Laplacian pyramid to naturally achieve texture consistency by blending not colors, but highpass-filtered image color details into that Laplacian hierarchy. In order to also aggregate depth values, however, the approach proposed in this chapter has to overcome several challenges intrinsic to range images that make them harder to fuse than the typically high-quality color channels of RGB-D: (1) significantly increased noise, including outliers and missing data, often correlated with salient features like silhouettes, (2) lower effective resolution, and (3) relative alignment errors with respect to the color imager. To cope with these depth errors and artifacts, the proposed progressive and adaptive *depth* refinement uses an explicit depth model instead of a Laplacian pyramid to prevent noise amplification. Moreover, the standard averaging approach, frequently used in popular online 3D geometry reconstruction approaches, is traded for a progressive per-pixel voting scheme.

The resulting system enables reliable image capture of general scenes, using an RGB-D camera where the operator first takes an overview shot before walking into the scene to take close-ups where added image detail is desired. By bridging between 2D and 3D approaches, the proposed system manages to mitigate limitations of either modality. Parallax-induced errors of 2D imaging approaches are virtually eliminated, and texture inconsistencies, that to date require global post-optimization, yielding non-progressive and non-interactive systems [ZK14, FYL\*21], are resolved on the fly. Last but not least, by anchoring the reconstruction in the initial overview shot, camera-drift that plagues existing 3D scene reconstruction methods is eliminated.

Bridging between 2D and 3D

The proposed system is evaluated by comparisons against state-of-the-art methods in 2D progressive refinement and 3D scene reconstruction, using high-detail indoor and outdoor data sets comprising medium to large disparities. As we will see, the restriction to 2.5-D from a fixed viewpoint affords added robustness (particularly against self-localization drift, as well as backprojection errors near silhouettes), increased geometric and photometric fidelity, even in the presence of illumination changes, as well as greatly improved storage efficiency, compared to more general 3D reconstructions.

Contributions

In summary, this work contributes:

- *Disparity-corrected* adaptive image refinement that fuses observations into a high-quality, geometrically consistent, adaptive-resolution 2.5-D image, even in the presence of *silhouettes* and strong *scene parallax*, while retaining photometric consistency.
- *Progressive and local* geometric and photometric optimization for *drift-free* color and depth alignment.
- *Decoupled color and depth representation*, using a sparse Laplacian for

color and sparse Gaussian for depth, that straddles high color fidelity with artifact-prone depth readings.

- A bespoke *progressive per-pixel depth voting scheme* that outperforms conventional cumulative average weighting.

Apart from creating high-fidelity, adaptive-resolution 2D content, the proposed depth-enhanced representation has the potential to enable scene exploration with realistic parallax from within a constrained range of vantage points, including stereo pair generation, visual surface inspection, or scene presentation within a fixed VR viewing volume.

## 4.1 Photometric Scene Reconstruction with Parallax Compensation

**Previous methods** As image refinement approaches for 2D RGB images (see Chapter 3) intrinsically assume an almost planar (or far-distant) scenery, they are restricted in handling the disparity in non-planar scenes in closer vicinity to the camera; see Section 4.6 for an evaluation of this limitation. In contrast, by exploiting range maps from a depth camera to compensate for scene parallax, 3D scene reconstruction methods are inherently linked to the objective of this chapter, as these methods implicitly handle disparity by fusing depth information into a full 3D model.

**Photometrically optimized 3D scene reconstruction** Commonly, there is a significant amount of photometric inconsistencies in a 3D-reconstructed scene, mainly due to sensor noise and inaccurate camera pose estimates. Therefore, high-quality photometric reconstruction is commonly achieved via post-optimization applied to a pre-reconstructed scene geometry, which is potentially converted into a mesh. For example, Zhou and Koltun [ZK14] propose a post-processing approach to optimize the poses of the color frames in a non-rigid manner using image space deformations to achieve improved photometric consistency. Moreover, photometric consistency is achieved using pose refinement for keyframes [ZK14, JJKL16], potentially segmenting the model and applying intensity and gain correction or synthesizing textures from the RGB imagery [FYL\*21, BKR17, FYY\*18, HDGN17, RFB18, WG18], or using super-resolution approaches projecting individual observations into the keyframes [BPC17, WMG14, MSC15]. Alternatively, the photometric information can be accumulated in a voxel grid with a higher resolution than the one used for fusing the geometric information [LLOC15]. Other methods aiming at high-quality photometric reconstruction use joint optimization for the

**Offline texture optimization**

**Joint optimization**

camera poses, the scene's geometry and texture [FYLX20, WG18], or intrinsic material properties [MKC\*17, WG18] to improve the overall 3D geometric and photometric consistency.

While all approaches mentioned above are not interactive or real-time capable, some methods reduce the computational complexity to achieve interactive framerates. Meilland and Comport [MC13] propose a 2.5-D scene representation. They fuse low-resolution RGB-D image sequences into a single super-resolution  $2560 \times 1920$  px RGB-D map by applying a fixed super-resolution factor (4 in this case) and deblur the result in a post-processing step. Lee et al.'s TextureFusion approach [LHD\*20] generates a full 3D model representing higher-resolved texture information using an axis-aligned parallel projection onto the implicit surface within individual TSDF voxels containing the iso-surface. This allows for real-time geometry reconstruction and texture fusion using standard weighted blending methods. Their follow-up work [HLMK21] allows for the real-time acquisition of photometric normals jointly represented with texture information.

Online texture optimization

**NeRF and other learning-based approaches** Recently, *Neural Radiance Field (NeRF)* approaches have gained much attention, which generally learn an implicit latent representation of a radiance field captured at known camera poses [MST\*20]. There have been several attempts to enhance NeRF-like approaches towards the interactive processing of real-world RGB or RGB-D data. For example, the NeRF in the wild method [MBRS\*21] addresses photometric variations and transient objects in an unstructured photo collection with known camera poses, while the GNeRF approach [MCL\*21] learns the camera pose parameters utilizing Generative Adversarial Networks (GANs) for this task. Moreover, neural implicit representations have been enhanced towards interactive RGB-D scene reconstruction [SLOD21, ZPL\*22]. The recent NICE-SLAM approach [ZPL\*22] achieves interactive frame rates of  $\sim 5$  fps. Still, compared to classical 3D scene reconstruction methods, the reconstruction quality of methods utilizing implicit neural representations is significantly lower than for classical approaches (see, e.g., the camera pose comparison in [ZPL\*22, Table 2]).

Learning-based reconstruction

In summary, none of these methods can handle high-quality photometric and geometric RGB-D image refinement in an interactive progressive fashion. Most specifically, existing RGB-D approaches do not involve direct updates of an unbounded multi-scale world representation to achieve local photometric and geometric refinement. Conceptually, the approach proposed in this chapter has been inspired by the 2.5-D scene representation from Meilland and Comport [MC13] to handle disparity properly, whereas maintaining color consistency is based on *progressive refinement imaging* using Laplacian pyramid fusion, presented in Chapter 3.

## 4.2 Pipeline Overview

The proposed progressive RGB-D image refinement pipeline is depicted in Figure 4.1. The input to this pipeline is a stream of RGB-D images  $\{\mathcal{I}_i, \mathcal{D}_i\}$  comprising color and depth images for frame indices  $i$ . The initial frame  $\{\mathcal{I}_0, \mathcal{D}_0\}$  is expected to be a reference frame that covers the region and viewing direction of interest of the observed scene for all following frames  $\{\mathcal{I}_i, \mathcal{D}_i\}$ ,  $i > 0$ . Unlike the usual  $360^\circ$  lateral scan in scene reconstruction, *progressive refinement imaging* deliberately aims at a “walking closer to the scene”-like camera path. The overall assumption here is that by approaching the scene, subsequent frames provide novel geometric and photometric details of the scene. Taking the reference frame as initial model  $\mathcal{M}$ , the proposed approach progressively refines this model by fusing the RGB-D stream into  $\mathcal{M}$ , yielding a geometric and photometric consistent RGB-D image with locally refined resolution. The model  $\mathcal{M}$  comprises a Laplacian color pyramid ( $\mathcal{I}_{\mathcal{M}}$ ) and a depth image ( $\mathcal{D}_{\mathcal{M}}$ ) with locally adapted resolution (see Section 4.3 for a detailed motivation). The main components of the pipeline are as follows (see Table 4.1 for a list of conventions used).

Pipeline overview

*Pre-Processing:* As the main objective is to improve the photometric quality of the final image, a *frame selection* is applied to identify the frame  $\{\mathcal{I}_{\text{curr}}, \mathcal{D}_{\text{curr}}\}$  with the sharpest color image within a small set of the latest consecutive input frames. Moreover, *noise reduction* is performed on the depth image to discard erroneous, e.g., flying pixels. Finally, the color and the depth image are *registered* by generating a high-resolution RGB-D image. See Section 4.4.1 for further details.

*Pose Estimation:* The current camera pose, represented by the rigid transformation  $\mathbf{T}_{\text{curr} \rightarrow \mathcal{M}}$  between the currently selected frame  $\{\mathcal{I}_{\text{curr}}, \mathcal{D}_{\text{curr}}\}$  and the model  $\mathcal{M}$ , is estimated in a two-stage process using sparse feature matching (*SURF*) and a subsequent *dense ICP* (see Section 4.4.2).

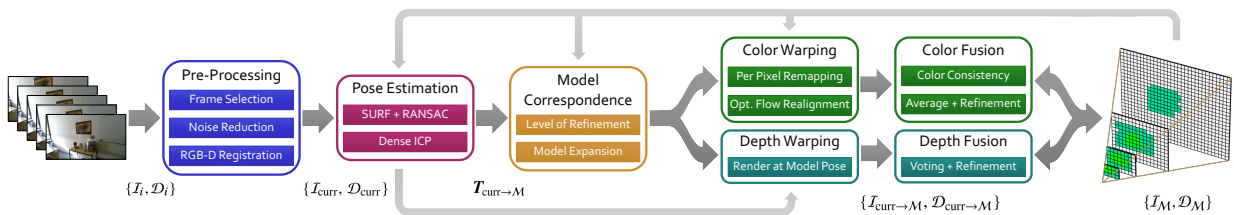


Figure 4.1: The proposed *progressive refinement imaging* pipeline for 3D scenes.



Table 4.1: List of conventions.

$\mathcal{I}_i, \mathcal{D}_i$	$i$ th input color and depth frame
$\mathcal{I}_{\text{curr}}, \mathcal{D}_{\text{curr}}$	Selected input frame of current iteration (observation)
$\mathcal{M}$	Model comprising components $\mathcal{I}_{\mathcal{M}}, \mathcal{D}_{\mathcal{M}}$
$\bar{\mathcal{I}}_{\mathcal{M}}, \bar{\mathcal{D}}_{\mathcal{M}}$	Pyramidal representations of accumulated color and depth, consisting of pyramid levels $\bar{\mathcal{I}}_{\mathcal{M}}^l, \bar{\mathcal{D}}_{\mathcal{M}}^l$ with level indices $l$
$c_{\mathcal{M}}$	Counter of observations fused into $\bar{\mathcal{I}}_{\mathcal{M}}$ (per-pixel attribute)
$v_{\mathcal{M}}$	Voting counter of $\bar{\mathcal{D}}_{\mathcal{M}}$ (per-pixel attribute)
$\mathbf{T}_{\text{curr} \rightarrow \mathcal{M}}$	Rigid camera transformation from observation to $\mathcal{M}$
$\mathcal{W}_{\mathcal{M} \rightarrow \text{curr}}$	Image-space mapping between $\mathcal{M}$ and the observation
$\bar{\mathcal{I}}_{\text{curr} \rightarrow \mathcal{M}}, \bar{\mathcal{D}}_{\text{curr} \rightarrow \mathcal{M}}$	$\mathcal{I}_{\text{curr}}$ and $\mathcal{D}_{\text{curr}}$ warped to $\mathcal{M}$ 's image space
$\bar{\mathcal{I}}_{\text{curr} \rightarrow \mathcal{M}}^l$	$\bar{\mathcal{I}}_{\text{curr} \rightarrow \mathcal{M}}$ decomposed into Laplacian levels with indices $l$
$\mathcal{L}_{\text{curr}}, \mathcal{L}_{\mathcal{M}}$	Level-of-refinement of $\bar{\mathcal{I}}_{\text{curr} \rightarrow \mathcal{M}}$ and $\bar{\mathcal{I}}_{\mathcal{M}}$ (per-pixel attribute)
$l_{\text{min}}$	Corresp. level index of warped obs. within pyramid $\mathcal{M}$
$\text{roi}(\dots)$	Lateral boundaries of warped observation on $\mathcal{M}$ (region of interest)
$\mathcal{F}_{\mathcal{M} \rightarrow \text{curr}}$	Flow field between $\bar{\mathcal{I}}_{\mathcal{M}}$ and $\bar{\mathcal{I}}_{\text{curr} \rightarrow \mathcal{M}}$
$s_{\text{curr}}^l$	Similarity score of $\bar{\mathcal{I}}_{\text{curr} \rightarrow \mathcal{M}}^l$ (per-pixel attribute)
$\mathbf{K}_I, \mathbf{K}_D$	Intrinsic camera matrices of color and depth imager
$\mathcal{K}_{\text{prev}}, \mathcal{K}_{\text{curr}}$	2D keypoints of prev. and curr. iteration
$\mathcal{P}_{\text{prev}}, \mathcal{P}_{\text{curr}}$	3D keypoints of prev. and curr. iteration
$\mathcal{V}_{\text{curr}}, \mathcal{V}_{\mathcal{M}}$	Vertex maps of $\bar{\mathcal{D}}_{\text{curr}}$ and $\bar{\mathcal{D}}_{\mathcal{M}}$

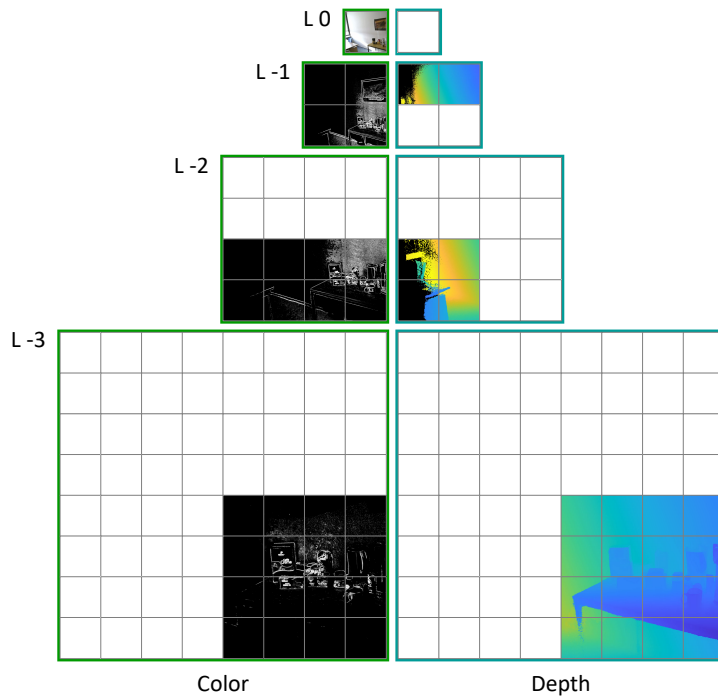
List of conventions

*Model Correspondence:* Dependent on the current frame's pose, the observation's potential to refine the model is estimated by determining a per-pixel *level-of-refinement* map. This may trigger an *expansion* of the model  $\mathcal{M}$  by extending the color pyramid and the adaptive depth representation appropriately (see Figure 4.2). For more details, see Section 4.4.3.

*Color & Depth Warping:* Due to their different nature, noise level, and purpose of the color and depth information, at this stage, both modalities are processed separately by splitting the reconstruction pipeline into two parallel strands (see Figure 4.1). The color warping is a per-pixel *remapping* using the estimated camera pose  $\mathbf{T}_{\text{curr} \rightarrow \mathcal{M}}$  and model depths  $\bar{\mathcal{D}}_{\mathcal{M}}$  to correct for parallaxes in the current color observation  $\bar{\mathcal{I}}_{\text{curr}}$ . An *optical*

*flow* is then applied for local re-alignment, resulting in  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$ . The depth information, however, is warped via *rendering* the meshed depth map  $\mathcal{D}_{\text{curr}}$  from the model's camera pose, yielding the warped depth map  $\mathcal{D}_{\text{curr} \rightarrow \mathcal{M}}$  (see Section 4.4.4).

*Color & Depth Fusion*: The color fusion is based on a cumulative *averaging* scheme, *refining* the initial reference image by adding details in a frequency-oriented way. Here, a *color consistency* check ensures the exclusion of inconsistent details. In contrast, depth fusion is performed using a combination of blending and replacement based on a progressive *voting* scheme, as the initial model depths can be very erroneous. For further details, see Sections 4.4.5 and 4.4.6.



**Figure 4.2:** An example layout of the model representation  $\mathcal{M}$ . Each pyramid level is regularly tiled with a fixed size. A tile is occupied by image data if refined data has been acquired; otherwise, it is unallocated. Color data is stored as sparsely occupied Laplacian pyramid in corresponding tiles across multiple levels, whereas depth data is stored as-is, within tiles that occupy the finest level of the respective depth observations.

## 4.3 Adaptive RGB-D Model Representation

The presented depth-assisted *progressive refinement imaging* approach for 3D scenes is based on data obtained by a commodity, handheld RGB-D camera such as Kinect v1, Xtion, or Kinect v2, that provides the RGB-D stream  $\{\mathcal{I}_i, \mathcal{D}_i\}$  with color images  $\mathcal{I}_i \in \mathbb{R}^3$  with RGB intensities and depth maps  $\mathcal{D}_i \in \mathbb{R}$  with camera-to-surface-distances in meters.

Any capture is expected to begin with an overview shot that defines the reference frame of the *variable-resolution* output image. Thus, the first frame,  $i = 0$ , sets a fixed reference viewpoint of the scene and initializes *model*  $\mathcal{M}$ , the representation for reconstructed RGB-D data.  $\mathcal{M}$  consists of two components,  $\mathcal{I}_{\mathcal{M}}$  and  $\mathcal{D}_{\mathcal{M}}$ , which represent the *variable-resolution* color image and depth map, respectively.

Decoupled model representation

Model color component  $\mathcal{I}_{\mathcal{M}}$  is a multi-scale representation based on Chapter 3, a sparsely occupied and dynamically expandable Laplacian pyramid [BA83a], consisting of pyramid levels  $\mathcal{I}_{\mathcal{M}}^l$ , where the level index  $l \in \mathbb{Z}$  decreases with finer resolution (i.e., receive negative indices).  $\mathcal{I}_{\mathcal{M}}$  is initialized by the input color image  $\mathcal{I}_0$ , which serves as a reference for maintaining color consistency. Over time, new, finer Laplacian levels  $\mathcal{I}_{\mathcal{M}}^{l < 0}$  are appended to the bottom of the pyramid, refining the initial reference image as novel details are added from subsequent frames  $\mathcal{I}_{i > 0}$ . As not all image regions are captured at the same level of object-space resolution when approaching the scene in a free-form camera path,  $\mathcal{I}_{\mathcal{M}}$  is sparsely occupied. Therefore, each pyramid level  $\mathcal{I}_{\mathcal{M}}^l$  is regularly tiled, where a tile ( $1024 \times 1024$  px) is allocated only if refined data was acquired. Moreover, each pixel has the following attributes: a counter  $c_{\mathcal{M}} \in \mathbb{N}$ , representing the number of fused observations (initialized with 1), and the model's level of refinement  $\mathcal{L}_{\mathcal{M}} \in \mathbb{R}$ , the so-far accumulated amount of detail (initialized with 0).

Color representation

In addition, this chapter introduces the model component  $\mathcal{D}_{\mathcal{M}}$ , an adaptively subdivided depth map representation. In contrast to color, the accumulated depth is not decomposed into band-pass filtered Laplacian levels but is stored as-is: experiments revealed that the difference operators produce artifacts in the range data due to amplifying noise, leading to erroneous model depths when merging frequencies of different observations.  $\mathcal{D}_{\mathcal{M}}$  can be interpreted as a sparsely occupied Gaussian pyramid that shares the pyramidal structure of  $\mathcal{I}_{\mathcal{M}}$  but has tiles allocated only at the finest level (see Figure 4.2). The first input depth map  $\mathcal{D}_0$  initializes  $\mathcal{D}_{\mathcal{M}}$ , and additionally, a voting counter  $v_{\mathcal{M}} \in \mathbb{R}$  is stored as a per-pixel attribute, representing a depth's reliability (initialized with 1).

Depth representation

## 4.4 Depth-Assisted Progressive Refinement Imaging

### 4.4.1 Pre-processing

The input to the reconstruction pipeline is a continuous stream of color images  $\mathcal{I}_{i>0}$  and depth maps  $\mathcal{D}_{i>0}$  that progressively refine the model color  $\mathcal{I}_M$  and model depth  $\mathcal{D}_M$ .

#### Frame Selection

Filtering of blurred frames

To avoid merging highly redundant data and to reduce processing time, the sharpest of 15 subsequent frames is selected for further processing if a maximum blur threshold  $\varepsilon_b = 0.32$  is not exceeded. As in [ZK14], the blur metric from [CDLN07] is used, applied to the color image  $\mathcal{I}_i$ . The selected frame of the current iteration  $\{\mathcal{I}_{\text{curr}}, \mathcal{D}_{\text{curr}}\} = \{\mathcal{I}_i, \mathcal{D}_i\}$ , the *current observation*, is then passed to the following pipeline stages.

#### Noise Reduction

Flying pixels removal

First, outliers from the depth map  $\mathcal{D}_{\text{curr}}$  are removed by discarding pixels incompatible with their local neighborhood (*flying pixels*). A pixel  $\mathcal{D}_{\text{curr}}(x, y)$  is considered an inlier (i.e., not an outlier) if at least one pixel in its 4-neighborhood differs in depth by less than the tolerance  $\varepsilon_f = 0.1\text{m}$ .

Bilateral filtering

Subsequent bilateral filtering [TM98] of  $\mathcal{D}_{\text{curr}}$  mitigates noise, smoothing homogeneous regions while preserving depth discontinuities. As parameterization,  $\sigma_s = 2.5$  is used for the spatial Gaussian kernel and  $\sigma_r = 0.03$  for the range kernel. For noisy outdoor scenery,  $\sigma_r$  is increased to 0.15.

#### RGB-D Registration

RGB-D registration

If  $\mathcal{I}_{\text{curr}}$  and  $\mathcal{D}_{\text{curr}}$  are not pre-registered, both modalities are registered using the extrinsic transformation  $\mathbf{T}_{\mathcal{D} \rightarrow \mathcal{I}} = [\mathbf{R}_{\mathcal{D} \rightarrow \mathcal{I}}, \mathbf{t}_{\mathcal{D} \rightarrow \mathcal{I}}] \in \mathbb{SE}^3$  between both camera coordinate systems, with 3D rotation matrix  $\mathbf{R}_{\mathcal{D} \rightarrow \mathcal{I}} \in \mathbb{SO}^3$  and translation vector  $\mathbf{t}_{\mathcal{D} \rightarrow \mathcal{I}} \in \mathbb{R}^3$ . As high color resolution is prioritized, the proposed method breaks with the 3D reconstruction tradition of transforming color images into the viewpoint of the depth camera and, instead, projects depth  $\mathcal{D}_{\text{curr}}$  onto the color camera's image plane. While the former only requires a simple backward remapping operation on  $\mathcal{I}_{\text{curr}}$  for each pixel position  $(x, y)^T$  of  $\mathcal{D}_{\text{curr}}$  using its depth value  $\mathcal{D}_{\text{curr}}(x, y)$  (see Section 2.1.3), the latter is more complex: first,

$\mathcal{D}_{\text{curr}}$  is triangulated (see Section 4.4.4 for details), and then the resulting triangle mesh is rendered from the position and orientation of the color camera using  $\mathbf{T}_{\mathcal{D} \rightarrow I}$  and its intrinsic parameters, i.e., the principal point  $(c_x^I, c_y^I)^\top$  and the focal lengths  $f_x^I, f_y^I$ .

### 4.4.2 Camera Pose Estimation

To globally align the observation with the model  $\mathcal{M}$ , the current 6-DoF rigid camera transformation  $\mathbf{T}_{\text{curr} \rightarrow \mathcal{M}} = [\mathbf{R}, \mathbf{t}] \in \mathbb{SE}^3$ , with  $\mathbf{T}_{\text{curr} \leftarrow \mathcal{M}} = \mathbf{T}_{\text{curr} \rightarrow \mathcal{M}}^{-1}$ , needs to be estimated.

For this, 3D scene reconstruction approaches usually perform frame-to-model tracking by concatenating a chain of relative poses over all consecutive frames, which suffers from accumulating a temporal pose drift. This drift is the consequence of aligning the current frame with a proxy of the model, a rendering from the previous, already drift-affected pose. Instead, the proposed system benefits from the fact that refinement takes place in the reference pose, and the current frame is always aligned with the model itself. While the previous pose is also used as a prediction, it only serves as an initialization. This makes the system robust against self-localization drift, and it does not depend on loop closures to detect and correct error accumulation in a chain of relative poses.

Camera drift

The proposed pose estimation is based on a two-step, coarse-to-fine approach. First, the current frame is aligned with the “current” one of the previous pipeline run by searching for and matching sparse correspondences using scale-invariant, *speeded-up robust features* (SURF) [BTVG06]. A dense *iterative-closest-point* (ICP) algorithm [BM92, CM92] is then initialized with the resulting pre-alignment, estimating a final, fine-scale alignment  $\mathbf{T}_{\text{curr} \rightarrow \mathcal{M}}$  between the current frame and the model  $\mathcal{M}$ .

#### Pre-alignment Using Sparse Keypoints

As potentially large displacements are expected between the current and the reference pose (see Section 4.4.1), a coarse pre-alignment is estimated using sparse photometric correspondences. First, a set of photometric SURF features with 2D keypoint locations  $\mathcal{K}_{\text{curr}} \in \mathbb{R}^2$  are detected in the current color image  $\mathcal{I}_{\text{curr}}$ , using the Hessian feature threshold  $\varepsilon_h = 1000$  and four SURF octaves with four scales in each octave. These features are then matched against the feature descriptors of keypoints  $\mathcal{K}_{\text{prev}} \in \mathbb{R}^2$  of the previously selected “current” frame processed by the pipeline using RANSAC [FB81].

Local features

## Keypoint filtering

The keypoint sets  $\mathcal{K}_{\text{curr}}$  and  $\mathcal{K}_{\text{prev}}$  are pruned by filtering potential mismatches and error-prone keypoints. Here, keypoint matches are pre-filtered by applying Lowe’s ratio test [Low04]. A keypoint is tested for its integrity by comparing its two best matches using their distance ratio. If both matches are similarly rated, the keypoint is discarded, with the intuition that a correct match is unique. As a ratio threshold,  $\varepsilon_r = 0.675$  is used.

Additionally, keypoints  $\mathcal{K}_{\text{curr}}$  are filtered in the vicinity of unreliable depths. While 2D keypoint locations are based on high-resolution color imagery, their 3D locations rely on coarse depth maps, which is highly prone to error at inaccurate depth discontinuities and surfaces with a flat angle to the camera. Therefore, a binary mask  $\mathcal{G} \in \mathbb{Z}_2$  of inhomogeneous areas is computed: first, morphologically eroded and dilated depth map versions  $\mathcal{D}_{\text{curr}}^{\text{min}}$  and  $\mathcal{D}_{\text{curr}}^{\text{max}}$  are generated, using a  $5 \times 5$  box-shaped structuring element. Then, by thresholding  $\mathcal{D}_{\text{curr}}^{\text{diff}} = \mathcal{D}_{\text{curr}}^{\text{max}} - \mathcal{D}_{\text{curr}}^{\text{min}}$ , pixels are excluded with differences that exceed  $\varepsilon_d = 0.03\text{m}$ .

Finally, 2D keypoint locations  $\mathcal{K}_{\text{curr}}$  are back-projected using their corresponding depths in  $\mathcal{D}_{\text{curr}}$  and the input camera’s intrinsic matrix  $\mathbf{K}_{\text{curr}} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$  to get the 3D point set

$$\mathcal{P}_{\text{curr}} = \mathcal{D}_{\text{curr}}(\mathcal{K}_{\text{curr}}) \mathbf{K}_{\text{curr}}^{-1} (\mathcal{K}_{\text{curr}}, 1)^\top \in \mathbb{R}^3. \quad (4.1)$$

Knowing the correspondences between  $\mathcal{P}_{\text{curr}}$  and  $\mathcal{P}_{\text{prev}}$  by the feature matching process, a rigid transformation  $\mathbf{T}_{\text{curr} \rightarrow \text{prev}}$  can be computed by minimizing the MMSE [Ume91]. This results in the coarse pre-alignment  $\mathbf{T}_{\text{curr} \rightarrow \mathcal{M}}^{\text{pre}} = \mathbf{T}_{\text{curr} \rightarrow \text{prev}} \circ \mathbf{T}_{\text{prev} \rightarrow \mathcal{M}}$ , using the previous pose estimation.

## Final Alignment Using Dense Correspondences

For the final transformation  $\mathbf{T}_{\text{curr} \rightarrow \mathcal{M}}$ , the current frame is directly aligned with the model  $\mathcal{M}$  itself on a dense, fine-scale basis using the pre-alignment  $\mathbf{T}_{\text{curr} \rightarrow \mathcal{M}}^{\text{pre}}$  as initialization. This is done by performing a dense Colored ICP [PZK17], which is summarized in the following: Colored ICP optimizes for photometric consistency in addition to geometric consistency, which is formulated as the joint objective

## Colored ICP

$$E_{\text{hybrid}} = (1 - \sigma_{\text{ICP}})E_{\mathcal{I}} + \sigma_{\text{ICP}}E_{\mathcal{D}}, \quad (4.2)$$

with  $E_{\mathcal{I}}$  and  $E_{\mathcal{D}}$  being the photometric and geometric least-squares objectives. As in Park et al. [PZK17],  $\sigma_{\text{ICP}} = 0.968$  is set.  $E_{\mathcal{D}}$  is formulated as the traditional point-to-plane error metric,

$$E_{\mathcal{D}}(\mathbf{T}_{\text{curr} \rightarrow \mathcal{M}}) = \sum_{(p,q) \in \mathcal{R}} \langle \mathbf{T}_{\text{curr} \rightarrow \mathcal{M}} \mathcal{V}_{\text{curr}}(\mathbf{q}) - \mathcal{V}_{\mathcal{M}}(\mathbf{p}), \mathcal{N}_{\mathcal{M}}(\mathbf{p}) \rangle^2, \quad (4.3)$$

between the current input depth map  $\mathcal{D}_{\text{curr}}$  and model depth  $\mathcal{D}_{\mathcal{M}}$ , back-projected to camera space, i.e.,  $\mathcal{V}_{\text{curr}}$  and  $\mathcal{V}_{\mathcal{M}}$  (see Section 4.4.4). The model's normal map  $\mathcal{N}_{\mathcal{M}}$  is determined from central-differences of  $\mathcal{V}_{\mathcal{M}}$ .

The photometric objective  $E_I$  is expressed as the squared differences of intensities

$$E_I(\mathbf{T}_{\text{curr} \rightarrow \mathcal{M}}) = \sum_{(p,q) \in \mathcal{R}} \left( \mathcal{I}_{\text{curr}}(\mathbf{q}) - \mathcal{I}_{\mathcal{M}}^{\text{comp}}(\mathbf{p}) \right)^2, \quad (4.4)$$

between the current input color image  $\mathcal{I}_{\text{curr}}$  and  $\mathcal{I}_{\mathcal{M}}^{\text{comp}}$ , which is the re-composed model color image from the Laplacian pyramid  $\mathcal{I}_{\mathcal{M}}$ .

The dense correspondence set  $\mathcal{R} = \{(\mathbf{p}, \mathbf{q})\}$  is determined via projective data association, that is, projecting each pixel in  $\mathcal{D}_{\text{curr}}$  with location  $\mathbf{q} \in \mathbb{N}^2$  onto  $\mathcal{D}_{\mathcal{M}}$ , getting the corresponding pixel location

$$\mathbf{p} = \pi \left( \mathbf{K}_{\mathcal{M}} \mathbf{T}_{\text{curr} \rightarrow \mathcal{M}} \underbrace{\mathcal{D}_{\text{curr}}(\mathbf{q}) \mathbf{K}_{\text{curr}}^{-1}(\mathbf{q}, 1)^{\top}}_{\text{back-projection}} \right) \in \mathbb{R}^2, \quad (4.5)$$

with  $\mathbf{K}_{\text{curr}}$  and  $\mathbf{K}_{\mathcal{M}}$  being the camera's intrinsic matrices of the current frame and the model, and  $\pi(x, y, z) = (x/z, y/z)^{\top}$ , the de-homogenization. To prune potential correspondences, the Euclidean distance threshold  $\varepsilon_{\text{dist}}$  and angle threshold  $\varepsilon_{\text{angle}} = 45^\circ$  are used as compatibility criteria. Here,  $\varepsilon_{\text{dist}} = \{0.1\text{m}, 0.065\text{m}, 0.03\text{m}\}$  is set for a three-level coarse-to-fine ICP, and the criterion is softened for noisy outdoor footage to  $\varepsilon_{\text{dist}} = \{0.3\text{m}, 0.165\text{m}, 0.03\text{m}\}$ .

Projective data association

### 4.4.3 Model Correspondence

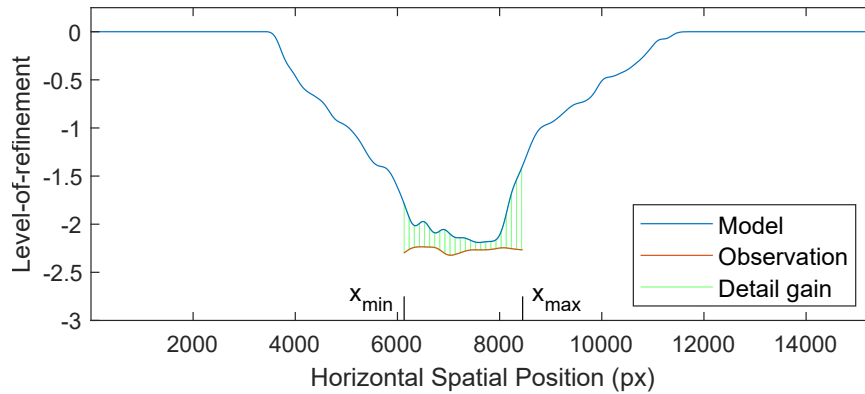
The correspondence between the observation and the model refers to the *region of interest* in the model  $\mathcal{M}$  affected by the current frame, and the observation's *level of refinement*, representing the observation's potential to refine  $\mathcal{M}$ . Figure 4.3 illustrates these properties with an example.

The current region of interest  $roi = (x_{\min}, x_{\max}, y_{\min}, y_{\max})$ , i.e., the observation's lateral boundaries within model  $\mathcal{M}$ , is calculated by the forward projection of  $\mathcal{D}_{\text{curr}}$  onto  $\mathcal{D}_{\mathcal{M}}$  using Equation (4.5).

Region of interest

The observation's level of refinement refers to the spatial sampling rate that is inverse-proportional to distance, i.e., the sampling rate increases the closer the camera is moved to the scene compared to the reference viewpoint. Thus, the level-of-refinement map  $\mathcal{L}_{\text{curr}} \in \mathbb{R}$  is determined as the corresponding pyramid level in  $\mathcal{M}$  per pixel. By back-projecting and transforming each model depth of  $roi(\mathcal{D}_{\mathcal{M}})$  into the camera space of the current observation, its distance to the current frame's camera plane is obtained by extracting its  $z$ -component.

Level of refinement



**Figure 4.3:** A one-dimensional graphic representation of the level-of-refinement maps  $\mathcal{L}_{\mathcal{M}}$  and  $\mathcal{L}_{\text{curr}}$ . The model’s accumulated level of refinement (blue) is shown after the camera has been moved centrally towards the scene, with details accumulated up to level -2.2. The current observation (red) offers a higher object-space resolution (lower corresp. level), where the per-pixel gain in visual detail is colored in green (see  $\Delta_{\text{curr}}$  in Equation (4.12)). Its lateral boundaries within the model are  $x_{\min}$  and  $x_{\max}$  (region of interest), the minimum pyramid level is  $l_{\min} = -3$ .

The scale factor between both depths is then mapped to a pyramid level index, where the sampling rate for each level increases by one octave. The (fractional) number of octaves between both distances is given by

$$\mathcal{L}_{\text{curr}}(x, y) = \log_2 \frac{\left( \mathbf{T}_{\text{curr} \leftarrow \mathcal{M}} \mathcal{D}_{\mathcal{M}}(x, y) \mathbf{K}_{\mathcal{M}}^{-1}(x, y, 1)^{\top} \right)_z}{\mathcal{D}_{\mathcal{M}}(x, y)} \in \mathbb{R}, \quad (4.6)$$

where  $(\cdot)_z$  is the z-component of a 3D point. For this estimate, the accumulated model depths are used, as they are more accurate, complete, and reliable than observation depths. Here, a gain in level of refinement, i.e.,  $\mathcal{L}_{\text{curr}}(x, y) \leq \mathcal{L}_{\mathcal{M}}(x, y)$ , indicates the observation’s ability to contribute superior information for refining the model by updating its data in the fusion stage (Section 4.4.6).

Finest corresponding level

Furthermore, the overall minimum pyramid level index  $l_{\min} = \lfloor \min(\mathcal{L}_{\text{curr}}) \rfloor \in \mathbb{Z}$  is determined. If this level is beyond the current level boundaries of  $\mathcal{M}$ , the model is expanded as follows: a new level of unallocated tiles is appended to the bottom of Laplacian pyramid  $\mathcal{I}_{\mathcal{M}}$ . For the sparsely occupied Gaussian pyramid  $\mathcal{D}_{\mathcal{M}}$ , all tiles affected by the region of interest are up-sampled to  $l_{\min}$ , using nearest-neighbor interpolation to avoid introducing flying pixels. The model’s counters  $c_{\mathcal{M}}$ ,  $v_{\mathcal{M}}$  and the accumulated level-of-refinement  $\mathcal{L}_{\mathcal{M}}$  inherit their values from coarser levels on demand, as needed during fusion.



#### 4.4.4 Parallax-Aware Warping

##### Color Warping

To allow a fusion with the model, a perspective warping of the color image  $\mathcal{I}_{\text{curr}}$  into the model's image space is performed. In contrast to *progressive refinement imaging* for planar scenes (see Chapter 3), which estimates a homography by assuming a (quasi) planar scenery, the proposed method for 3D scenes now has to rely on depth values for a disparity-corrective mapping between both image spaces. Therefore, the pixel mapping  $\mathcal{W}_{\mathcal{M} \rightarrow \text{curr}} \in \mathbb{R}^2$  that relates model to observation locations is calculated:

Disparity-corrected  
color warping

$$\mathcal{W}_{\mathcal{M} \rightarrow \text{curr}}(x, y) = \pi \left( \mathbf{K}_{\text{curr}} \mathbf{T}_{\text{curr} \leftarrow \mathcal{M}} \mathcal{D}_{\mathcal{M}}(x, y) \mathbf{K}_{\mathcal{M}}^{-1}(x, y, 1)^{\top} \right), \quad (4.7)$$

with  $(x, y) \in [x_{\min}, \dots, x_{\max}] \times [y_{\min}, \dots, y_{\max}]$ . That is, each regular lattice grid position within  $\text{roi}(\mathcal{M})$  is mapped to an irregular sub-pixel coordinate in the current frame using refined model depths  $\mathcal{D}_{\mathcal{M}}$  and camera transformation  $\mathbf{T}_{\text{curr} \leftarrow \mathcal{M}}$ .

The color image  $\mathcal{I}_{\text{curr}}$  is then warped to  $\mathcal{M}$  using a backward remapping  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}(x, y) = \mathcal{I}_{\text{curr}}(\mathcal{W}_{\mathcal{M} \rightarrow \text{curr}}(x, y))$ , i.e., a resampling of  $\mathcal{I}_{\text{curr}}$  at sub-pixel positions  $\mathcal{W}_{\mathcal{M} \rightarrow \text{curr}}$  using bi-linear interpolation. As color will be fused using Laplacian pyramids (see Section 4.4.6),  $\mathcal{I}_{\text{curr}}$  is warped to the finest corresponding model level  $\mathcal{M}^{l=l_{\min}}$  at level index  $l_{\min}$ .

Resampling

Finally, by subtly smoothing  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$  at depth discontinuities, a natural color transition between foreground and background objects is obtained instead of a binary one. For that, a Gaussian kernel with radius  $r_{\text{G}} = 2\text{px}$  is used.

Note that warping the 2D image  $\mathcal{I}_{\text{curr}}$  inevitably leads to inconsistencies with the model in occluded regions, which are addressed in the outlier removal stage (Section 4.4.5).

##### Depth Warping

Changing the perspective of a 2.5-D depth map requires retrieving the underlying 3D geometry represented by the discretized range values. Therefore,  $\mathcal{D}_{\text{curr}}$  is converted to a polygon mesh by computing a vertex map  $\mathcal{V}_{\text{curr}}(x, y) = \mathcal{D}_{\text{curr}}(x, y) \mathbf{K}_{\text{curr}}^{-1}(x, y, 1)^{\top}$ , and then, neighboring vertices  $\mathcal{V}_{\text{curr}}(x, y)^{\top}$ ,  $\mathcal{V}_{\text{curr}}(x + 1, y)^{\top}$ ,  $\mathcal{V}_{\text{curr}}(x, y + 1)^{\top}$ , and  $\mathcal{V}_{\text{curr}}(x + 1, y + 1)^{\top}$  are triangulated by choosing the diagonal with the shorter length. To open the mesh at discontinuities, triangles with edges longer than  $\varepsilon_{\text{d}} = 0.03\text{m}$  are omitted. Finally, the warped depth map  $\mathcal{D}_{\text{curr} \rightarrow \mathcal{M}}$  is obtained by rendering the mesh as seen from the model's

Depth map warping

Triangulation

camera, by setting the view matrix to  $\mathbf{T}_{\text{curr} \leftarrow \mathcal{M}}$  and the viewport to  $roi$ , with the resolution of level  $l_{\min}$ .

#### 4.4.5 Local Color Consistency

After aiming for global consistency in the camera alignment stage (Section 4.4.2), local consistency is now sought as the warped observation and the model share the same image space. This is done by matching the warped input frame  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$  to the reference  $\mathcal{M}$  on a per-pixel basis, using a two-step approach: first,  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$  is re-aligned locally by estimating a per-pixel displacement w.r.t.  $\mathcal{M}$ . Second, pixels that are still inconsistent with the model are classified as outliers.

##### Local Re-alignment

Dense optical flow

Based on *progressive refinement imaging* for planar scenes, a dense *Optical Flow* is computed between grayscale variants of  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$  and  $\mathcal{I}_{\mathcal{M}}^{\text{comp}}$  using [Far03]. The resulting flow field  $\mathcal{F}_{\mathcal{M} \rightarrow \text{curr}}$  provides the sub-pixel lateral motion to reduce local misalignments.

To account for various input scales, an adaptive number of scale levels is used for the optical flow algorithm, that is, the number of pyramid levels between model levels  $l = 0$  and  $l_{\min}$  is used (i.e.,  $-l_{\min}$ ).  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$  is then re-aligned w.r.t.  $\mathcal{M}$  by applying the backward flow of  $\mathcal{F}_{\mathcal{M} \rightarrow \text{curr}}$ .

##### Local Outlier Removal

To avoid merging inconsistent color data, the warped color frame is searched pixel-wise for geometric discrepancies to detect mismatches that could not be re-aligned or regions that cannot be incorporated, e.g., due to occlusion. Similar to Chapter 3, outliers are detected on band-pass filtered Laplacian levels, while explicitly omitting the top (Gaussian) level  $l = 0$  in order to be resilient to photometric deviations due to local illumination changes. However, a different outlier classification scheme is proposed, as described in the following.

Outlier classification

In the outlier removal stage, the main challenge is to correctly classify novel details as inliers, even if they create discrepancies with the model. By comparing a Laplacian decomposition of the warped frame,  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l$ , with the Laplacian model pyramid  $\mathcal{I}_{\mathcal{M}}^l$ , it is possible to exploit that true outliers are geometrically inconsistent across all levels, whereas novel details are in a mismatch on the finest level(s) only (see Figure 4.4). Thus, a per-pixel *similarity*

score  $s_{\text{curr}}^l \in \mathbb{R}$  w.r.t.  $\mathcal{M}$  is determined separately for each Laplacian level  $l < 0$ , Similarity score starting with the coarsest Laplacian level  $l = -1$ :

$$s_{\text{curr}}^{l=-1} = SSIM^{C,S} \left( \mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^{l=-1}, \mathcal{I}_{\mathcal{M}}^{l=-1} \right), \quad (4.8)$$

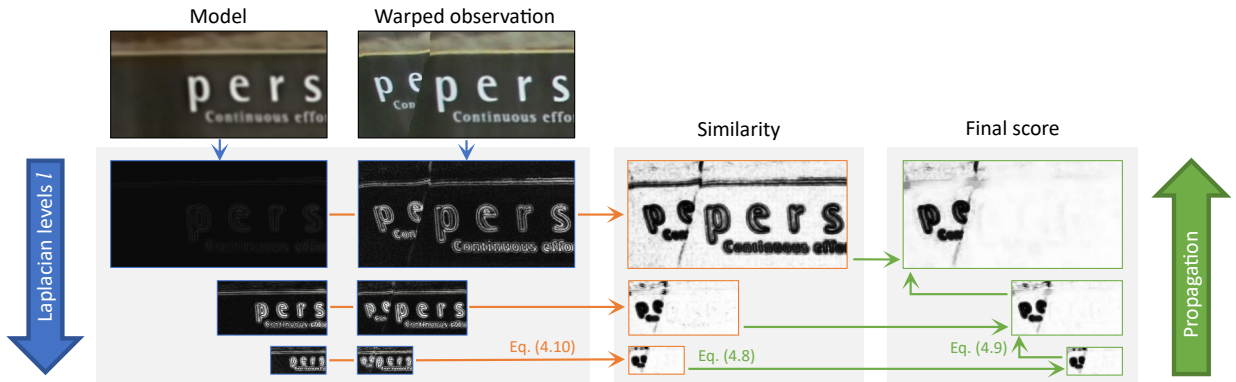
where  $SSIM^{C,S} \in \mathbb{R}$  is the similarity metric given in Equation (4.10).

Since outliers are only distinguishable from novel details on coarser levels, where these frequencies are already present in the model, the similarity score is then propagated to the finest level  $l = l_{\text{min}}$  by retaining high similarities from the coarser levels: Coarse-to-fine propagation

$$s_{\text{curr}}^l = \max \left( \underbrace{SSIM^{C,S} \left( \mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l, \mathcal{I}_{\mathcal{M}}^l \right)}_{\text{level } l}, \underbrace{[s_{\text{curr}}^{l+1}]_{\uparrow 2}}_{\text{level } l+1} \right), \quad (4.9)$$

where  $[\dots]_{\uparrow 2}$  indicates an up-sampling by one octave. Figure 4.4 illustrates this scheme, showing the computation of the similarity score and the effect of the proposed propagation strategy.

As similarity metric  $SSIM^{C,S}$ , a variant of  $SSIM$  [WBSS04] suitable for being Similarity metric applied to Laplacian images is used. The original  $SSIM$  offers a *structural similarity index measure* between two intensity images  $X \in \mathbb{R}$  and  $Y \in \mathbb{R}$ , with  $SSIM \in [-1, +1]$  and can be broken down into three independent components: a comparison for luminance, contrast, and structure. Since the metric is applied



**Figure 4.4:** The proposed outlier removal scheme. Between the model  $\mathcal{I}_{\mathcal{M}}^l$  and the warped observation  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l$ , the *similarity* is determined for each Laplacian level  $l$  ( $SSIM^{C,S}$  in Equation (4.10)). The information is then propagated upwards to compute the *final similarity score* ( $s_{\text{curr}}^l$  in Equation (4.9)). Novel details not yet in the model are in a mismatch on the finest level but are correctly classified as inliers.

on Laplacian levels, the luminance component is discarded, resulting in

$$SSIM^{C,S}(X, Y) = \max\left(\underbrace{\left[\frac{2\sigma_X\sigma_Y}{\sigma_X^2 + \sigma_Y^2}\right]^\beta}_{\text{contrast}} \underbrace{\left[\frac{\sigma_{XY}}{\sigma_X\sigma_Y}\right]^\gamma}_{\text{structure}}, 0\right), \quad (4.10)$$

comprising the product of contrast and structure similarity.  $\sigma_X$ ,  $\sigma_Y$  are the standard deviations of  $X$  and  $Y$  within a local window;  $\sigma_X^2$ ,  $\sigma_Y^2$  the variances; and  $\sigma_{XY}$  the covariance. The weighting parameters are set to  $\beta = 1$ ,  $\gamma = 1$  and the result is clamped to ensure  $SSIM^{C,S} \in [0, 1]$ .

While the contrast comparison serves a similar purpose as the error metric described in Chapter 3, it additionally compares the local structure instead of individual pixels. The local window size is set adaptively and increases according to finer pyramid levels  $l$ , starting with radius  $r_o = 1$ px.

Finally, pixels  $(x, y)$  on levels  $l$  are classified as outliers if their similarity score  $s_{\text{curr}}^l(x, y)$  falls below  $\varepsilon_o = 0.15$ . In the following fusion stage,  $s_{\text{curr}}^l \in [0, 1]$  is further used to weight inliers according to their achieved score (see Equation (4.11)).

#### 4.4.6 Fusion

In the final stage of the pipeline, the current frame  $\{\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}, \mathcal{D}_{\text{curr} \rightarrow \mathcal{M}}\}$  is fused with the current model  $\{\mathcal{I}_{\mathcal{M}}, \mathcal{D}_{\mathcal{M}}\}$ .

##### Color Fusion

Frequency-oriented  
color fusion

Conceptually, the frequency-oriented color fusion approach is based on *progressive refinement imaging* for planar scenes. That is, the Laplacian levels of the color pyramids are merged while the base color of the Gaussian level is retained and, thus, progressive refinement is enabled without requiring local or global optimization for color harmonization. However, the proposed approach designed for fusing warped observations of 3D scenes requires a different accumulation scheme.

The accuracy and reliability of the warped color  $\mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}$  are primarily limited by the underlying depth data due to inaccurate or even false depth estimates captured at low(er) resolution. Thus, in contrast to the method described in Chapter 3, which is based on a planar scene and a replacement strategy, a blending scheme of multiple observations is required, as each single, warped observation is not reliable enough by itself.

To prevent coarser observations from degrading the model, only inlier pixels  $(x, y)$  with a finer level of refinement are fused, i.e., if  $\mathcal{L}_{\text{curr}}^l(x, y) \leq \mathcal{L}_{\mathcal{M}}^l(x, y)$ . The proposed blending scheme then applies

$$\mathcal{I}_{\mathcal{M}}^l \leftarrow \frac{\mathcal{I}_{\mathcal{M}}^l + w_{\text{curr}} s_{\text{curr}}^l \mathcal{I}_{\text{curr} \rightarrow \mathcal{M}}^l}{1 + w_{\text{curr}} s_{\text{curr}}^l}, \quad (4.11)$$

Color blending  
scheme

to levels  $l \in [l_{\min}, \dots, -1]$ , thus updating all corresponding Laplacian levels with data from the new observation. Here,  $s_{\text{curr}}^l \in [0, 1]$  is the score determined in Section 4.4.5, which is used to lower the contribution of less reliable input color. Apart from that, the weight  $w_{\text{curr}}$  applied to the observation is computed as

$$w_{\text{curr}} = \underbrace{\Delta_{\text{curr}}}_{\text{gain}} + \underbrace{\frac{1}{c_{\mathcal{M}}^l}}_{\text{counter}}, \quad (4.12)$$

Detail gain-based  
weighting

$$\text{with } \Delta_{\text{curr}} = \min \left( \left| \mathcal{L}_{\text{curr}}^l - \mathcal{L}_{\mathcal{M}}^l \right|, \Delta_{\max} \right),$$

where  $\Delta_{\text{curr}}$  represents the gain in level of refinement (colored green in Figure 4.3),  $c_{\mathcal{M}}^l$  is the model's counter, and  $\mathcal{L}_{\text{curr}}^l$  is the Gaussian decomposition of  $\mathcal{L}_{\text{curr}}$ . To reflect the amount of detail blended into the model so far, the model's level of refinement,  $\mathcal{L}_{\mathcal{M}}^l$ , is updated analogously to Equation (4.11) as a weighted average, using

$$\mathcal{L}_{\mathcal{M}}^l \leftarrow \frac{\mathcal{L}_{\mathcal{M}}^l + w_{\text{curr}} s_{\text{curr}}^l \mathcal{L}_{\text{curr}}^l}{1 + w_{\text{curr}} s_{\text{curr}}^l}, \quad (4.13)$$

while the counter is incremented by

$$c_{\mathcal{M}}^l \leftarrow c_{\mathcal{M}}^l + 1. \quad (4.14)$$

With  $\Delta_{\text{curr}} = 0$ , Equation (4.12) reduces to the basic blending scheme in incremental scene reconstruction, a *cumulative average* of samples [NIH\*11, CL96], i.e., the observation's weight  $w = 1/c_{\mathcal{M}}$  is decreasing continuously as the model's counter  $c_{\mathcal{M}} \in [1, \dots, \infty]$  is incremented with each observation. In refinement imaging, this averaging scheme potentially prevents details captured by later observations from getting into the model (see Section 3.5.4). This happens specifically when many (early) observations with less details force up the weight. The proposed approach, therefore, takes the gain in level of refinement  $\left| \mathcal{L}_{\text{curr}}^l - \mathcal{L}_{\mathcal{M}}^l \right|$  into account and combines it with the traditional confidence counter, defined by the number of observations ( $1/c_{\mathcal{M}}$ ). To limit the maximum contribution of a single observation and, thus, to prevent the model from being replaced,  $\Delta_{\text{curr}}$  is clamped at  $\Delta_{\max} = 0.1$ .

## Depth Fusion

**Depth map fusion** The imperfect nature of depth images requires a different way of fusion, as no reliable initial reference depth is available, which could be used for (additive) refinement. Instead, inaccurate depths need to be corrected and false values have to be detected and replaced.

To filter observation depths  $\mathcal{D}_{\text{curr} \rightarrow \mathcal{M}}$  that are incompatible with the model  $\mathcal{D}_{\mathcal{M}}$ , the depth tolerance threshold  $|\mathcal{D}_{\mathcal{M}} - \mathcal{D}_{\text{curr} \rightarrow \mathcal{M}}| \leq \varepsilon_d$  is used as compatibility criterion. Compatible pixels are then blended on pyramid level  $l_{\min}$  by the weighted average

**Depth blending**

$$\mathcal{D}_{\mathcal{M}} \leftarrow \frac{v_{\mathcal{M}} \mathcal{D}_{\mathcal{M}} + \mathcal{D}_{\text{curr} \rightarrow \mathcal{M}}}{v_{\mathcal{M}} + 1}, \quad (4.15)$$

to improve the accuracy of model depths  $\mathcal{D}_{\mathcal{M}}$  over time. However, in the case of initializing  $\mathcal{D}_{\mathcal{M}}(x, y)$  with a false value, further observations will fail the compatibility test, inhibiting any refinement.

**Progressive voting scheme**

Therefore, an incremental voting strategy is proposed to find a suitable model value progressively (see Figure 4.5). With the intention that each new observation votes either for or against the reliability of a model pixel's depth,  $v_{\mathcal{M}} \in \mathbb{R}$  is interpreted as a *voting counter*. For each fusion that failed due to incompatibility with the model, the model pixel's counter is decreased, yielding the following counter update:

$$v_{\mathcal{M}} \leftarrow \begin{cases} v_{\mathcal{M}} - e^{-(v_{\mathcal{M}}/\sigma)^2}, & \text{if } |\mathcal{D}_{\mathcal{M}} - \mathcal{D}_{\text{curr} \rightarrow \mathcal{M}}| > \varepsilon_d, \\ v_{\mathcal{M}} + 1, & \text{otherwise.} \end{cases} \quad (4.16)$$

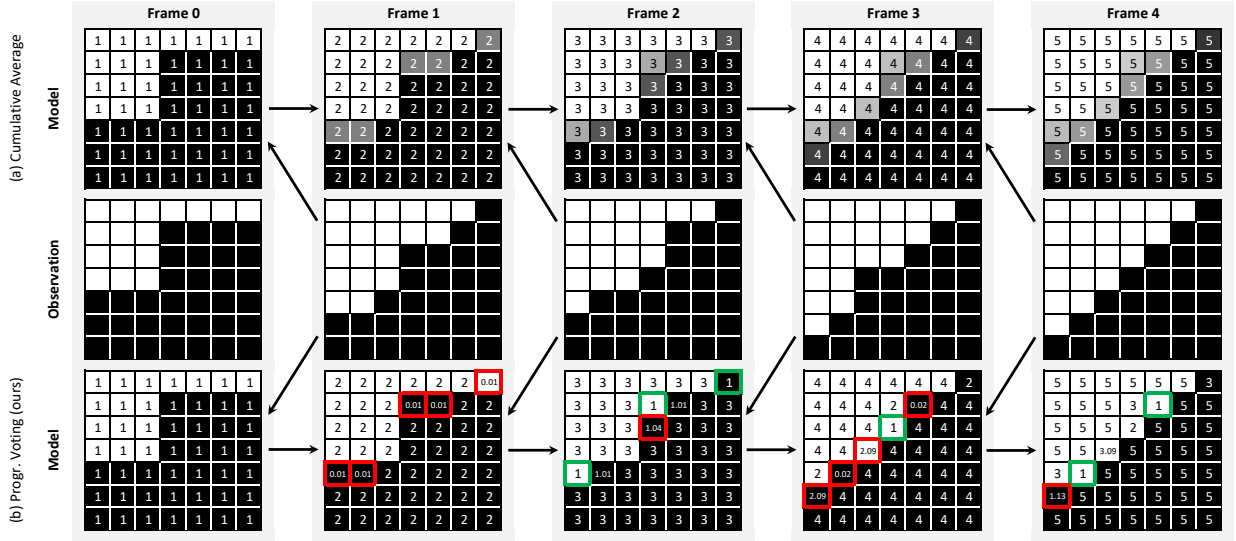
Here,  $e^{-(v_{\mathcal{M}}/\sigma)^2}$  is used to control the amount of decrease in case of an incompatible observation. This approach ensures a stable result once a model depth has been consolidated, while it quickly discards less reliable model values in favor of a more frequently observed depth value. For all experiments,  $\sigma$  is set to  $\sigma = 10$ .

**Depth replacement**

In case a pixel's voting counter falls below 0, i.e., if  $v_{\mathcal{M}} \leq 0$ , its depth value is replaced and the counter is reset:

$$\mathcal{D}_{\mathcal{M}} \leftarrow \mathcal{D}_{\text{curr} \rightarrow \mathcal{M}}, \quad v_{\mathcal{M}} \leftarrow 1. \quad (4.17)$$

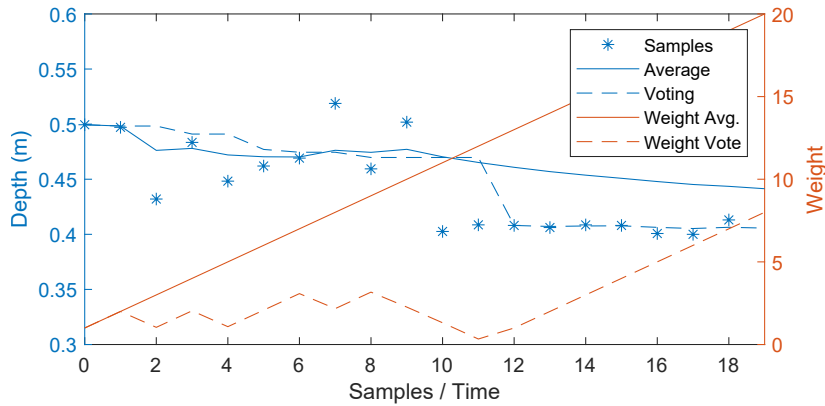
Figure 4.5 illustrates this voting scheme, showing the resulting fusion compared to a cumulative average. To demonstrate the effect of the resulting weighting, a one-dimensional visualization is shown in Figure 4.6.



**Figure 4.5:** Fusion of erroneous pixels at depth discontinuities for a set of example depth maps with foreground (black) and background depths (white). At frame 0, the model is initialized with the coarse observation depths (*left column*). At subsequent frames, the current observation (*middle row*) is fused with the model of the previous frame. (a) *Top row*: cumulative average, where the pixel’s counter is successively incremented. Blending of incompatible pixels results in flying pixels between the foreground and background depth. (b) *Bottom row*: using the proposed voting strategy, the model depths progressively approach the correct depths (shown in frame 4 for the observation) by replacing pixels at depth discontinuities. The pixel’s voting counter is decremented (Equation (4.16)) if the observation and its corresponding model depth are incompatible (highlighted in red); otherwise incremented. In case too many observations voted against a model pixel, i.e., a pixel’s voting counter becomes negative, its depth is replaced by the current observation (highlighted in green), and the counter is reset to 1 (Equation (4.17)).

#### 4.4.7 Final Output

After the final frame of the RGB-D input sequence has passed the pipeline stages described in Section 4.4.1 to Section 4.4.6, the model pyramids  $\mathcal{I}_M$  and  $\mathcal{D}_M$  are recomposed to produce the final refined RGB-D image  $\mathcal{I}_M^{\text{comp}}$  and  $\mathcal{D}_M^{\text{comp}}$  from the fixed viewpoint  $T_M$ . That is, the Laplacian color pyramid  $\mathcal{I}_M$  is recomposed by upsampling and summing all Laplacian levels  $\mathcal{I}_M^l$ . For the model depth  $\mathcal{D}_M$ , all tiles are sampled up to the finest pyramid level existing in the model  $\mathcal{M}$ . Finally, after combining all tiles to a full image,  $\{\mathcal{I}_M^{\text{comp}}, \mathcal{D}_M^{\text{comp}}\}$  is a refined version of the initial frame  $\{\mathcal{I}_0, \mathcal{D}_0\}$ , with a resolution up to a multiple of the initial resolution. Theoretically, by using the entire operating range of 8.0 m to 0.5 m for a typical RGB-D camera such as Kinect v2, the object-space resolution can be increased by a factor of 16, reaching several hundred



**Figure 4.6:** The proposed progressive voting scheme for depth fusion applied to a series of unreliable samples, followed by relatively stable samples of the real depth. While a *cumulative average* of samples slowly adapts to the new samples, *progressive voting* quickly discards less reliable data in favor of a compatible value by adjusting the weighting. If too many new samples fail the compatibility test, i.e., the weight (counter) falls below 0, the depth is set to the new sample and the weight is reset to 1.

megapixels for the final reconstruction (e.g., 530.8 MP when using a 2.1 MP image sensor). In the evaluation presented in Section 4.6, however, a scale factor of 6 to 10 was reached for the outdoor data sets.

## 4.5 Implementation

The reconstruction pipeline is implemented in C++, incorporating basic image processing operations from the OpenCV library. The pre-processing, outlier classification, and dense ICP are implemented on the GPU using CUDA. OpenCV’s SURF feature detection is used for the camera pre-alignment, whereas Farneback’s optical flow variant [Far03], provided by OpenCV, is used for local re-alignment. For rendering the input depth map from the reference pose, OpenGL is used by exploiting z-buffering. Lastly, the fusion of color and depth data is performed in image space using CUDA operations.

Although model color and depth share the same hierarchical structure (see Figure 4.2), they are stored separately in two sparsely occupied image pyramids, each with additional layers for the associated attribute maps (e.g., the counter). Each pyramid level comprises a 2D array of pointers referring to the allocated image tiles currently in use.



## 4.6 Results

### 4.6.1 Data Sets

Figure 4.7 shows the reference images of the eleven data sets used for evaluation. Besides the *Fountain* and the *LongOfficeHousehold* data sets, the following indoor as well as outdoor data sets are created that comprise medium to large disparities and, partially, very challenging situations in terms of reflective objects, fine scene details, and high noise levels (dark/black objects). For each data set,  $scale_{\max}$  denotes the maximum scale factor of object-space resolution with respect to the reference image that is featured by the input data.

*Fountain*: This outdoor scene, taken from Zhou and Koltun [ZK14], comprises a fountain with a specular tilework, where the camera is only slightly approaching the scene ( $scale_{\max} = 2.46$ ).

*LongOfficeHousehold*: This indoor data set, acquired by Sturm et al. [SEE\*12], shows an office with a 360° camera path around a table ( $scale_{\max} = 2.77$ ).

*CoffeeTable*: This indoor scene comprises highly reflective objects, e.g., a coffee machine and a black metal box ( $scale_{\max} = 4.38$ ).

*BooksGlobe*: An indoor scene that contains several books, a blanket, and a globe arranged on a couch/bed ( $scale_{\max} = 2.25$ ).

*VillageModel*: An indoor scene that comprises a set of model houses arranged on a table in front of a display screen. This scene comprises very small, dark, and mainly diffuse objects ( $scale_{\max} = 3.89$ ).

*BrickWall*: An outdoor scene with low depth variations that displays mainly diffuse stone colors ( $scale_{\max} = 6.96$ ).

*Memorial*: This outdoor data set comprises mainly diffuse objects with medium disparities ( $scale_{\max} = 9.71$ ).

*Statue*: An outdoor data set with statues at a fountain with large disparities and highly reflective water ( $scale_{\max} = 5.70$ ).

*Cannon*: This outdoor data set contains a cannon (glossy, black) and large disparities ( $scale_{\max} = 6.23$ ).

*FlowerBed*: An outdoor scene of a flowerbed with very unreliable depth data due to semi-transparent leaves and very fine details ( $scale_{\max} = 6.10$ ).

*BunnySynth*: This synthetic scene was generated using [LHK15], where the camera approaches a figurine of a bunny, a magazine, and other objects arranged on a table ( $scale_{\max} = 5.28$ ).

Fig. 4.8 shows the final level-of-refinement maps for each data set to visualize the amount of detail incorporated into the final reconstructions using the proposed method. Table 4.2 summarizes the main data set specifications. Each reconstruction is displayed from the initial pose, with a challenging sub-region shown as zoomed-in insets.

[Overview of data sets](#)

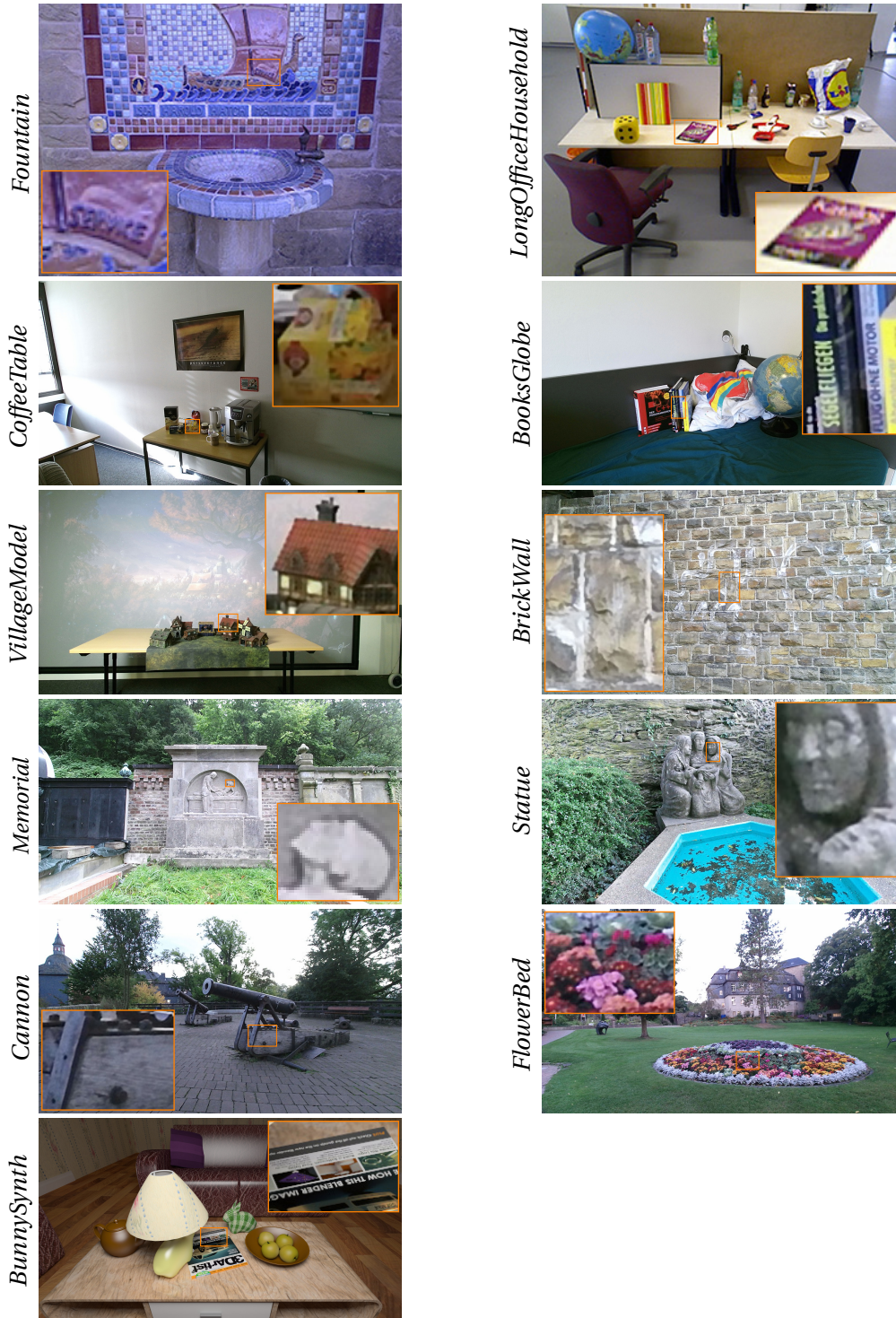
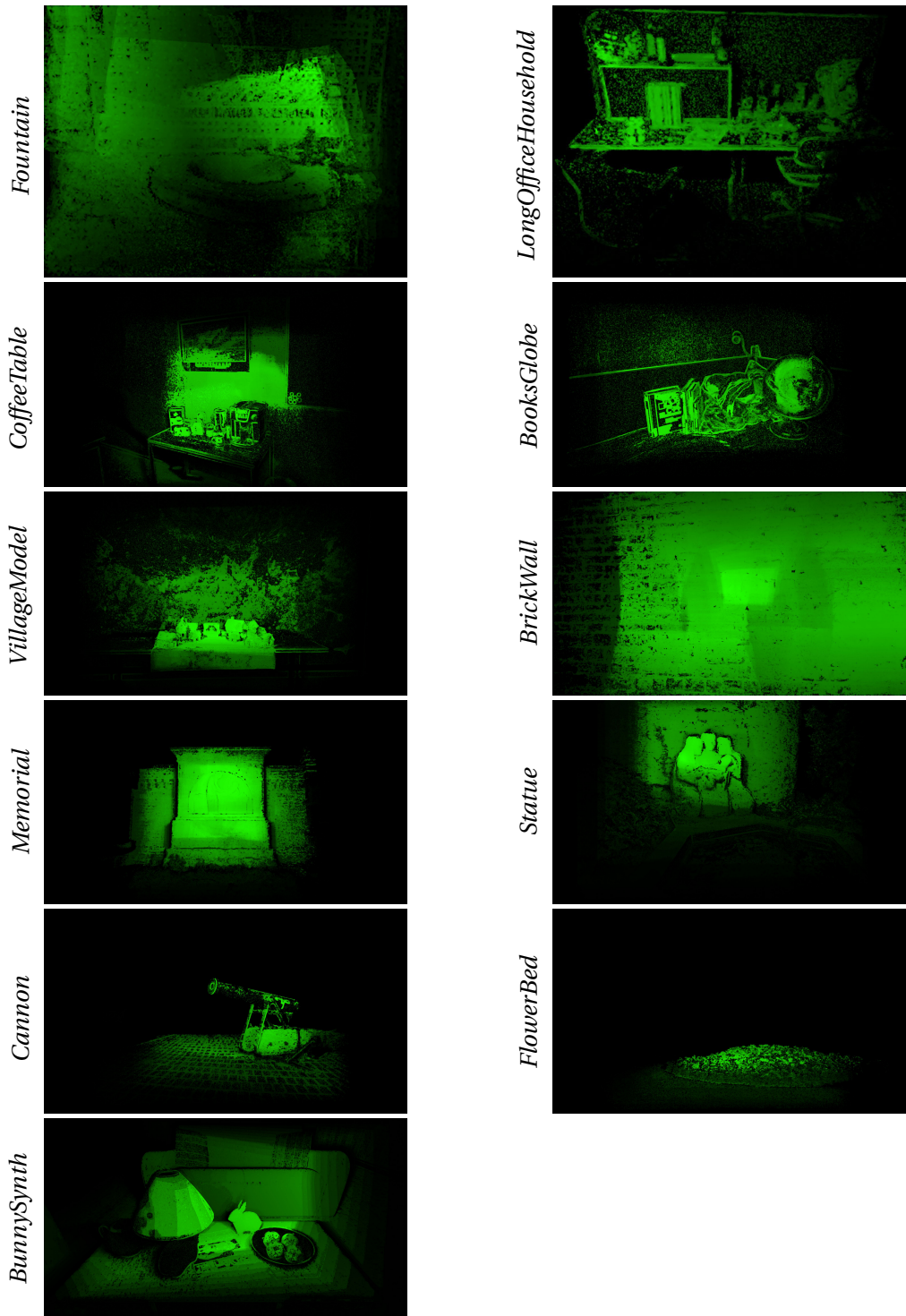


Figure 4.7: The unrefined reference images (initial frames) of the data sets.



**Figure 4.8:** The final level-of-refinement maps, visualizing the amount of detail incorporated into the final reconstruction using the proposed method. During the refinement, the current level-of-refinement map can be visualized to guide the user to areas needing more refinement. Brighter colors indicate a higher amount of incorporated details.

**Table 4.2:** Data set specifications. The data sets are acquired using the Asus Xtion Pro Live (pre-registered RGB-D:  $640 \times 480$  px) and the Kinect v2 (pre-registered RGB-D:  $1920 \times 1080$  px), comprising ‘# frames’ frames, where ‘# fused frames’ frames are selected by the specific method to be fused into the final result. The synthetic data set *BunnySynth* is rendered pre-registered with ground truth depths.

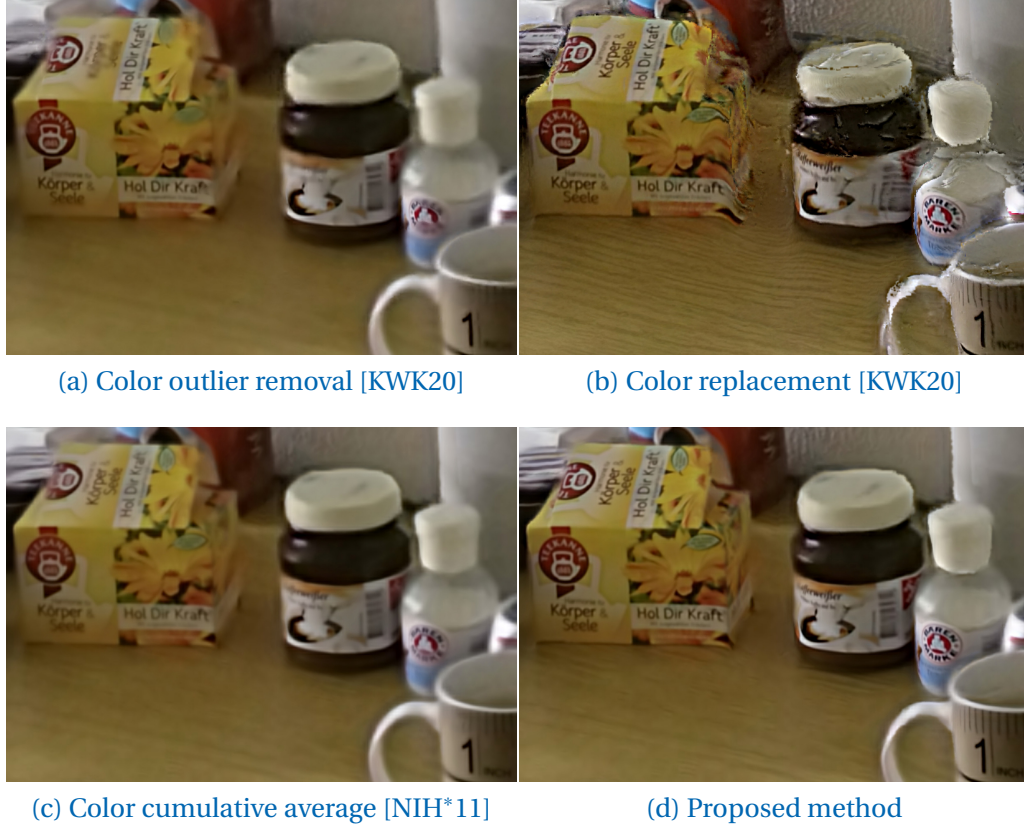
	Resolution	# frames	# fused frames			
			<i>Kluge20</i>	<i>Fu21</i>	<i>Niessner13, Kluge23</i>	<i>Lee20, Ha21</i>
<i>Fountain</i>	$640 \times 480$	1086	-	36	1086	59
<i>LongOfficeH.</i>	$640 \times 480$	2488	-	-	-	31
<i>CoffeeTable</i>	$1920 \times 1080$	2778	-	28	2778	186
<i>BooksGlobe</i>	$1920 \times 1080$	370	-	5	370	26
<i>VillageModel</i>	$1920 \times 1080$	2472	-	-	2472	162
<i>BrickWall</i>	$1920 \times 1080$	7420	498	-	7420	496
<i>Memorial</i>	$1920 \times 1080$	4037	356	-	4037	266
<i>Statue</i>	$1920 \times 1080$	1515	-	-	1515	96
<i>Cannon</i>	$1920 \times 1080$	677	-	-	677	43
<i>FlowerBed</i>	$1920 \times 1080$	728	-	-	728	48
<i>BunnySynth</i>	$1920 \times 1080$	190	-	4	190	15

## 4.6.2 Ablation Study

In this section, the performance of *progressive refinement imaging with depth-assisted disparity correction* is evaluated by replacing core concepts of the proposed pipeline with earlier approaches. The resulting effects are shown in Figure 4.9 for the *CoffeeTable* and in Figure 4.10 for the *VillageModel* data set.

### Outlier Classification Scheme

Figure 4.9a shows the outlier removal to achieve local color consistency as described in the previous chapter in Section 3.4.3, referred to as *Kluge20* [KWK20]. Figure 4.9d depicts the result when applying the *SSIM*-based Laplacian scheme proposed in this chapter (see Section 4.4.5). The result obtained with the *SSIM*-based outlier removal scheme yields further color refinement, specifically at object borders with less reliable warped color information, avoiding misclassifying novel details as outliers.



**Figure 4.9:** Ablation study for color reconstruction. (a) The proposed approach combined with the Laplacian outlier removal scheme from *Kluge20* [KWK20], as described in Chapter 3. (b) The proposed approach combined with the Laplacian color replacement strategy for color fusion from *Kluge20* [KWK20]. (c) The proposed approach combined with the conventional cumulative average weighting, e.g., [NIH\*11]. (d) The proposed approach with the presented *SSIM*-based outlier removal scheme and the presented color blending with detail gain-based weighting.

### Accumulation Strategy for Color Fusion

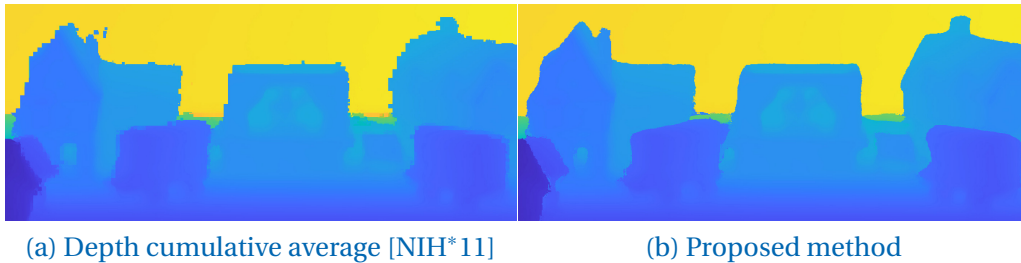
In Figure 4.9b, the pyramidal color replacement strategy as described in Section 3.4.4 for planar scenes is shown, while Figure 4.9d depicts the result obtained by the proposed blending method presented in Section 4.4.6. Comparing both results, it can be seen that the replacement scheme leads to strong artifacts at object boundaries and other areas with unreliable depth data, causing the reconstruction to suffer from noise and distorted colors. In contrast, the blending approach results in a geometrically and photometrically consistent reconstruction.

### Weighting Scheme for Color Fusion

Figure 4.9c shows the color fusion result using a conventional cumulative averaging scheme used by, for instance, Newcombe et al. [NIH\*11], and Figure 4.9d gives the result when applying the proposed approach that takes the gain of visual detail into account, as described in Section 4.4.6. It can be observed that the classical weighting scheme is not able to incorporate as much color detail as the proposed weighting scheme, leading to a more blurred result.

### Voting Strategy for Depth Fusion

The effect of using the novel voting scheme for depth values presented in Section 4.4.6 is given as a depth map in Figure 4.10b, compared to the application of a conventional depth averaging of compatible pixels as used by, for instance, Newcombe et al. [NIH\*11], depicted in Figure 4.10a. The strength of the proposed depth voting scheme becomes specifically apparent at depth discontinuities, i.e., object silhouettes, where the initial depths of the coarse object boundaries are refined by detecting and replacing erroneous measurements.



**Figure 4.10:** Ablation study for depth reconstruction. (a) The proposed approach, but with the conventional cumulative average weighting, e.g., [NIH\*11]. (b) The proposed approach with the presented depth voting scheme. Depths are shown using a *Parula* colormap ranging from 1.5 m to 2.75 m.

## 4.6.3 Qualitative Comparisons

As the proposed approach provides a high-quality image refinement method robust to disparity and occlusions and, thus, aims at filling the gap between interactive 2D image refinement methods, online 3D reconstruction techniques with high-resolution textures, and offline texture optimization methods for 3D scene reconstruction, it is compared to the following state-of-the-art techniques in these contexts.

*Kluge20*: 2D interactive *progressive refinement imaging* for (almost) planar scenes with only small amounts of disparity, as described in Chapter 3 and proposed by Kluge et al. [KWK20].

*Niessner13*: The online 3D scene reconstruction method using voxel hashing from Niessner et al. [NZIS13]. This approach is used by most of the color optimization methods, such as [ZK14, FYL\*21, FYLX20].

*Lee20*: The online 3D scene reconstruction method *TextureFusion* from Lee et al. [LHD\*20] stores sub-voxel textures in the TSDF voxel grid cells containing the scene surface.

*Ha21*: The online 3D scene reconstruction method *NormalFusion* from Ha et al. [HLMK21], a follow-up work of [LHD\*20], additionally obtains photometric normals, enabling geometric enhancement.

*Fu21*: The offline texture optimization proposed by Fu et al. [FYL\*21]. The initial scene reconstruction and camera poses are generated using Voxel-Hashing [NZIS13], and a subset of input frames is selected based on the angle and distance between corresponding poses, as proposed in [FYL\*21].

*Kluge23*: The method described in this chapter (see Section 4.2 and Sections 4.4.1 to 4.4.7) and proposed by Kluge et al. [KWK23].

### Comparison to 2D Image Reconstruction

*Progressive refinement imaging* for planar scenes (see Chapter 3) is compared to the proposed approach for 3D scenes, referred to as *Kluge20* and *Kluge23*, respectively, on the *BrickWall* and the *Memorial* data sets, which comprise a low to moderate amount of disparity; see Figure 4.11.

For the *BrickWall* data set, the approach for (almost) planar scenes works robustly and yields quite good results. However, for the *Memorial* data set, the limitations of the geometric alignment using a homography lead to strong geometric ghosting artifacts, while the proposed method for 3D scenes is able to reconstruct the silhouettes and captures more details. Note that *Kluge20* does not generate results on any of the other data sets due to alignment failures.

### Comparison to Online Scene Reconstruction

All data sets are reconstructed using the online 3D scene reconstruction approaches *Niessner13* (VoxelHashing), *Lee20* (NormalFusion), and *Ha21* (TextureFusion) as a comparison to the proposed method, referred to as *Kluge23*; see Figures 4.12 and 4.13. To achieve the most detailed results, the smallest possible voxel size was used to successfully process a specific data set with 24 GB of GPU memory, if the reconstruction failed with the default size of 4 mm; see Table 4.3. Note that *Ha21* generates photometric normals as addi-



**Figure 4.11:** Comparison of *progressive refinement imaging* for (almost) planar scenes with the proposed approach for 3D scenes, referred to as *Kluge20* and *Kluge23*, respectively. See also Figure 4.7 for a comparison with the unrefined reference image.

tional per-voxel attribute maps besides the texture patches, which requires a significant amount of memory, depending on the scene.

All methods successfully reconstruct all scenes, but due to the nature of the 3D scene representation, 3D scene reconstruction methods potentially produce holes or incomplete color reconstructions. Further scene-dependent deficiencies can be observed, which are exemplified in the following.

**Table 4.3:** Voxel sizes (mm) for the data sets, used by the competing methods.

	<i>Fountain</i>	<i>CoffeeTable</i>	<i>BooksGlobe</i>	<i>VillageModel</i>	<i>BrickWall</i>	<i>Memorial</i>	<i>Statue</i>	<i>Cannon</i>	<i>FlowerBed</i>	<i>BunnySynth</i>
<i>Niessner13</i>	4	4	4	4	4	4	4	4	4	4
<i>Lee20</i>	4	4	4	4	8	5	4	4	6	4
<i>Ha21</i>	4	6	4	8	25	9	10	11	14	4





Figure 4.12: Comparison with online scene reconstruction methods. See also Figure 4.7 for a comparison with the unrefined reference image.

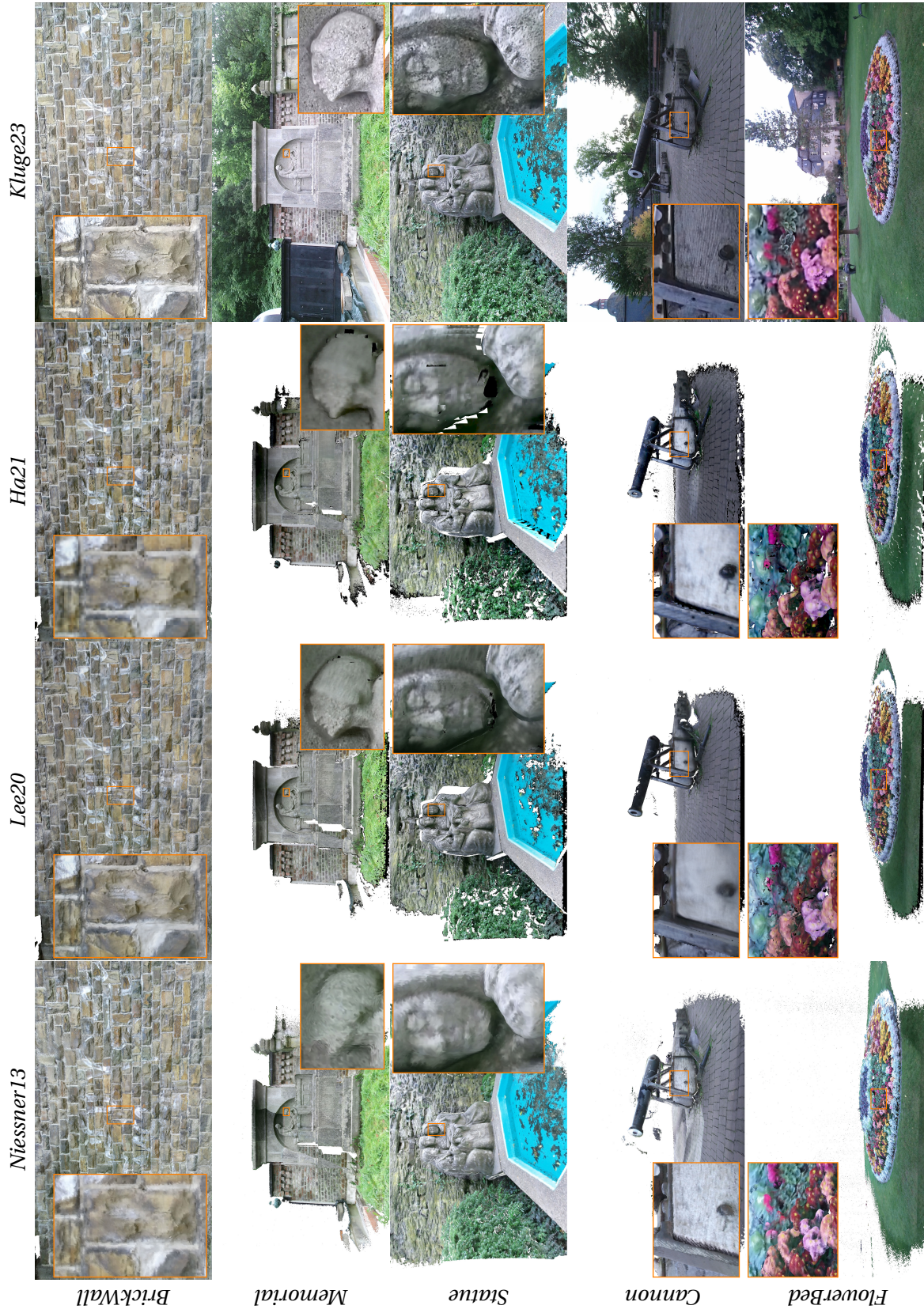


Figure 4.13: Comparison with online scene reconstruction methods. See also Figure 4.7 for a comparison with the unrefined reference image.

*Niessner13* exhibits, for example, local geometric inconsistencies (*Fountain*, *BooksGlobe*, *VillageModel*), as well as smoothed-out photometric reconstructions (*CoffeeTable*, *Statue*), but is partially able to reconstruct texture details (*Cannon*). *Lee20* partly reconstructs sharp details (*Fountain*) and silhouettes (*Memorial*), but also produces very blurred results (*CoffeeTable*, *BooksGlobe*, *Cannon*). Likewise, *Fu21* can partially reconstruct sharp details (*Fountain*, *BooksGlobe*) while delivering blurry results in other cases (*VillageModel*, *Cannon*).

Besides the *FlowerBed* data set, the method proposed in this chapter yields high-quality results regarding geometric and photometric consistency. It can successfully refine the reference image in geometrically homogeneous regions as well as at object silhouettes, and suppresses locally misaligned information (e.g., due to erroneous input range values).

The *FlowerBed* data set is very challenging, as it comprises many detailed silhouettes for which the range maps are not detailed and reliable enough. This leads to a large amount of outliers and to a comparably small amount of details that pass the outlier test and get incorporated into the refined RGB-D image.

### Comparison to Offline Optimization

Figure 4.14 shows the results of comparing the presented method to the offline, global post-optimization approach *Fu21* for the *Fountain*, *CoffeeTable*, and *BooksGlobe* data sets. Note that the *Fountain* data set footage comprises only limited amounts of close-ups of the specular tilework. For all three data sets, *Fu21* delivers geometrically good results, but there are photometric inconsistencies. The proposed method yields reconstructions with significantly improved photometric consistency, as the reference frame’s illumination condition is retained.

### Comparison of Reconstructed Depths

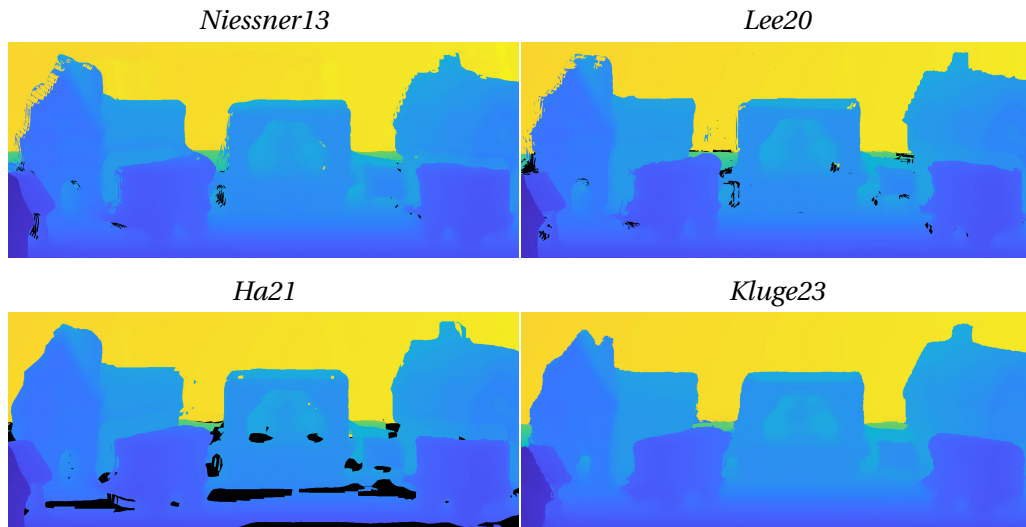
While the pipeline’s main output is a high-quality color reconstruction, the resulting depth map may have its uses (e.g., for stereo image generation). Therefore, the depth map reconstruction is compared to the scene reconstructions of *Niessner13*, *Lee20*, and *Ha21* by rendering the surface from the same viewpoint. The results of this experiment are shown in Figure 4.15 for the *VillageModel* data set. While all approaches show competitive results, more consistent object silhouettes and fewer holes are provided by the proposed method.



**Figure 4.14:** Comparison with the offline, post-processing approach *Fu21* [FYL\*21]. See also Figure 4.7 for a comparison with the unrefined reference image.

#### 4.6.4 Robustness against Self-Localization Drift

To demonstrate the robustness of the method against drift effects in camera tracking, the 360° data set *LongOfficeHousehold* is used, comprising 2488 RGB-D frames. The proposed system processes the first 326 frames, i.e., it selects 13 frames to be incorporated into the model. Later on, when the camera turns closer to the reference pose again, frames 1771–2488 are processed, from which 18 frames are selected. Figure 4.16 shows the refinement before exiting the reference viewpoint (left) and the final refinement after re-entering the reference viewpoint (right), yielding a sharper reconstruction.



**Figure 4.15:** Comparison of the reconstructed depths for the *VillageModel* data set, using a *Parula* colormap ranging from 1.5 m to 2.75 m.



**Figure 4.16:** Robustness against self-localization drift. Refinement of the *LongOfficeHousehold* data set before exiting the reference viewpoint (left) and the final refinement after re-entering the reference viewpoint (right) using the presented approach. See also Figure 4.7 for a comparison with the unrefined reference image.

### 4.6.5 Quantitative Comparison

To provide a quantitative comparison, all methods are evaluated on the synthetic data set *BunnySynth* by comparing the final, refined color and depth images (see Fig. 4.17 and Fig. 4.19) to the ground truth, i.e., the initial color and depth frame at four times the resolution ( $7680 \times 4320$  px).

In Tab. 4.4, the PSNR, the structural similarity SSIM [WBSS04], and the perceptual quality LPIPS [ZIE\*18] are reported for the refined color, while in Tab. 4.5, RMSE and MAE are shown for the resulting depths, revealing a significant advantage of the proposed method. Furthermore, the error maps (per-pixel absolute error) for the refined color images compared to the ground truth are shown in Fig. 4.18, and the absolute distance error [mm] for the corresponding depth maps in Fig. 4.20. Note that invalid (unknown) pixels were excluded in all error calculations for per-pixel metrics.

PSNR, SSIM, LPIPS

**Table 4.4:** Quantitative evaluation of the refined color for the synthetic data set *BunnySynth*. The average PSNR (dB (higher is better) and SSIM [WBSS04] (higher is better) are reported over the full image to evaluate the overall consistency to the ground truth, as well as the average error over a selected region *R1* (see Figs. 4.17 and 4.18) to evaluate the amount of detail achieved in the refined image. To evaluate perceptual quality, the LPIPS [ZIE\*18] score (lower is better) is employed, which uses deep features.

	Full image			Region <i>R1</i>		
	PSNR (dB) (↑)	SSIM (↑)	LPIPS (↓)	PSNR (dB) (↑)	SSIM (↑)	LPIPS (↓)
<i>Fu21</i>	17.30	0.76	0.50	9.82	0.33	0.44
<i>Niessner13</i>	19.59	0.87	0.41	11.23	0.44	0.64
<i>Lee20</i>	22.26	0.81	0.42	17.08	0.65	0.31
<i>Ha21</i>	15.22	0.66	0.58	16.07	0.46	0.51
<i>Kluge23</i>	29.50	0.96	0.16	18.32	0.73	0.18

RMSE, MAE

**Table 4.5:** Quantitative evaluation of the resulting depths for the synthetic data set *BunnySynth*. The average RMSE (mm) (lower is better) and MAE (mm) (lower is better) are reported over the full image to evaluate the overall consistency to the ground truth, as well as the average error over a selected region *R2* (see Figs. 4.19 and 4.20) to evaluate the achieved accuracy at object silhouettes.

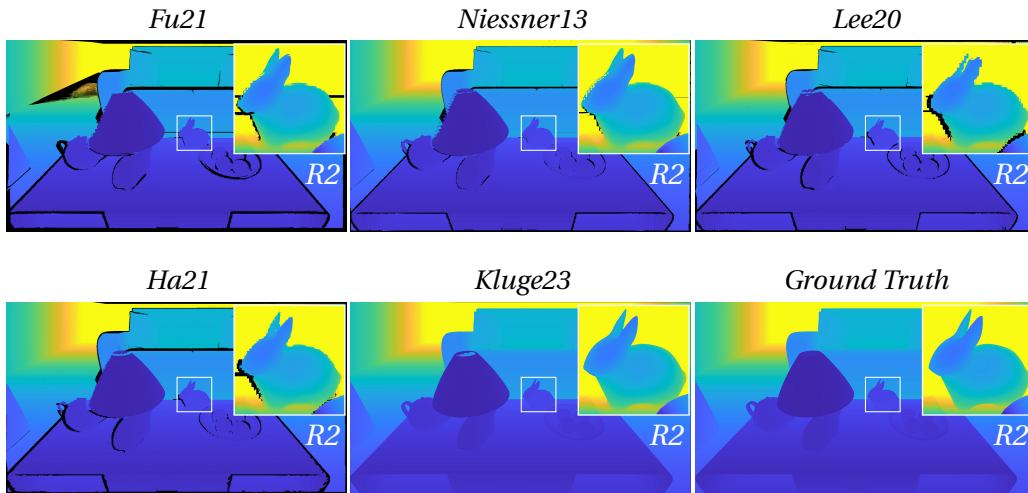
	Full image		Region <i>R2</i>	
	RMSE (mm) (↓)	MAE (mm) (↓)	RMSE (mm) (↓)	MAE (mm) (↓)
<i>Fu21</i>	101.59	10.40	194.93	41.07
<i>Niessner13</i>	87.92	7.79	181.99	36.12
<i>Lee20</i>	82.05	7.12	163.76	28.68
<i>Ha21</i>	80.60	7.03	146.53	23.57
<i>Kluge23</i>	65.22	3.54	69.56	5.66



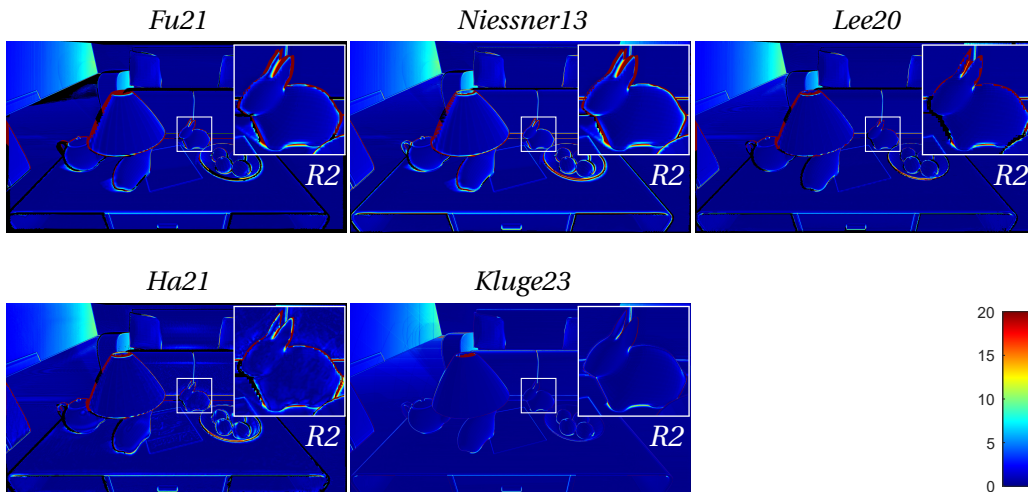
**Figure 4.17:** Results (refined color) of the synthetic data set *BunnySynth*, used for the quantitative comparison in Tab. 4.4. See also Fig. 4.7 for a comparison with the initial, unrefined frame.



**Figure 4.18:** Error maps (per-pixel absolute error) corresponding to the refined color images shown in Fig. 4.17.



**Figure 4.19:** Results (refined depth) of the synthetic data set *BunnySynth*, used for the quantitative comparison in Tab. 4.5. Depth maps are shown using a *Parula* colormap ranging from 0.5 m to 4.5 m for the full image and from 0.85 m to 1.0 m for the inset.



**Figure 4.20:** Absolute distance error [mm] corresponding to the resulting depth maps shown in Fig. 4.19.



### 4.6.6 Performance

All experiments are performed using an AMD Ryzen Threadripper 3970X with 128 GB main memory and an NVIDIA GeForce RTX 4090 with 24 GB GPU memory. Table 4.6 compares the timings for a complete reconstruction process of each method and the required peak memory. For *Niessner13*, *Lee20*, and *Ha21*, the memory consumption using the minimal amount of pre-allocated data structure elements is shown, determined using two passes. Note that in a true online scenario, this is not known beforehand, and thus, more memory would have been pre-allocated. Since the offline, post-processing method *Fu21* requires a large amount of processing time, up to several weeks (*CoffeeTable*), only three data sets are shown for this method; the *Statue* and *FlowerBed* data sets were stopped after ten and six days, respectively, when only the first of 30 iterations had been completed. For the presented method, the average frame rate over all data sets is 1.0 *fps*, with a minimum frame rate of 0.5 *fps* for the *Statue* data set and a maximum frame rate of 2.1 *fps* for the *Fountain* data set.

### 4.6.7 Limitations

In order to enable refinement imaging with parallax effects in the scene, the processing pipeline primarily depends on depth values to guide the alignment and disparity-corrective warping of the color information. However, in contrast to high-quality color data, depth images exhibit lower effective resolution and significantly increased noise, often correlated with visually important features like silhouettes. While the approach is explicitly designed for resilience against these low-quality characteristics, it is ultimately limited by the depth data provided.

The proposed method is not able to reliably refine RGB-D data sequences containing too fine-grained depth variations and silhouettes, resulting in too much unreliable depth information and outliers to be used for disparity correction. This is particularly evident in the *FlowerBed* data set, evaluated in Section 4.6.3, which comprises very detailed silhouettes in the color data for which the depth data’s reliability is insufficient. Even in homogeneous depth areas, range estimations may exhibit increased noise and erroneous values, e.g., on specular surfaces (such as the coffee machine in the *CoffeeTable* data set). Since this directly affects the accuracy of the color warping, the local realignment may not be sufficient.

Furthermore, the pipeline maintains photometric and geometric consistency with respect to a reference image that needs to cover the scene of interest entirely. To avoid introducing photometric inconsistencies, in contrast to Chapter 3, the lateral dimensions of the model are not extended to incorporate novel scene areas if the camera is exiting the region defined by the reference image.

Table 4.6: Required resources.

	Total processing time (h:min:s)					
	<i>Kluge20</i>	<i>Niessner13</i>	<i>Lee20</i>	<i>Ha21</i>	<i>Fu21</i>	<i>Kluge23</i>
<i>Fountain</i>	-	0:00:11	0:00:27	0:00:43	142:19:20	0:00:28
<i>LongOfficeH.</i>	-	-	-	-	-	0:00:18
<i>CoffeeTable</i>	-	0:01:03	0:02:35	0:05:13	483:25:02	0:03:38
<i>BooksGlobe</i>	-	0:00:16	0:00:20	0:00:40	23:13:39	0:00:28
<i>VillageModel</i>	-	0:00:56	0:02:23	0:04:49	-	0:04:01
<i>BrickWall</i>	0:08:58	0:03:15	0:07:55	0:13:17	-	0:10:21
<i>Memorial</i>	0:04:47	0:01:45	0:03:43	0:08:56	-	0:05:04
<i>Statue</i>	-	0:00:39	0:01:32	0:02:51	-	0:03:10
<i>Cannon</i>	-	0:00:16	0:00:44	0:01:03	-	0:01:01
<i>FlowerBed</i>	-	0:00:20	0:00:43	0:01:07	-	0:00:55

	Peak total main memory consumption (GB)					
	<i>Kluge20</i>	<i>Niessner13</i>	<i>Lee20</i>	<i>Ha21</i>	<i>Fu21</i>	<i>Kluge23</i>
<i>Fountain</i>	-	1.78	7.01	9.55	7.60	1.14
<i>LongOfficeH.</i>	-	-	-	-	-	1.13
<i>CoffeeTable</i>	-	4.88	60.23	60.38	13.90	1.32
<i>BooksGlobe</i>	-	1.77	10.96	12.57	4.40	1.27
<i>VillageModel</i>	-	4.54	53.35	54.21	-	1.37
<i>BrickWall</i>	1.64	21.56	122.04	123.07	-	1.44
<i>Memorial</i>	1.71	10.36	85.12	86.15	-	1.33
<i>Statue</i>	-	5.06	35.42	35.21	-	1.48
<i>Cannon</i>	-	3.25	18.74	18.51	-	1.41
<i>FlowerBed</i>	-	4.57	19.62	19.48	-	1.34

	Peak total GPU memory consumption (GB)					
	<i>Kluge20</i>	<i>Niessner13</i>	<i>Lee20</i>	<i>Ha21</i>	<i>Fu21</i>	<i>Kluge23</i>
<i>Fountain</i>	-	2.16	6.18	13.47	0.36	2.77
<i>LongOfficeH.</i>	-	-	-	-	-	1.86
<i>CoffeeTable</i>	-	3.67	11.78	20.80	0.25	9.46
<i>BooksGlobe</i>	-	3.24	7.20	12.58	0.51	4.99
<i>VillageModel</i>	-	3.87	12.48	20.46	-	6.48
<i>BrickWall</i>	8.24	9.55	22.19	19.12	-	11.61
<i>Memorial</i>	7.54	5.26	21.29	22.49	-	8.47
<i>Statue</i>	-	4.60	16.49	19.63	-	12.33
<i>Cannon</i>	-	4.86	21.41	20.21	-	10.79
<i>FlowerBed</i>	-	5.87	21.71	20.12	-	8.18

## 4.7 Summary

A novel progressive RGB-D image refinement pipeline was presented that instantaneously produces a high-quality, geometrically and photometrically consistent RGB-D image reconstruction from RGB-D image sequences. Assisted by depth values to guide the alignment and to correct for disparity, the proposed design allows for the refinement of general 3D scenes and, thus, fills the gap between 2D *progressive refinement imaging* and online 3D reconstruction techniques with high-resolution textures.

Colors and depths are hierarchically fused into an adaptive-resolution, progressively improving model of the scene, while strictly decoupling color data from the coarse and potentially incomplete geometry representation. The pipeline modules are designed for resilience against low-quality, low-resolution depth information while refining the high-resolution color data in homogeneous depth regions as well as at object silhouettes. To that end, the presented method performs local color consistency operations in image space before applying a novel blending strategy for color fusion, taking the gain in visual detail into account. To benefit from progressively refined range values, depths are fused based on a novel depth voting scheme that allows for correcting inaccurate depth estimates.



# 5

## Conclusions

*This chapter summarizes the contributions and concludes the thesis. It also opens up possible directions of future work.*

---

### 5.1 Summary

In the past years, various algorithmic approaches have been proposed that address the fusion of multiple camera observations, enabling the acquisition of scenes that cannot be captured with a single photograph. Despite various improvements in seamless image blending, a key challenge to creating a convincing composite remains in compensating for geometric and photometric discrepancies (due to, for example, changes in viewpoint and illumination conditions). While previous methods mitigate these inconsistencies mainly through global optimization, any kind of computationally intensive post-processing prevents an acquisition in an interactive, online fashion.

In this thesis, novel methods for fusing a stream of camera observations into a *progressively refined*, consistent image representation have been proposed. By enriching a low-resolution image with high-resolution details from close-ups, the user is allowed to interactively increase resolution locally where added image detail is desired.

First, a method has been proposed to fuse an RGB image sequence with substantial geometric and photometric discrepancies into a single consistent output image. It can handle large sets of images, acquired from a nearly planar or far-distant scene at variable object-space resolutions and under varying local or global illumination conditions. At its core, a dynamically extendable multi-scale representation allows for *variable-resolution* image fusion. Details from the incoming image data are selectively merged in a way that removes artifacts such as lens distortions, lighting changes, or varying exposure and color balance.

Progressive  
refinement imaging

Second, by bridging between 2D and 3D approaches, a *disparity-corrected*

Depth-assisted  
disparity correction

method has been proposed that allows adaptive image refinement for general 3D scenes, even in the presence of silhouettes and strong scene parallax. It features the fusion of handheld RGB-D camera streams into a high-quality, *variable-resolution* 2.5-D reconstruction (color and range data). This is enabled by a parallax-aware image warping, assisted by adaptively refined depth values to compensate for parallax effects due to depth disparities. All pipeline modules are designed for resilience against low-resolution, artifact-prone depth readings while refining the high-resolution color data.

## 5.2 Future Work

**Other modalities** The individual pipeline modules mainly apply generic techniques to align and correct image data with respect to a specific reference image (e.g., optical flow to correct distortions). That is, even though the presented pipeline has been validated only on RGB and RGB-D data, the proposed approach can potentially be applied to other image modalities. Therefore, it would be interesting to employ and evaluate the proposed pipeline for modalities such as satellite imagery or computed tomography.

**Other scenarios** Future work may address the adaption of the proposed pipeline to other scenarios involving the detection of local inconsistencies, for example, in the context of satellite imaging or cultural heritage for detecting (bio-)deterioration. To this end, a re-entry functionality would be of practical relevance to support the continuation of a previously acquired reconstruction.

**Keyframe texture generation** In future work, the proposed refinement may be applied to multiple keyframes during 3D scene reconstruction, generating high-resolution textures from multiple viewpoints, independent from a potentially low-grade geometry reconstruction.

**Real-time capability** To improve the computational efficiency of the proposed online approaches towards real-time applications, ideally, concurrent kernel scheduling should be applied to overlap data transfers and other operations by performing multiple CUDA operations simultaneously, which has yet to be realized in the current implementation.

# Bibliography

- [3DV] 3DVISTA: Stitcher 4. [www.3dvista.com/en/products/stitcher](http://www.3dvista.com/en/products/stitcher). [Accessed on 8 Nov. 2023].
- [AAB\*84] ADELSON E. H., ANDERSON C. H., BERGEN J. R., BURT P. J., OGDEN J. M.: Pyramid methods in image processing. *RCA engineer* 29, 6 (1984), 33–41.
- [ABD12] ALCANTARILLA P. F., BARTOLI A., DAVISON A. J.: Kaze features. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part VI 12* (2012), Springer, pp. 214–227.
- [ADA\*04] AGARWALA A., DONTCHEVA M., AGRAWALA M., DRUCKER S., COLBURN A., CURLESS B., SALESIN D., COHEN M.: Interactive digital photomontage. *ACM Trans. Graphics* 23, 3 (2004), 294–302.
- [Ado] ADOBE INC.: Photoshop cc 20.0.1. [www.adobe.com/products/photoshop.html](http://www.adobe.com/products/photoshop.html). [Accessed on 8 Nov. 2023].
- [Alb] ALBATROSS DESIGN GROUP: Adg panorama tools professional 5.3. [www.albatrossdesign.com/products/panorama](http://www.albatrossdesign.com/products/panorama). [Accessed on 8 Nov. 2023].
- [AlM06] ALMARE: File:herculaneum neptune and amphitrite.jpg. [commons.wikimedia.org/wiki/File:Herculaneum\\_Neptune\\_And\\_Amphitrite.jpg](https://commons.wikimedia.org/wiki/File:Herculaneum_Neptune_And_Amphitrite.jpg), 5 June 2006.
- [Amp16] AMPHIPOLIS: File:herculaneum - house of neptune and amphitrite (14732583079).jpg. [commons.wikimedia.org/wiki/File:Herculaneum\\_%E2%80%94\\_House\\_of\\_Neptune\\_and\\_Amphitrite\\_\(14732583079\).jpg](https://commons.wikimedia.org/wiki/File:Herculaneum_%E2%80%94_House_of_Neptune_and_Amphitrite_(14732583079).jpg), 1 Nov. 2016.
- [BA83a] BURT P., ADELSON E.: The laplacian pyramid as a compact image code. *IEEE Transactions on Communications* 31, 4 (1983), 532–540.
- [BA83b] BURT P. J., ADELSON E. H.: A multiresolution spline with application to image mosaics. *ACM Trans. Graphics* 2, 4 (1983), 217–236.

- [BK93] BURT P. J., KOLCZYNSKI R. J.: Enhanced image capture through fusion. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)* (1993), pp. 173–182.
- [BKR17] BI S., KALANTARI N. K., RAMAMOORTHI R.: Patch-based optimization for image-based texture mapping. *ACM Trans. Graphics* 36, 4 (2017), 106–1.
- [BL07] BROWN M., LOWE D. G.: Automatic panoramic image stitching using invariant features. *Int. Journal of Computer Vision (IJCV)* 74, 1 (2007), 59–73.
- [BM92] BESL P. J., MCKAY N. D.: Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures* (1992), vol. 1611, pp. 586–606.
- [Boo89] BOOKSTEIN F. L.: Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence* 11, 6 (1989), 567–585.
- [BPC17] BURNS C., PLYER A., CHAMPAGNAT F.: Texture super-resolution for 3d reconstruction. In *Proc. International Conference on Machine Vision Applications (MVA)* (2017), pp. 350–353.
- [Bro18] BROWN M.: Autostitch 3.0. Available from: [matthewalunbrown.com/autostitch/autostitch.html](http://matthewalunbrown.com/autostitch/autostitch.html), 2018. [Accessed on 8 Nov. 2023].
- [BTVG06] BAY H., TUYTELAARS T., VAN GOOL L.: Surf: Speeded up robust features. In *Proc. Europ. Conf. Computer Vision (ECCV)* (2006), pp. 404–417.
- [Buc80] BUCHSBAUM G.: A spatial processor model for object colour perception. *Journal of the Franklin institute* 310, 1 (1980), 1–26.
- [Bur81] BURT P. J.: Fast filter transform for image processing. *Computer graphics and image processing* 16, 1 (1981), 20–51.
- [Can86] CANNY J.: A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, 6 (1986), 679–698.
- [CDLN07] CRETE F., DOLMIERE T., LADRET P., NICOLAS M.: The blur effect: perception and estimation with a new no-reference perceptual blur metric. In *Human vision and electronic imaging XII* (2007), vol. 6492, p. 64920I.



- [CL96] CURLESS B., LEVOY M.: A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), pp. 303–312.
- [CM92] CHEN Y., MEDIONI G.: Object modelling by registration of multiple range images. *Image and vision computing* 10, 3 (1992), 145–155.
- [Cra14] CRASH TEST MIKE: Herculaneum. [www.flickr.com/photos/crashtestmike/14762599170/](http://www.flickr.com/photos/crashtestmike/14762599170/), 8 Aug. 2014.
- [d'A] D'ANGELO P.: Hugin 2018. [hugin.sourceforge.net/download](http://hugin.sourceforge.net/download). [Accessed on 8 Nov. 2023].
- [Dav07] DAVIDSON J.: Herculaneum. [www.flickr.com/photos/49519215@N00/622102957/](http://www.flickr.com/photos/49519215@N00/622102957/), 8 Mar. 2007.
- [Duc77] DUCHON J.: Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive Theory of Functions of Several Variables: Proceedings of a Conference Held at Oberwolfach April 25–May 1, 1976* (1977), Springer, pp. 85–100.
- [Eas] EASYPANO HOLDINGS INC.: Panoweaver 10 professional edition. [www.easypano.com/download-panorama-software.html](http://www.easypano.com/download-panorama-software.html). [Accessed on 8 Nov. 2023].
- [EESM10] EISEMANN M., EISEMANN E., SEIDEL H.-P., MAGNOR M.: Photo zoom: High resolution from unordered image collections. In *Proceedings of Graphics Interface 2010* (2010), Canadian Information Processing Society, pp. 71–78.
- [EF01] EFROS A. A., FREEMAN W. T.: Image quilting for texture synthesis and transfer. In *Proc. SIGGRAPH* (2001), pp. 341–346.
- [EUS06] EDEN A., UYTENDAELE M., SZELISKI R.: Seamless image stitching of scenes with large motions and exposure differences. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)* (2006), vol. 2, pp. 2498–2505.
- [Far02] FARNEBÄCK G.: *Polynomial expansion for orientation and motion estimation*. PhD thesis, Linköping University Electronic Press, 2002.
- [Far03] FARNEBÄCK G.: Two-frame motion estimation based on polynomial expansion. In *Proc. Scandinavian Conf. Image analysis* (2003), Springer, pp. 363–370.

- [FB81] FISCHLER M. A., BOLLES R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24, 6 (1981), 381–395.
- [Fir] FIRMTOOLS: Panorama composer 3. [panorama.firmtools.com/download.php](http://panorama.firmtools.com/download.php). [Accessed on 8 Nov. 2023].
- [FP02] FORSYTH D. A., PONCE J.: *Computer vision: a modern approach*. prentice hall professional technical reference, 2002.
- [FYL\*21] FU Y., YAN Q., LIAO J., ZHOU H., TANG J., XIAO C.: Seamless texture optimization for rgb-d reconstruction. *IEEE Transactions on Visualization and Computer Graphics* (2021).
- [FYLX20] FU Y., YAN Q., LIAO J., XIAO C.: Joint texture and geometry optimization for rgb-d reconstruction. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 5950–5959.
- [FYY\*18] FU Y., YAN Q., YANG L., LIAO J., XIAO C.: Texture mapping for 3d reconstruction with rgb-d sensor. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)* (2018), pp. 4645–4653.
- [Gig] GIGAPAN SYSTEMS: Gigapan stitch 2.1. [www.gigapan.com/cms/support/download-gigapan-stitch](http://www.gigapan.com/cms/support/download-gigapan-stitch). [Accessed on 8 Nov. 2023].
- [Gos88] GOSHTASBY A.: Registration of images with geometric distortions. *IEEE Transactions on Geoscience and Remote Sensing* 26, 1 (1988), 60–64.
- [Haa10] HAAR A.: Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen* 69 (1910), 331–371.
- [HD72] HARDER R. L., DESMARAIS R. N.: Interpolation using surface splines. *Journal of aircraft* 9, 2 (1972), 189–191.
- [HDGN17] HUANG J., DAI A., GUIBAS L. J., NIESSNER M.: 3dlite: towards commodity 3d scanning for content creation. *ACM Trans. Graphics* 36, 6 (2017), 203–1.
- [Hec12] HECHT E.: *Optics*. Pearson Education India, 2012.
- [HK13] HILL D., KLEERUP M.: [File:house of the neptune mosaic \(7254083622\).jpg](https://commons.wikimedia.org/wiki/File:House_of_the_Neptune_Mosaic_(7254083622).jpg). [commons.wikimedia.org/wiki/File:House\\_of\\_the\\_Neptune\\_Mosaic\\_\(7254083622\).jpg](https://commons.wikimedia.org/wiki/File:House_of_the_Neptune_Mosaic_(7254083622).jpg), 16 Dec. 2013.

- [HLMK21] HA H., LEE J. H., MEULEMAN A., KIM M. H.: Normalfusion: Real-time acquisition of surface normals for high-resolution rgb-d scanning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2021).
- [HLSH17] HE M., LIAO J., SANDER P. V., HOPPE H.: Gigapixel panorama video loops. *ACM Trans. Graphics* 37, 1 (2017), 3:1–3:15.
- [HS81] HORN B. K., SCHUNCK B. G.: Determining optical flow. *Artificial intelligence* 17, 1-3 (1981), 185–203.
- [HS\*88] HARRIS C., STEPHENS M., ET AL.: A combined corner and edge detector. In *Alvey vision conference* (1988), vol. 15, Citeseer, pp. 10–5244.
- [HZ03] HARTLEY R., ZISSERMAN A.: *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [IKH\*11] IZADI S., KIM D., HILLIGES O., MOLYNEAUX D., NEWCOMBE R., KOHLI P., SHOTTON J., HODGES S., FREEMAN D., DAVISON A., FITZGIBBON A.: KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In *Proc. ACM Symp. User Interface Softw. & Tech.* (2011), pp. 559–568.
- [IKL\*10] IHRKE I., KUTULAKOS K. N., LENSCH H. P., MAGNOR M., HEIDRICH W.: Transparent and specular object reconstruction. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 2400–2426.
- [JJKL16] JEON J., JUNG Y., KIM H., LEE S.: Texture map generation for 3d reconstructed scenes. *The Visual Computer* 32, 6 (2016), 955–965.
- [JRAA00] JACOBSON R., RAY S., ATTRIDGE G. G., AXFORD N.: *Manual of Photography*. Taylor & Francis, 2000.
- [KE12] KHOSHELHAM K., ELBERINK S. O.: Accuracy and resolution of kinect depth data for indoor mapping applications. *sensors* 12, 2 (2012), 1437–1454.
- [KH08] KAZHDAN M., HOPPE H.: Streaming multigrid for gradient-domain operations on large images. *ACM Trans. Graphics* 27, 3 (2008), 21:1–21:10.
- [KLL\*13] KELLER M., LEFLOCH D., LAMBERS M., IZADI S., WEYRICH T., KOLB A.: Real-time 3D reconstruction in dynamic scenes using point-based fusion. In *Proc. Conf. Joint 3DIM/3DPVT (3DV)* (2013), p. 8.

- [Kol18] KOLOR: Kolor autopano giga 4.4.2. Available from: [download.kolor.com/apg/stable/history](http://download.kolor.com/apg/stable/history), 2018. [Accessed on 8 Nov. 2023].
- [KP15] KOLB A., PECE F.: Range imaging. In *Digital Representations of the Real World: How to Capture, Model, and Render Visual Reality*, Magnor M., Grau O., Sorkine-Hornung O., Theobalt C., (Eds.). AK Peters / CRC Press, 2015, ch. 4, p. 51–64.
- [KSE\*03] KWATRA V., SCHÖDL A., ESSA I., TURK G., BOBICK A.: Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graphics* 22, 3 (2003), 277–286.
- [KUDC07] KOPF J., UYTTENDAELE M., DEUSSEN O., COHEN M. F.: Capturing and viewing gigapixel images. In *ACM Trans. Graphics* (2007), vol. 26, p. 93.
- [KWK20] KLUGE M., WEYRICH T., KOLB A.: Progressive refinement imaging. *Computer Graphics Forum* 39, 1 (2020), 360–374.
- [KWK23] KLUGE M., WEYRICH T., KOLB A.: Progressive refinement imaging with depth-assisted disparity correction. *Computers & Graphics* 115 (2023), 446–460.
- [Lam14a] LAMBERT M.: Meetingofstylesuk. [www.flickr.com/photos/mike\\_lambert/14411692449/](http://www.flickr.com/photos/mike_lambert/14411692449/), 7 July 2014.
- [Lam14b] LAMBERT M.: Meetingofstylesuk. [www.flickr.com/photos/mike\\_lambert/14594982691/](http://www.flickr.com/photos/mike_lambert/14594982691/), 7 July 2014.
- [LCA18] LAKSHMINARAYANAN V., CALVO M. L., ALIEVA T.: *Mathematical optics: Classical, quantum, and computational methods*. CRC Press, 2018.
- [LCS11] LEUTENEGGER S., CHLI M., SIEGWART R. Y.: Brisk: Binary robust invariant scalable keypoints. In *2011 International conference on computer vision* (2011), Ieee, pp. 2548–2555.
- [LFS21] LICORISH C., FARAJ N., SUMMA B.: Adaptive compositing and navigation of variable resolution images. In *Computer Graphics Forum* (2021), vol. 40, pp. 138–150.
- [LHD\*20] LEE J. H., HA H., DONG Y., TONG X., KIM M. H.: Texturefusion: High-quality texture acquisition for real-time rgb-d scanning. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 1272–1280.

- [LHK15] LAMBERS M., HOBERG S., KOLB A.: Simulation of time-of-flight sensors for evaluation of chip layout variants. *IEEE Sensors Journal* 15, 7 (2015), 4019–4026.
- [Lin10] LINDNER M.: *Calibration and real-time processing of time-of-flight range data*. PhD thesis, Universität Siegen, Germany, 2010.
- [LK81] LUCAS B. D., KANADE T.: An iterative image registration technique with an application to stereo vision. In *Proceedings DARPA Image Understanding Workshop* (1981), pp. 121–130.
- [LK07] LINDNER M., KOLB A.: Calibration of the intensity-related distance error of the pmd tof-camera. In *Intelligent Robots and Computer Vision XXV: Algorithms, Techniques, and Active Vision* (2007), vol. 6764, SPIE, pp. 338–345.
- [LKS\*17] LEFLOCH D., KLUGE M., SARBOLANDI H., WEYRICH T., KOLB A.: Comprehensive use of curvature for robust and accurate online surface reconstruction. *IEEE transactions on pattern analysis and machine intelligence* 39, 12 (2017), 2349–2365.
- [LLOC15] LIU S., LI W., OGUNBONA P., CHOW Y.-W.: Creating simplified 3d models with high quality textures. In *Proc. International Conference on Digital Image Computing: Techniques and Applications (DICTA)* (2015), pp. 1–8.
- [LM71] LAND E. H., MCCANN J. J.: Lightness and retinex theory. *Josa* 61, 1 (1971), 1–11.
- [Low04] LOWE D. G.: Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60, 2 (2004), 91–110.
- [LSTS04] LI Y., SUN J., TANG C.-K., SHUM H.-Y.: Lazy snapping. *ACM Trans. Graphics (Proc. SIGGRAPH)* 23, 3 (2004), 303–308.
- [Mal89] MALLAT S. G.: A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence* 11, 7 (1989), 674–693.
- [MBRS\*21] MARTIN-BRUALLA R., RADWAN N., SAJJADI M. S., BARRON J. T., DOSOVITSKIY A., DUCKWORTH D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 7210–7219.

- [MC13] MEILLAND M., COMPORT A. I.: Super-resolution 3d tracking and mapping. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)* (2013), pp. 5717–5723.
- [MCL\*21] MENG Q., CHEN A., LUO H., WU M., SU H., XU L., HE X., YU J.: Gnerf: Gan-based neural radiance field without posed camera. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)* (2021), pp. 6351–6361.
- [Mic] MICROSOFT: Image composite editor 2.0.3. [www.microsoft.com/en-us/research/product/computational-photography-applications/image-composite-editor](http://www.microsoft.com/en-us/research/product/computational-photography-applications/image-composite-editor). [Accessed on 8 Nov. 2023].
- [Mil75] MILGRAM D. L.: Computer methods for creating photomosaics. *IEEE Transactions on Computers* 100, 11 (1975), 1113–1119.
- [MJF\*21] MA J., JIANG X., FAN A., JIANG J., YAN J.: Image matching from handcrafted to deep features: A survey. *International Journal of Computer Vision* 129 (2021), 23–79.
- [MKC\*17] MAIER R., KIM K., CREMERS D., KAUTZ J., NIESSNER M.: Intrinsic3d: High-quality 3d reconstruction by joint appearance and geometry optimization with spatially-varying lighting. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)* (2017), pp. 3114–3122.
- [MSC15] MAIER R., STÜCKLER J., CREMERS D.: Super-resolution keyframe fusion for 3d modeling with high-quality textures. In *Proc. International Conference on 3D Vision (3DV)* (2015), pp. 536–544.
- [MST\*20] MILDENHALL B., SRINIVASAN P. P., TANCİK M., BARRON J. T., RAMAMOORTHY R., NG R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proc. Europ. Conf. Computer Vision (ECCV)* (2020), pp. 405–421.
- [New] NEW HOUSE INTERNET SERVICES BV: Ptgui pro 11.6. [www.ptgui.com/download.html](http://www.ptgui.com/download.html). [Accessed on 8 Nov. 2023].
- [NIH\*11] NEWCOMBE R. A., IZADI S., HILLIGES O., MOLYNEAUX D., KIM D., DAVISON A. J., KOHI P., SHOTTON J., HODGES S., FITZGIBBON A.: Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality* (2011), Ieee, pp. 127–136.
- [NM14] NASROLLAHI K., MOESLUND T. B.: Super-resolution: a comprehensive survey. *Machine Vision and Applications* 25, 6 (2014), 1423–1468.

- [NZIS13] NIESSNER M., ZOLLHÖFER M., IZADI S., STAMMINGER M.: Real-time 3D reconstruction at scale using voxel hashing. *ACM Trans. Graphics* 32, 6 (2013), 169.
- [PBP02] PRAUTZSCH H., BOEHM W., PALUSZNY M.: *Bézier and B-spline techniques*, vol. 6. Springer, 2002.
- [PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. *ACM Transactions on Graphics (TOG)* 22, 3 (2003), 313–318.
- [PPK03] PARK S. C., PARK M. K., KANG M. G.: Super-resolution image reconstruction: a technical overview. *IEEE signal processing magazine* 20, 3 (2003), 21–36.
- [PTX10] PULLI K., TICO M., XIONG Y.: Mobile panoramic imaging system. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR) - Workshops* (2010), pp. 108–115.
- [PZK17] PARK J., ZHOU Q.-Y., KOLTUN V.: Colored point cloud registration revisited. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 143–152.
- [Rap07] RAPP H.: *Experimental and theoretical investigation of correlating TOF-camera systems*. Master's thesis, University of Heidelberg, Germany, 2007.
- [Ras17] RASO C.: Glass mosaic of "triclinium" (dining-room) in the house of "neptune and amphitrite" at herculaneum, buried by vesuvius' eruption on 79 ad. [www.flickr.com/photos/70125105@N06/38425629544](http://www.flickr.com/photos/70125105@N06/38425629544), 22 Nov. 2017.
- [Rej57] REJLANDER O. G.: Two ways of life. First Manchester Art Treasures Exhibition, 1857.
- [RFB18] ROUHANI M., FRADET M., BAILLARD C.: A multi-resolution approach for color correction of textured meshes. In *Proc. International Conference on 3D Vision (3DV)* (2018), pp. 71–78.
- [RHHL02] RUSINKIEWICZ S., HALL-HOLT O., LEVOY M.: Real-time 3d model acquisition. *ACM Transactions on Graphics (TOG)* 21, 3 (2002), 438–446.
- [Rie09] RIEGER W.: File:herculaneum - casa di nettuno ed anfitrite - mosaic.jpg. [commons.wikimedia.org/wiki/File:Herculaneum\\_-\\_Casa\\_di\\_Nettuno\\_ed\\_Anfitrite\\_-\\_Mosaic.jpg](https://commons.wikimedia.org/wiki/File:Herculaneum_-_Casa_di_Nettuno_ed_Anfitrite_-_Mosaic.jpg), 15 Mar. 2009.

- [RL01] RUSINKIEWICZ S., LEVOY M.: Efficient variants of the icp algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling* (2001), IEEE, pp. 145–152.
- [Rob69] ROBINSON H. P.: *Pictorial Effect in Photography: Being Hints on Composition and Chiaro-oscuro for Photographers. To which is Added a Chapter on Combination Printing*. Piper & Carter, 1869.
- [RRKB11] RUBLEE E., RABAUD V., KONOLIGE K., BRADSKI G.: Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision* (2011), Ieee, pp. 2564–2571.
- [RSH\*99] RUECKERT D., SONODA L. I., HAYES C., HILL D. L., LEACH M. O., HAWKES D. J.: Nonrigid registration using free-form deformations: application to breast mr images. *IEEE transactions on medical imaging* 18, 8 (1999), 712–721.
- [Rue01] RUECKERT D.: Nonrigid registration: Concepts, algorithms, and applications. In *Medical image registration*, Hajnal J. V., Hill D. L., Hawkes D. J., (Eds.). CRC press, 2001, ch. 13, pp. 281–301.
- [SEE\*12] STURM J., ENGELHARD N., ENDRES F., BURGARD W., CREMERS D.: A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)* (Oct. 2012).
- [SG15] SERAFIN J., GRISETTI G.: Nicp: Dense normal based point cloud registration. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2015), IEEE, pp. 742–749.
- [SLK15] SARBOLANDI H., LEFLOCH D., KOLB A.: Kinect range sensing: Structured-light versus time-of-flight kinect. *Computer vision and image understanding* 139 (2015), 1–20.
- [SLOD21] SUCAR E., LIU S., ORTIZ J., DAVISON A. J.: imap: Implicit mapping and positioning in real-time. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)* (2021), pp. 6229–6238.
- [SP86] SEDERBERG T. W., PARRY S. R.: Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques* (1986), pp. 151–160.
- [SS97] SZELISKI R., SHUM H.-Y.: Creating full view panoramic image mosaics and environment maps. In *Proc. SIGGRAPH* (1997), pp. 251–258.



- [SUS11] SZELISKI R., UYTENDAELE M., STEEDLY D.: Fast poisson blending using multi-splines. In *Proc. IEEE Int. Conf. Computational Photography* (2011), pp. 1–8.
- [Taw] TAWBAWARE: Ptassembler 6.3. [www.tawbaware.com/ptasmlr.htm](http://www.tawbaware.com/ptasmlr.htm). [Accessed on 8 Nov. 2023].
- [Teo] TEOREX: Photostitcher 2.0. [www.photostitcher.com/download.html](http://www.photostitcher.com/download.html). [Accessed on 8 Nov. 2023].
- [TH84] TSAI R. Y., HUANG T. S.: Multiframe image restoration and registration. *Multiframe image restoration and registration 1* (1984), 317–339.
- [TM98] TOMASI C., MANDUCHI R.: Bilateral filtering for gray and color images. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)* (1998), IEEE, pp. 839–846.
- [TM\*08] TUYTELAARS T., MIKOLAJCZYK K., ET AL.: Local invariant feature detectors: a survey. *Foundations and trends® in computer graphics and vision* 3, 3 (2008), 177–280.
- [TMHF00] TRIGGS B., MCLAUCHLAN P. F., HARTLEY R. I., FITZGIBBON A. W.: Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings* (2000), Springer, pp. 298–372.
- [TS18] TAREEN S. A. K., SALEEM Z.: A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. In *2018 International conference on computing, mathematics and engineering technologies (iCoMET)* (2018), IEEE, pp. 1–10.
- [tsh] TSHSOFT: Panoramastudio 3 pro. [www.tshsoft.de/en/download\\_en](http://www.tshsoft.de/en/download_en). [Accessed on 8 Nov. 2023].
- [Ume91] UMEYAMA S.: Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)* 13, 04 (1991), 376–380.
- [Uni57] UNIVERSITY OF MICHIGAN LIBRARY DIGITAL COLLECTIONS: Fig. 7. oscar gustav rejlander, two ways of life, 1857, composite albumen print, 41.0 x 79.0 cm.; the mountain and the mole-hill: Julia margaret cameron's allegories. [https://quod.lib.umich.edu/b/bulletinic/x-07101-und-07/07101\\_07](https://quod.lib.umich.edu/b/bulletinic/x-07101-und-07/07101_07), 1857. [Accessed on 12 Dec., 2023].

- [WB89] WILLIAMS C. S., BECKLUND O. A.: *Introduction to the optical transfer function*. Wiley, 1989.
- [WBSS04] WANG Z., BOVIK A. C., SHEIKH H. R., SIMONCELLI E. P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Processing (TIP)* 13, 4 (2004), 600–612.
- [WG18] WANG C., GUO X.: Plane-based optimization of geometry and texture for rgb-d reconstruction of indoor scenes. In *Proc. International Conference on 3D Vision (3DV)* (2018), pp. 533–541.
- [WMG14] WAECHTER M., MOEHRLE N., GOESELE M.: Let there be color! large-scale texturing of 3d reconstructions. In *Proc. Europ. Conf. Computer Vision (ECCV)* (2014), pp. 836–850.
- [ZF03] ZITOVA B., FLUSSER J.: Image registration methods: a survey. *Image and vision computing* 21, 11 (2003), 977–1000.
- [Zha00] ZHANG Z.: A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence* 22, 11 (2000), 1330–1334.
- [ZIE\*18] ZHANG R., ISOLA P., EFROS A. A., SHECHTMAN E., WANG O.: The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 586–595.
- [ZK14] ZHOU Q.-Y., KOLTUN V.: Color map optimization for 3d reconstruction with consumer depth cameras. *ACM Trans. Graphics* 33, 4 (2014), 1–10.
- [ZK15] ZHOU Q.-Y., KOLTUN V.: Depth camera tracking with contour cues. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 632–638.
- [ZPL\*22] ZHU Z., PENG S., LARSSON V., XU W., BAO H., CUI Z., OSWALD M. R., POLLEFEYS M.: Nice-slam: Neural implicit scalable encoding for slam. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 12786–12796.
- [ZSG\*18] ZOLLHÖFER M., STOTKO P., GÖRLITZ A., THEOBALT C., NIESSNER M., KLEIN R., KOLB A.: State of the art on 3d reconstruction with rgb-d cameras. In *Computer graphics forum* (2018), vol. 37, Wiley Online Library, pp. 625–652.

- [Zuc12a] ZUCKER S.: Christ and john, deësis mosaic, hagia sophia. [www.flickr.com/photos/profzucker/14068351579](http://www.flickr.com/photos/profzucker/14068351579), 1 Jan. 2012.
- [Zuc12b] ZUCKER S.: Christ (bust), deësis mosaic, hagia sophia. [www.flickr.com/photos/profzucker/14068426300](http://www.flickr.com/photos/profzucker/14068426300), 1 Jan. 2012.
- [Zuc12c] ZUCKER S.: Christ, deësis mosaic (bust), hagia sophia. [www.flickr.com/photos/profzucker/14068340119](http://www.flickr.com/photos/profzucker/14068340119), 1 Jan. 2012.
- [Zuc12d] ZUCKER S.: Christ, deësis mosaic in sunlight, hagia sophia. [www.flickr.com/photos/profzucker/14068394737](http://www.flickr.com/photos/profzucker/14068394737), 1 Jan. 2012.
- [Zuc12e] ZUCKER S.: Christ, deësis mosaic in sunlight, hagia sophia. [www.flickr.com/photos/profzucker/14231876576](http://www.flickr.com/photos/profzucker/14231876576), 1 Jan. 2012.
- [Zuc12f] ZUCKER S.: Christ's face (close), deësis mosaic, hagia sophia. [www.flickr.com/photos/profzucker/14068317039](http://www.flickr.com/photos/profzucker/14068317039), 1 Jan. 2012.
- [Zuc12g] ZUCKER S.: Christ's face, deësis mosaic in sunlight, hagia sophia. [www.flickr.com/photos/profzucker/14254970265](http://www.flickr.com/photos/profzucker/14254970265), 1 Jan. 2012.
- [Zuc12h] ZUCKER S.: Deësis mosaic, hagia sophia. [www.flickr.com/photos/profzucker/14275161473](http://www.flickr.com/photos/profzucker/14275161473), 1 Jan. 2012.
- [Zuc12i] ZUCKER S.: Mary, deësis mosaic in sunlight, hagia sophia. [www.flickr.com/photos/profzucker/14275199463](http://www.flickr.com/photos/profzucker/14275199463), 1 Jan. 2012.
- [Zuc12j] ZUCKER S.: Virgin mary (close), deësis mosaic, hagia sophia. [www.flickr.com/photos/profzucker/14231848946](http://www.flickr.com/photos/profzucker/14231848946), 1 Jan. 2012.