

Department of
Informatics, Systems and Communication

PhD program in Computer Science

Cycle XXXV

A computational framework for Mixed Reality catalogs: from 3D reconstruction to rendering and interaction

Marelli Davide

748334

Supervisor: Prof. Gianluigi Ciocca

Co-supervisor: Prof. Simone Bianco

Tutor: Prof. Federico Cabitza

Coordinator: Prof. Leonardo Mariani

ACADEMIC YEAR 2021/2022

To the people who made this journey memorable

At least I learned something from all of this...

What's that?

- > How to deal with frustration, disappointment, and irritating cynicism.
It's not the size of the ship...
- Never pay more than 20 bucks for a computer game.

– Guybrush Threepwood

Abstract

Presentation and communication of characteristics about products and, in general, collections of objects or services are of crucial relevance in several fields. The most prominent application is given by the necessity of manufacturers and stores to advertise their offers to potential customers. To this purpose, paper or digital catalogs are often employed. Furthermore, catalogs may have goals that differ from the advertisement of physical products. They may also build on the presentation of knowledge (e.g. in a library or museum) or refer to services provided by some company or individual. With the growing diffusion of smartphones, tablets, and head-mounted displays, new possibilities arise for the catalogs to be presented by taking advantage of Mixed Reality (MR) concepts that allow the blending of virtual elements within the real world. To this end, this thesis proposes a computational framework that supports the creation of MR catalogs throughout the stages of design, development, and fruition of the catalog itself. The creation of MR catalogs involves several aspects. The key ones among them are investigated in this research. These include 3D geometry reconstruction, material appearance acquisition, and presentation of the MR catalog. Researchers have largely investigated these fields separately. This thesis aims to tackle specific problems of their usage in the creation of an MR catalog. 3D geometry reconstruction finds use in the creation of virtual 3D models of real elements for display in MR catalogs. The problem of selecting the most suitable 3D reconstruction method is addressed. In addition to 3D geometry, the look and feel of virtual elements are crucial to make them appear realistic. This thesis proposes the design and development of a low cost portable device for material appearance acquisition. Finally, the presentation of the MR catalog involves the rendering of the virtual elements and interaction with the user. In order to accomplish that, a generic framework for the creation of MR catalogs is defined. Possible use case scenarios of the proposed framework in the development of MR catalog applications are investigated.

Contents

Abstract	i
List of Figures	vi
List of Tables	ix
1 Introduction	1
1.1 Contributions	3
1.2 Organization of this thesis	4
2 Background and related work	8
2.1 Background	8
2.1.1 Introduction to e-catalogs	9
2.1.2 Introduction to Mixed Reality	11
2.1.3 Fields involved in a Mixed Reality catalog	13
2.2 Related work	13
3 3D geometry reconstruction	20
3.1 Challenges of 3D reconstruction in Mixed Reality catalogs	22
3.2 Related work	23
3.2.1 Structure from Motion	23
3.2.1.1 SfM building blocks	24
3.2.1.2 Incremental SfM pipelines	28
3.2.2 3D reconstruction datasets	30
3.3 Evaluation method for SfM 3D reconstruction	35
3.3.1 Alignment and registration	36
3.3.2 Evaluation of sparse point cloud	37
3.3.3 Evaluation of camera pose	39
3.3.4 Evaluation of dense point cloud	41
3.4 SfM Flow: synthetic data generation	42
3.4.1 Software functionalities	43

3.4.2	Impact	50
3.5	Proposed synthetic datasets for 3D reconstruction evaluation	50
3.5.1	The IVL-SYNTHSFM dataset	53
3.5.2	The IVL-SYNTHSFM-v2 dataset	56
3.5.3	The ENRICH dataset	58
3.5.3.1	Data generation method	60
3.6	Experimental results	67
3.6.1	Evaluation of SfM pipelines	69
3.6.1.1	The IVL-SYNTHSFM dataset	71
3.6.1.2	The ENRICH-Statue dataset	75
3.6.1.3	The ENRICH-Square dataset	77
3.6.2	The effects of depth of field, motion blur, and light changes on camera registration	81
3.6.3	The effects of Ground Control Points spatial distribution on the 3D accuracy	84
3.6.4	Monocular depth estimation	88
3.6.5	Impact	91
4	Material Appearance Acquisition	93
4.1	The Bidirectional Reflectance Distribution Function	94
4.1.1	BRDF models	97
4.1.1.1	Phenomenological models	97
4.1.1.2	Physically-based models	100
4.1.2	The Spatially Varying BRDF	102
4.1.3	Texture maps	103
4.1.4	Rendering	105
4.2	Related work	107
4.2.1	BRDF acquisition	107
4.2.2	SVBRDF/BTF acquisition	109
4.3	Photometric Stereo	111
4.4	The proposed SVBRDF acquisition device	115
4.4.1	Scope and design choices	115
4.4.2	Hardware	117

4.4.3	Software	121
4.4.3.1	Onboard firmware	121
4.4.3.2	Main software and processing pipeline	122
4.4.4	Device calibration	127
4.5	Experimental results	131
4.5.1	Albedo map evaluation	131
4.5.2	Normal map evaluation	135
4.5.3	Roughness map evaluation	137
4.5.4	Visual results	138
4.6	Known limitations	138
4.7	Impact	140
5	Rendering and Interaction	142
5.1	The Mixed Reality catalog framework	143
5.1.1	Related works	145
5.1.2	Workflow of a generic Mixed Reality catalog system	149
5.2	The textiles virtual catalog use case	153
5.2.1	Workflow	153
5.2.1.1	Simulating the light interaction	154
5.2.1.2	User interface	156
5.2.2	User evaluation	158
5.2.3	Known limitations and future work	163
5.3	The eyeglasses Virtual Try-On use case	164
5.3.1	3D face reconstruction	164
5.3.2	Comparing the 3D face reconstruction approaches	167
5.3.3	Workflow	176
5.3.3.1	Face detection and 3D reconstruction from a single image	178
5.3.3.2	Face size estimation	180
5.3.3.3	Fitting parameter estimation	181
5.3.3.4	User interface	184
5.3.4	User evaluation	187
5.3.5	Known limitations	190

5.4	Experimenting Mixed Reality catalogs on Smart Mirrors . . .	191
5.4.1	Related work	192
5.4.2	The proposed Smart Mirror	194
5.4.2.1	Functional requirements	195
5.4.2.2	Technology	196
5.4.3	Known limitations	202
6	Conclusion	203
	Appendices	209
	Appendix A SfM Flow: software architecture	210
	Appendix B SfM Flow: illustrative example	212
	Appendix C Face size estimation algorithm	215
	Acknowledgements	216
	Bibliography	217

List of Figures

2.1	Reality-Virtuality Continuum.	12
3.1	3D reconstruction using Structure from Motion.	21
3.2	Incremental Structure from Motion pipeline.	23
3.3	Samples from the most popular benchmark datasets.	32
3.4	Example of Ground Control Point patterns.	44
3.5	3D models used for synthetic data generation.	53
3.6	Example of synthetic dataset generation steps.	55
3.7	Example of data generation steps for the Jeep model.	57
3.8	Sample of rendered images of the Jeep model.	58
3.9	Orthographic view of the GCPs placement on the ENRICH-Aerial dataset.	61
3.10	Sample images from the ENRICH-Aerial dataset.	62
3.11	Camera paths on the ENRICH-Aerial dataset.	63
3.12	Equirectangular projection of the ENRICH-Square scene showing GCPs placement.	64
3.13	Sample images from the ENRICH-Square dataset.	64
3.14	Camera paths on the ENRICH-Square dataset.	65
3.15	GCPs locations on the ENRICH-Statue dataset.	66
3.16	Sample images from the ENRICH-Statue dataset.	67
3.17	Camera paths on the ENRICH-Statue dataset.	68
3.18	3D models used for synthetic dataset generation and Ignatius ground truth.	72
3.19	Example of dense point clouds using CMVS/PMVS on different SfM reconstructions.	74
3.20	Results obtained with COLMAP + RootSift.	76
3.21	Example of the ENRICH-Square sparse reconstruction.	78
3.22	Tie point extraction under rotation and illumination changes.	78
3.23	Tie point extraction under scale and illumination changes.	79

3.24	Tie point extraction under large scale changes.	79
3.25	Tie point extraction under large view changes.	80
3.26	Results of image registration on the IVL-SYNTHSFM-v2 – Hydrant dataset.	84
3.27	Schemes of the eleven GCPs configurations.	86
3.28	RMSE residuals on Check Points (CPs) with eleven GCPs configurations.	86
3.29	Average Cloud-to-Mesh distance in different GCPs configura- tion scenarios.	88
3.30	Example of predicted depth maps for the ENRICH-Square, ENRICH-Statue, and ENRICH-Aerial datasets.	91
4.1	Geometry of the BRDF notation.	96
4.2	Cook-Torrance BRDF microfacets.	100
4.3	Example of a conventional gonioreflectometer.	107
4.4	CAD drawings of the device and the assembled device.	121
4.5	Main software pipeline.	123
4.6	Roughness estimation pipeline.	126
4.7	User interface of the main processing pipeline.	127
4.8	RAW image before and after lens shading correction.	128
4.9	Incident light direction calibration target.	129
4.10	Incident light direction calibration pipeline.	130
4.11	DC ColorChecker comparison between ground truth and esti- mated albedo.	134
4.12	Normal map evaluation target.	135
4.13	Normal map evaluation pipeline and heatmap.	136
4.14	Samples of acquired roughness maps.	137
4.15	Samples of acquired maps and re-renderings.	139
5.1	Workflow of a generic MR catalog system.	150
5.2	Workflow of the virtual catalog of textiles system.	154
5.3	Scene setup for light interaction simulation.	155
5.4	Screen captures of the virtual catalog of textiles.	157
5.5	Virtual scene setup for light interaction simulation and rendering.	157

5.6	Previews of textile samples used for evaluation of the virtual catalog of textiles.	159
5.7	User ranking of the in-app textiles with respect to the real samples.	161
5.8	Examples of 3D face reconstruction results on images from the UMB-DB dataset.	173
5.9	Some samples used in the texture quality experiment.	175
5.10	User ranking of the 3D face reconstruction methods in the texture quality experiment.	176
5.11	Workflow of the 3D eyeglasses virtual try-on system.	177
5.12	Face size estimation workflow.	181
5.13	Example of keypoints for glasses fitting.	182
5.14	Eyeglasses pitch angle between ear and temple keypoints.	183
5.15	Examples of eyeglasses fitting using keypoints and face size.	184
5.16	Examples of virtual try-on results on images from the FFHQ dataset.	185
5.17	Screen captures of the virtual try-on web application user interface.	186
5.18	Hardware components of the proposed smart mirror.	197
5.19	A simple diagram showing the relationship between each software component.	199
5.20	The software components and their interactions.	199
5.21	Pages that the user can view.	201
A.1	Simplified modules diagram of SfM Flow.	211
B.1	Illustrative example of the software usage.	213
B.2	3D reconstruction evaluation.	214

List of Tables

3.1	Incremental SfM pipelines algorithm comparison.	29
3.2	Comparison of the most popular benchmark datasets and the proposed datasets.	31
3.3	Comparison of the proposed datasets.	52
3.4	List of available images sets for each object in the dataset. . .	56
3.5	Summary of the acquisition setup of each dataset in the ENRICH benchmark.	60
3.6	SfM cloud evaluation results.	72
3.7	SfM camera pose evaluation results.	73
3.8	MVS cloud evaluation results.	73
3.9	Pipelines execution times in seconds and memory usage. . . .	73
3.10	Bundle statistics and RMSE on GCPs and COPs for different SfM pipelines using the ENRICH-Statue dataset.	76
3.11	RMSE and statistics for the comparison of deep learning-based and hand-crafted local features with the ENRICH-Square dataset.	77
3.12	Effects on the Bicycle dataset.	82
3.13	Effects on the Empire Vase dataset.	82
3.14	Effects on the Hydrant dataset.	82
3.15	Effects on the Jeep dataset.	83
3.16	Effects on the Statue dataset.	83
3.17	RMSE of the planimetric, vertical, and global residuals on CPs with the eleven GCPs configurations.	87
3.18	Evaluation of the depth estimation on ENRICH-Square.	90
3.19	Evaluation of the depth estimation on ENRICH-Statue.	90
3.20	Evaluation of the depth estimation on ENRICH-Aerial.	90
4.1	Bill of material.	118
5.1	Comparison of the main features of virtual try-on applications.	149
5.2	System Usability Scale (SUS) results.	160

5.3	Application-specific question results.	161
5.4	Main characteristics of 3D face reconstruction methods in the state-of-the-art.	169
5.5	Evaluation results of 3D face reconstruction methods in the state-of-the-art.	170
5.6	Geometry evaluation results of 3D face reconstruction methods in the state-of-the-art.	172
5.7	System Usability Scale (SUS) results.	189
5.8	Application-specific question results.	189

1

Introduction

Presentation and communication of characteristics and knowledge about products and, in general, collections of objects or services are of crucial relevance in several fields. The most prominent application is given by the necessity of manufacturers and stores to advertise their offers to potential customers. To this purpose, paper or digital catalogs are often employed. Furthermore, catalogs may have goals that differ from the advertisement of physical products for sale. They may also build on the presentation of knowledge (e.g. in a library or museum) or refer to services provided by some company or individual.

Paper catalogs are among the older kind of ways to provide information about a collection of objects or services. They provide direct advertising of those elements by indicating their characteristics in a more or less exhaustive way which may include: pictures, textual description, options, prices, ordering information, payments, and delivery modalities. While we still find and use paper catalogs nowadays, new technological advances allow us to provide catalogs in digital forms. With the creation of the Internet and the diffusion of Personal Computers, we witnessed the development of digital twins of paper-printed catalogs and later the introduction of enriched digital experiences in such digital copies through new features, interactivity, and rich media presentation.

With the growing diffusion of smartphones, tablets, and head-mounted displays, new possibilities arise for the catalogs to be presented by taking

advantage of Mixed Reality (MR) concepts that allow the presentation and blending of virtual elements within the real world. To this end, this thesis proposes a computational framework that supports the creation of MR catalogs throughout the stages of design, development, and fruition of the catalog itself. The creation of MR catalogs involves several aspects. The key ones among them are investigated in this research. These include 3D geometry reconstruction, material appearance acquisition, and presentation of the MR catalog. Researchers have largely investigated these fields separately. This thesis aims to tackle specific problems of their usage in the creation of an MR catalog.

3D geometry reconstruction finds use in the creation of virtual 3D models of real elements for display in MR catalogs. Virtual elements may range from single objects, such as products, to buildings and entire cities, depending on the catalog content target. For this reason, different scales and levels of details of the 3D reconstruction may be necessary. Several 3D reconstruction solutions may be adopted as well. The research question is: how do we choose the best for our specific task?

In addition to 3D geometry, the look and feel of virtual elements are crucial to make them appear realistic. Material appearance acquisition is the task that allows us to capture the optical properties of surfaces and thus enabling their accurate reproduction in the virtual world. Precise acquisition of such properties requires complex and expensive hardware. The research challenge is: can we design an accurate enough but low cost and portable device for material appearance acquisition?

Finally, the presentation of the MR catalog involves the rendering of the virtual elements and interaction with the user. To accomplish this task, a generic framework for the creation of MR catalogs is defined. Such a framework embeds the concepts of 3D geometry reconstruction and material appearance acquisition. Considering that several kinds of MR catalogs can be created, each requiring specific abilities and characteristics, the framework acts as a blueprint customizable to fit specific needs. The framework is further explored and validated by investigating possible use case scenarios in the design and development of MR catalog applications. These include the

presentation of virtual textile products with realistic look-and-feel and the virtual try-on of eyeglasses.

This thesis discusses all of the above aspects and shows how it is possible to adapt methods and pre-existing research to support the creation of MR catalogs. Obtained results show that it is possible to tackle all of the challenges by providing cheap solutions. However, limitations, open problems, and future research directions arise as well.

1.1 Contributions

The main contribution of this research is the definition of a framework to aid the creation of mixed reality catalogs. However, in this thesis, several aspects of the design, development, and fruition of a Mixed Reality catalog are tackled. These include 3D geometry reconstruction, material appearance acquisition, rendering, and interaction. Each of them presents its research challenges that are addressed in this thesis. In more detail,

1. a method for the evaluation of 3D reconstruction pipelines that enables testing and stressing specific conditions and setups by using synthetic data is defined.
2. A software toolset for simplifying the creation of accurate synthetic data for 3D reconstruction evaluation is presented.
3. The aforementioned software is used to create different synthetic datasets and validate their usefulness in evaluating and comparing 3D reconstruction pipelines under several aspects.
4. A low cost device for material appearance acquisition of planar surfaces, which allows acquisition and modeling of the characteristics of real-world surfaces for later use in render engines, is proposed. The quality of the obtained appearance representations is also evaluated.
5. A generic framework for the creation of Mixed Reality catalogs that

embed methods for both the development and fruition of such catalogs is proposed. This framework includes the proposed 3D reconstruction evaluation and material appearance acquisition methods previously proposed. It also acts as a blueprint that can be customized to fit specific use cases.

6. Adaptations of the generic Mixed Reality catalog framework to specific use case scenarios through the design and development of catalog prototypes are proposed and evaluated. Those include a virtual catalog of textiles, an eyeglasses virtual try-on application, and a smart mirror that features the same virtual try-on application as a new means of interaction.

In an effort to support the reproducibility of results and encourage further research on the topics of this thesis, the software and datasets developed are publicly available online. Here below are summarized links to the resources.

- Software
 - SfM Flow – https://github.com/davidemarelli/sfm_flow
- Datasets
 - IVL-SYNTHSFM – <http://www.ivl.disco.unimib.it/activities/evaluating-the-performance-of-structure-from-motion-pipelines/>
 - IVL-SYNTHSFM-v2 – <https://doi.org/10.17632/fnxy8z8894>
 - ENRICH – <https://doi.org/10.17632/md7f7c5pzn> (to appear)

1.2 Organization of this thesis

This section details the structure of this thesis and the contents of each chapter. For each chapter are also listed the relevant publications in chronological order.

CHAPTER 1: INTRODUCTION

This is the current chapter, which introduces the content of this thesis, the challenges of the presented research, and the most relevant contributions. The rest of the chapter will outline the remainder of the document and its content.

CHAPTER 2: BACKGROUND AND RELATED WORK

This chapter presents the concepts and reasons behind the development of Mixed Reality catalogs. Existing types of catalogs, their characteristics, and issues are presented. The chapter also introduces the concepts of Virtual, Augmented, and Mixed Reality. It also discusses the research fields involved in the design, development, and fruition of Mixed Reality catalogs. Furthermore, the chapter reviews existing literature about Mixed Reality catalogs in their several forms and fields of application. For each task covered in the thesis, there will be a task-specific state-of-the-art review in the related chapter.

CHAPTER 3: 3D GEOMETRY RECONSTRUCTION

This chapter explores the use of 3D geometry reconstruction in the context of Mixed Reality catalogs. It reviews the Structure from Motion 3D reconstruction technique as well as existing 3D datasets. It then focuses on the definition of methods and datasets to simplify and speed up the selection of a 3D reconstruction pipeline that best suits a given task. A tool for generating accurate synthetic evaluation data and partially automating the procedure is also presented. Extensive experiments show the validity of the proposed synthetic datasets and their impact in evaluating 3D reconstructions.

The results and methods described in this chapter are based on the work published in the following papers:

- **Davide Marelli**, Simone Bianco, Luigi Celona, and Gianluigi Ciocca. “A Blender plug-in for comparing Structure from Motion pipelines”. In: *2018 IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*. 2018, pp. 1–5.

- Simone Bianco, Gianluigi Ciocca, and **Davide Marelli**. “Evaluating the Performance of Structure from Motion Pipelines”. In: *Journal of Imaging* 4.8 (2018).
- **Davide Marelli**, Simone Bianco, and Gianluigi Ciocca. “IVL-SYNTHSFM-v2: A synthetic dataset with exact ground truth for the evaluation of 3D reconstruction pipelines”. In: *Data in Brief* 29 (2020), p. 105041.
- **Davide Marelli**, Simone Bianco, and Gianluigi Ciocca. “SfM Flow: A comprehensive toolset for the evaluation of 3D reconstruction pipelines”. In: *SoftwareX* 17 (2022), p. 100931.
- **Davide Marelli**, Luca Morelli, Elisa Mariarosaria Farella, Simone Bianco, Gianluigi Ciocca, and Fabio Remondino. “ENRICH: multi-purposE datasets for beNchmaRking In Computer vision and pHo-togrammetry”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* (2023). (minor review).

CHAPTER 4: MATERIAL APPEARANCE ACQUISITION

This chapter focuses on the task of characterizing the optical properties of a surface that is complementary to the 3D geometry reconstruction. It reviews existing methods and models for material appearance acquisition. It then describes on the design and development of a low cost portable device for tackling such a task. The chapter presents the hardware and software component involved, as well as the evaluation of the obtained results.

Part of this work has been done with the supervision of Professor Alain Tremeau, Professor at University Jean Monnet, Laboratoire Hubert Curien, University of Saint-Etienne, during the (virtual) abroad period. The results of the collaboration will be presented in a joint journal paper that is currently in preparation and will be submitted in the next few months.

CHAPTER 5: RENDERING AND INTERACTION

This chapter presents the generic Mixed Reality catalog framework and

how the 3D reconstruction and material appearance acquisition fit it. The framework is responsible for both; the creation of the catalog and its fruition (rendering and interaction). The generic framework suits different kinds of Mixed Reality catalogs. It is also validated through the creation of two prototypes for different use cases; a virtual catalog of textiles and an eyeglasses virtual try-on application. Furthermore, this chapter presents the results of experimentation of eyeglasses virtual try-on on a prototype smart mirror which provides a new means of interaction.

The results and methods described in this chapter are based on the work published in the following papers:

- **Davide Marelli**, Simone Bianco, and Gianluigi Ciocca. “A Web Application for Glasses Virtual Try-on in 3D Space”. In: *2019 IEEE 23rd International Symposium on Consumer Technologies (ISCT)*. 2019, pp. 299–303.
- **Davide Marelli**, Simone Bianco, and Gianluigi Ciocca. “Faithful Fit, Markerless, 3D Eyeglasses Virtual Try-On”. In: *Pattern Recognition. ICPR International Workshops and Challenges*. Cham: Springer International Publishing, 2021, pp. 460–471.
- Simone Bianco, Luigi Celona, Gianluigi Ciocca, **Davide Marelli**, Paolo Napoletano, Stefano Yu, and Raimondo Schettini. “A Smart Mirror for Emotion Monitoring in Home Environments”. In: *Sensors* 21.22 (2021), p. 7453.
- **Davide Marelli**, Simone Bianco, and Gianluigi Ciocca. “Designing an AI-Based Virtual Try-On Web Application”. In: *Sensors* 22.10 (2022).

CHAPTER 6: CONCLUSION

Finally, this chapter concludes this thesis. Findings are highlighted and insights on possible future research that may spark from the results presented within this thesis are provided.

2

Background and related work

2.1 Background

A catalog provides direct advertising, presenting the products manufactured or sold by a firm, indicating their characteristics in a more or less exhaustive way (references, sizes, options, prices, payment, delivery, use, and aftersales service modalities). Furthermore, catalogs may have goals that differ from the advertisement of products for sale. They may also build on the presentation of knowledge (e.g. in a library or museum), provide virtual tours (e.g. of cultural heritage or buildings), or refer to services provided by some company or individual. In a world where advertising reaches customers in many ways, catalogs remain relevant tools in omnichannel marketing mixes [100] because consumers still use them.

Printed catalogs are now facing ecological concerns and new technological advances they can benefit from. Catalogs are increasingly taking digital forms, which allows for them to be displayed on multiple devices and reach customers in many forms. These range from digital copies of paper-printed catalogs to newer enriched digital experiences which are introduced in Section 2.1.1. These catalogs also benefit from the advantages of new technologies in terms of features, interactivity, and rich media presentation.

Research on printed catalogs is dense and covers several topics such as their design, impact on purchases, attractiveness, and consultation experiences [47, 70, 98, 100]. On the opposite, less research has been done on catalogs

that use innovative digital technologies providing features and experiences that were not possible with paper catalogs. As a result, the studies on printed catalogs appear somewhat outdated and not fully applicable to digital catalogs.

Although digital catalogs are not a new concept, new technologies are being integrated to provide those enriched experiences, and their adoption is now gaining momentum. Even IKEA in 2020 decided to drop its successful paper catalog [128]. IKEA started printing its catalog in 1951 and began to provide a digital version of it in 2000. The success of the catalog kept increasing till 2016 when it reached the peak of 200 million copies printed and distributed worldwide. But times have changed even for this successful media, and IKEA, as other companies, was hit by the change in media consumption and customer behaviors. To second this transformation they increasingly moved their investments toward the digital world and finally discontinued the paper catalog.

2.1.1 Introduction to e-catalogs

This section introduces e-catalogs which may appear in different variations thanks to the use of various technologies. The most direct transposition of the traditional paper-printed catalog is its digital or online version. The online catalog is thus a digital version of the paper catalog; it retains the same characteristics (e.g., layout, information, pages to turn), making them familiar to people used to paper-printed catalogs. It is thus an immaterial reproduction of the real-world paper catalog being its virtual twin. While this implies that the digital catalog is a simple transposition of the paper catalog, it can also be considered as an amplification of the real world and thus of the traditional catalog [157]. Thus, this creates the potential to provide new, enhanced, and rich experiences.

A new kind of catalog exploits these new possibilities and provides enriched experiences: the enriched digital catalog. It is still a digital catalog but also provides augmented content and richness by adding new information and

exploiting modern technologies for visualization. The quality of the catalog is thus improved by adding detailed and useful information that could not be included in printed catalogs (due to limits of the paper catalog in both layout and media type), but that becomes available in digital forms. These details include product and commercial information, informative or promotional videos about the product or the brand, and redirection toward a webpage or product sheet. The enriched catalog aims thus to provide an immersive and compelling experience. However, a study by Garnier and Poncin [91] shows that the presence of enrichment elements does favor immersion but does not create a compelling experience nor increase intentions to use the digital catalog. Nonetheless, enriched catalogs still solve one of the problems of digital ones; they offer a means to directly buy the product, whereas traditional catalogs require consumers to use another channel (e.g., website, physical store) to make a purchase.

In recent years newer technologies in the field of Mixed Reality (MR, which is explained in Section 2.1.2) have been increasingly used to create product catalogs. The goal is to provide more immersive and compelling experiences than enriched catalogs. However, MR technologies are usually employed to extend the functionalities of enriched catalogs. MR can be thus exploited to provide additional capabilities, such as virtually showing a product in a real-world environment or allowing a user to check the appearance of a wearable product on himself (i.e. the Virtual Try-On use case). This can be accomplished in different ways involving a variable degree of blending between real and virtual-world elements. While this blending allows unprecedented experiences for the users, the technology is still under heavy development, and several problems need to be tackled. Common issues are related to the reproduction of physical characteristics of the virtual elements, such as material properties and sizes. These problems usually make the MR catalog experience more entertaining than functional [147].

Nevertheless, it is also to consider that each type of catalog described in this section can be seen as an extension of its predecessor. For example, the enriched catalog extends the digital one by adding new content; the MR catalog adds new features to ones already provided by the enriched

catalog. Furthermore, in a world where advertising reaches customers through omnichannel strategies, it is also possible to see an integration of digital, enriched, and MR catalogs in other types of media as well as in paper-printed catalogs. This integration still presents some challenges since only 17% of consumers seek enrichment through QR codes leading to complementary information, and the remaining 83% feel indifferent or even annoyed by them [91].

Finally, it has also to be observed that these new catalogs eliminate printing and mailing costs but require substantial investments to create the catalogs themselves and the distribution platform.

2.1.2 Introduction to Mixed Reality

The kind of catalogs presented in the previous section make use of technologies that allow to blend together elements of the real and virtual worlds. This blending can be achieved in a variety of ways and in a broad spectrum: from a fully real world to a fully virtual world. This is known as the concept of *Reality-Virtuality Continuum* that was first introduced in 1994 by Milgram et al. [191]. Although concepts and keywords of Virtual and Augmented Reality started to emerge earlier [280, 286], the work of Milgram et al. is the first to provide a clear definition of this continuum, the concepts, and keywords. The proposed Reality-Virtuality Continuum is schematized in Figure 2.1. In this continuum, the real environment and the virtual one are placed at the left and right ends of the spectrum respectively. The real environment consists solely of real objects and includes elements that might be observed when viewing a real-world scene either directly in person, through some kind of window, or via some sort of display. The virtual environment consists solely of virtual objects, such as computer graphic simulations, either monitor-based or immersive. Everything that blends these two environments, with various degrees of balance between real and virtual, ends in the macro-category of the Mixed Reality (MR) environment. Common concepts in this range are Virtual Reality (VR) and Augmented Reality (AR).

A VR environment is one in which the observer is fully immersed in a virtual world. This synthetic world may or may not stick to concepts of the real-world such as physics, time, and material properties. It also may be a virtual copy of a real environment or a fictional one. VR is closely related to completely virtual environments, but in those environments, it is possible to include elements of the real world, thus moving towards the real environment.

Instead, AR presents more real elements than virtual ones. Virtual elements are thus usually superimposed onto directly viewed real scenes through panel-mounted or head-mounted displays (HMDs). Those displays provide the possibility to see the real world through them. While this is the state-of-the-art way to provide an augmented reality experience, new technologies allow the creation of monitor-based AR systems. These are non-fully immersive systems that still provide a "window-on-the-world" (WoW). The concept of AR still applies if computer-generated images are overlaid onto live or stored real-world content. Following this reasoning, smartphones have been recently used as HMDs and handheld displays to provide AR systems by superimposing virtual content on the live video feed of the onboard video camera [105, 211]. An MR system is thus defined by several factors that include but are not limited to the hardware and technology used to provide the experience (which defines immersion and directness of view of the real world) and the ratio of real/virtual world elements (which determines the reality or virtuality).

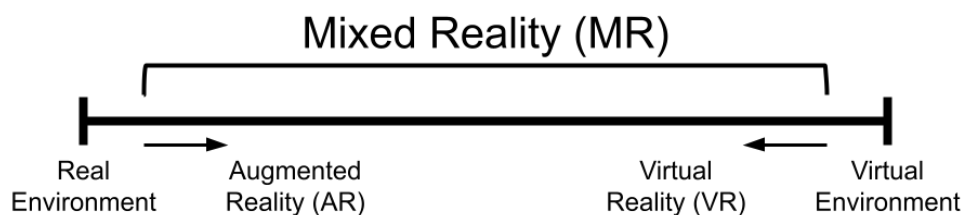


Figure 2.1: Reality-Virtuality Continuum.

2.1.3 Fields involved in a Mixed Reality catalog

A Mixed Reality catalog involves methods and techniques from several fields. Those can be additionally split into fields required for its development and its fruition. Those fields include 3D geometry reconstruction and modeling to provide 3D geometry of elements either of the real or virtual world, as well as material appearance acquisition to represent the material properties of such geometries. For the fruition of a mixed reality catalog rendering and interaction cover a crucial role. While this mainly focuses on software components such as rendering engines and computer vision, the hardware used to experience the catalog is of equal importance.

This thesis focuses on the software and hardware required for the acquisition of 3D objects and their materials, as well as the software platform required for the fruition of a mixed reality catalog.

2.2 Related work

This section reports a high-level state-of-the-art review of the works and methods used to create a mixed reality catalog. A detailed state-of-the-art review describing the existing literature for each investigated task and use case is provided in each chapter.

Mixed reality catalogs can dramatically change the way a user interacts with a product or environment. Such a technology shift finds usage in the form of several applications. Here are reviewed MR systems in the fields of interest.

AR product configurators make product visualizations accessible and configurable. A configurator is a software tool that allows the configuration of a product based on a list of possible customizations; depending on the complexity of the product, constraints on possible configurations may apply. Customization is performed using 3D models in AR that allow performing this task by superimposing the personalized product on the real one. This kind of

configurator finds application in several domains, such as apparel and fashion items, furniture, real estate, cars, and luxury objects. Gehring et al. [97] presented an early work that explored the use of an AR mobile application for the on-the-fly customization of products at the point of sale. The application allows the user to change the color of a soap dispenser. The customization works based on two possible interaction schemes. In the first scheme, the user picks a color directly; in the second one, the most fitting color is automatically computed from the environment. Thus, the app can dynamically adapt the product based on both user and environmental constraints. Another early concept of Gehring [96] used a smartphone to track a real object in the video feed of the camera and provide a customized version of it on the display of the device. Furthermore, the same work proposed the use of a phone with an integrated pico-projector to project a custom personalized version of a box over a real completely white box. Wiwatwattana et al. [314] presented a prototype for ice dessert customization and ordering, which comprises an iOS AR application (customer interface) and a web application (store interface). The ice flavors and toppings are augmented over the real cone/cup by the customer himself. The customer can see what he will get with his order as it is shown in real-time on the mobile screen before he can then order the preferred combination from the smartphone, and the shop assistant will receive it through the web application. Recently, Gottschalk et al. [107] proposed an AR configurator that splits the products into sub-components and allows users to customize them. While the concept can be applied to multiple products, they presented a case study in which a manufacturer of modular kitchens can provide the ability to order customized kitchens for their customers. The application considers product requirements, user needs, and environmental constraints, assuring that none is violated.

MR virtual tours are closely related to AR configurators. They still make product visualization accessible through Mixed Reality but are missing the ability to personalize and configure the objects. Virtual tours find application in the visualization of single objects (e.g., apparel, fashion items, cars, furniture) as well as complete environments (e.g., real estate, theme parks). Back

in 1987 Brooks [40] proposed a VR tour of buildings, allowing an architect to show a building prototype to the client and enabling discussion and design iteration about it. This application was mainly limited by the technology of its time, which allowed it to achieve only nine updates per second of the generated virtual environment. Thanks to technological advances, newer applications have been proposed. ARQuake (Thomas et al. [285]) was developed as an extension to the popular Quake desktop game. The application exploits AR and a see-through HMD to show game elements (monsters, weapons, objects of interest) in indoor and outdoor spaces. Interestingly, the application was also used as an AR visualization tool to aid architects in conveying their ideas to their clients. The application made it possible to show the building design in an outdoor AR environment, which allows users to discover issues and present alternative designs. Chhugani et al. [54] introduced visibility computation and data organization algorithms that enable high-fidelity walkthroughs of large 3D scenes. Their walkthrough system performs computation proportional only to the required detail in visible geometry at the rendering time, allowing them to build large virtual tours with higher framerates. The Arch-Explore user interface [42] tackled the problem of VR exploration of architectural 3D models in the limited interaction space of small laboratory setups. Du et al. [68] extended the virtual tour of buildings allowing multiple users to interact in the virtual environment at the same time.

Augmented and Virtual shops provide additional content in a physical store or in a virtual environment in which the customers can discover and interact with products in new ways. In an augmented shop, AR is used to provide additional information about products. A virtual shop may provide 3D real-size visualization of the products allowing the user to interact with them and discover alternative products. These new shops bring consumers an immersive online shopping experience. Guven et al. [112] presented a mobile AR application that extends such social content from the computer monitor into the physical world through mobile phones. The application provides information on products (by superimposition) to help in buying decisions. In detail, online or in-store reviews are provided, and the user can interact with

reviewers via text or voice using virtual avatars. ShelfTorchlight [172] uses a mobile phone and a pico-projector to help the user to find a product on a shelf in a store. The authors demonstrated the usability of the device in two different scenarios: a library and a retail environment. While searching for a book in a library, the system shows customer reviews for the books by highlighting a book and projecting the related review score. In the retail environment, the device takes user preferences to highlight products that fit them better. Rashid et al. [239] propose the use in conjunction of AR and RFID technology to provide augmentation of products on a smart shelf. A single marker is used to trigger an AR application in front of the shelf. The shelf provides the application with the approximate position and type of objects in the AR app thanks to the RFID technology. The application can thus superimpose additional information. Furthermore, interaction with the system is both visual and vocal. In 2016 Alibaba launched Buy+, a new virtual shop that uses VR technology to create a 3D shopping environment and provide virtual interaction with products sold by the platform. Later, Ouellet et al. [218] demonstrated the feasibility and validity of a virtual shop experience. They created a virtual shop that measures approximately 3.5x6.5m and contains common shop elements. Its layout comprises two central shelves and three refrigerators in a 3D virtual environment that is fully immersive and usable through HMDs. It supports navigation (via natural walking), item selection (via a pointer), and conversation with a character (via natural talking). However, the objective of their research was the assessment of everyday memory, not the creation of a VR shop.

MR tourist guides allow users to obtain information about the objects in their close surroundings ranging from objects in a museum or historical buildings of a city. The relevant information is superimposed based on the object in front of the camera. This finds application in the tourism and cultural heritage industries. An early example of an AR museum guide is a prototype of an automated audio tour guide proposed in 1995 by Bederson [23]. The system allows visitors to hear descriptions of exhibited pieces just by walking up to them. Descriptions may be heard in any order and can be cut

short by walking away. Mase et al. [185] proposed another early prototype, the Meta-Museum. This prototype blends virtual reality and artificial intelligence technologies with conventional museums to maximize the utilization of a museum's archives and knowledge base to provide an interactive experience for visitors. The system can present additional details about the exhibits through HMDs and personalizes the contents based on the visitor's interests. Hall et al. [113] propose to use MR to enhance user experience and learning in museums. The system uses a sophisticated portable MR device called the periscope inside exhibits to allow visitor interaction and visualization of artifacts and their related information. The Archeoguide system [303] allows users to see virtually restored cultural assets by superimposing a virtual 3D object on the real-world through HMDs. The system uses a back-end server, mobile units (HMDs and PDAs), a wireless network, and a tracking system (based on GPS location and imaging). The ARCO system [313] provides a complete tool chain for virtual museum environments. It supports the digitization of a museum collection through photogrammetry, collection refinement, management, and VR/AR visualization of galleries and artifacts. Liarokapis and White [164] experimented with the use of AR to visualize incomplete or broken real objects as they were in their original state. The result is achieved by the superimposition of the missing parts through 3D reconstruction and virtual model enhancement. Holz et al. [125] introduced the Mixed Reality Agent Guide (MIRA) guidance system, which combines HMDs and robots in museum rooms. MIRA guides visitors to aim AR devices toward markers placed in the museum to make the system recognize them and present virtual content. Miyashita et al. [193] proposed an AR museum guide that embeds AR content as well as audio guidance. AR mode is activated based on some AR stations inside the museum. The device enables the AR mode when the visitor is near an AR station. Otherwise, the visitor is guided through the exhibition by a voice guide. Sinthanayothin et al. [269] proposed an interactive virtual three-dimensional photo gallery on mobile devices. The application allows users to take pictures with their mobile device and exhibit them as a virtual 3D gallery and navigate or walk through the gallery by pressing a button or moving the device. Shih [267] created a virtual tour

of the city of London by integrating the ability to see tourists' avatars and interact with them in Google Street Map. The virtual tour is guided by a native English-speaking instructor. Soga [271] developed two systems to support exhibits at a museum with additional content. The first one is a walk-through system in 3D space using Kinect; the latter uses goggles to show a stereoscopic 3D image. Interaction is supported by gesture recognition. Recently, Hammady et al. [115] designed and developed MuseumEye, an HMD-based museum guidance system based on the immersion and presence theory. This approach examines the influence of interactivity, spatial mobility, and perceptual awareness of individuals within MR environments to enhance customer experience and reduce the number of human tour guides in museums.

MR publishing features refers to the integration of Augmented Reality with paper-printed documents. Using an AR-ready device over the surface of a printed document allows for its augmentation with virtual objects. It allows for the addition of dynamic content in newspapers, paper catalogs, advertisement flyers, books, and games. Löchtefeld et al. [171] proposed the use of a mobile phone and a portable projector to detect markers and display additional information. The system is suitable for different tasks, including providing digital navigable twins of paper-printed content (e.g. maps) and augmenting books with personalized content. Löchtefeld et al. [173] investigated the use of AR to bridge the digital divide on advertising leaflets. In particular, they explored the use of AR to easily compare products of different retailers and different strategies for visualizing cross-selling recommendations inside the leaflets. Kim and Kim [148] created a markerless augmented reality application system developed that uses leaflets and outdoor signboards of stores to augment the three-dimensional model-based indoor information and booking information of stores and makes possible users' direct booking. Zulkifli et al. [340] proposed a mobile AR application to complement a paper brochure. Through the application, users can access information in the form of virtual content which cannot be acquired from a typical paper brochure.

Virtual try-on allows customers to virtually try a product on themselves. Several types of virtual try-on have been proposed over the years, including the

try-on of eyeglasses, clothes, accessories, jewelry, makeup, hairstyle, and hair color. The try-on may appear on a real-time video feed or in a deferred manner on video, images, or 3D models. Eisert et al. [72] proposed an early concept of virtual try-on for shoes. The system uses a camera to acquire a video feed of the user's shoes and a display on which the result of the try-on is displayed. Sophisticated 3D imaging techniques are used to virtually wear the shoes on the user, track them in real-time, and adapt the rendering to the motion of the user. Later several applications and approaches have been proposed for the virtual try-on of different products [66, 145, 169, 227, 250, 293]. An in-depth review of existing virtual try-on applications is provided in Section 5.1.1.

3

3D geometry reconstruction

Three-dimensional geometry reconstruction is the process that allows to capture the geometry of an object or an entire scene. In the last years, interest has developed in the use of 3D reconstruction for reality capture, gaming, and virtual and augmented reality. These techniques have been used to create video game assets [114, 232], virtual tours [254] as well as mobile 3D reconstruction apps [73, 200, 209]. Some other areas in which 3D reconstruction can be used are Computer Aided Design (CAD) software [26], computer graphics and animation [132, 274], medical imaging [45], virtual and augmented reality [210], cultural heritage [241].

In the context of Mixed Reality catalogs 3D geometry reconstruction is employed in the creation of virtual 3D models of real elements for display in virtual or augmented reality. The virtual elements may range from single objects such as products (e.g., AR configurators, AR publishing features), and buildings (e.g., virtual tours), to entire cities (e.g. tourists guides). For this reason, different scales and levels of details of the 3D reconstruction may be necessary. In the literature there are several benchmark datasets that can be used to evaluate reconstruction algorithms. However, not all these datasets comprise the level of details, scales, and challenges necessary for Mixed Reality catalogs.

This chapter defines a method for the generation of synthetic datasets that can be used to evaluate different state-of-the-art 3D reconstruction pipelines. This method has been used to generate sample datasets to test different

aspects of the reconstruction pipelines. The data itself and the evaluation procedure are presented as well. Having real scenes as reference models is not trivial; thus, a software toolset has been developed to create an evaluation dataset starting from synthetic 3D scenes. This allows us to rapidly and efficiently create datasets with different subjects and levels of scales stressing the pipelines under various conditions and comparing them to select the best solution for a specific MR catalog.

Over the years, a variety of techniques and algorithms for 3D reconstruction have been developed to meet different needs in various fields of application ranging from active methods that require the use of special equipment to capture geometry information (e.g., laser scanners, structured lights, microwaves, ultrasound) to passive methods that are based on optical imaging techniques only. The latter techniques do not require special devices or equipment and are thus easily applicable in different contexts. Among the passive techniques for 3D reconstruction there is the Structure from Motion (SfM) pipeline [32]. As shown in Figure 3.1, given a set of images acquired from different observation points, it recovers the pose of the camera for each input image and a three-dimensional reconstruction of the scene in the form of a sparse point cloud. After this first sparse reconstruction, it is possible to run a dense reconstruction phase using Multi-View Stereo (MVS) [85].

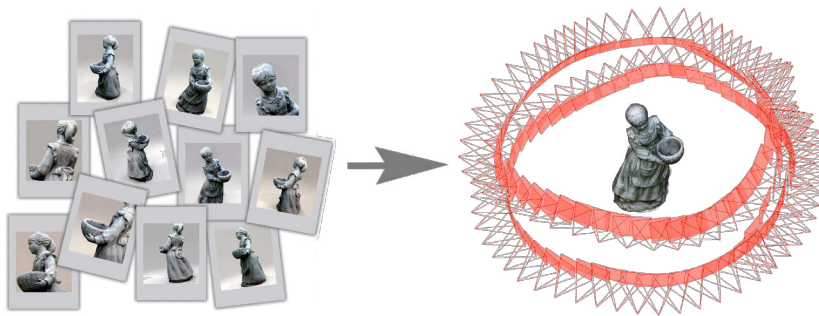


Figure 3.1: 3D reconstruction using Structure from Motion.

In recent years deep learning-based methods for 3D reconstruction have also been researched. They are usually constrained to specific reconstruction tasks or single subjects [331]. For this reason this chapter considers the SfM pipeline which presents a generic and suitable tool for several MR catalogs.

However, deep learning has also been used for some of the processing in the SfM pipeline (e.g. local feature extraction), and such techniques are thus considered in the experiments of this chapter. Deep learning-based methods are also later investigated for the Virtual Try-On use case (Section 5.3.1).

The chapter is organized as follows. Section 3.2 reviews the SfM 3D reconstruction method as well as the most relevant 3D benchmark datasets. Section 3.1 explains the usefulness and challenges of 3D reconstruction in the field of MR catalogs. Section 3.3 defines a common method for the evaluation of different 3D reconstructions. Section 3.4 presents a toolset for the creation of 3D reconstruction synthetic benchmarking datasets. Section 3.5 describes the proposed synthetic datasets. Finally, Section 3.6 proves the usefulness and quality of the proposed datasets and the method of their generation by using them to evaluate different phases or tasks involved in the 3D reconstruction process.

3.1 Challenges of 3D reconstruction in Mixed Reality catalogs

3D geometry reconstruction is a topic of interest in Computer Vision and Photogrammetry. For this reason, several pipelines and datasets for 3D reconstruction exist in the literature. However, datasets present specific acquisition setups and conditions, and not all 3D reconstruction methods are suitable for individual cases. In the creation of a Mixed Reality catalog, 3D geometry reconstruction covers a key topic, and being able to select the correct 3D reconstruction solution is crucial. 3D reconstruction may be employed to provide 3D visualization of objects to sell; it can also be handy in AR where virtual elements are blended with real environments (e.g. to understand placement and occlusions). Furthermore, depending on the context of the virtual catalog, different scales and levels of detail may be required in different situations. For example, an MR catalog of single objects may require reconstructing the object itself to be shown or a whole

environment. This environment can again be a single room or an outdoor scene ranging from a single location to an entire city. Another consideration is the modality of availability of such reconstruction. It can happen in an offline way where processing times are not relevant (e.g. a catalog of products) or in an online way where processing times directly impact the usability of the catalog (e.g. virtual try-on). For all of these reasons being able to evaluate, compare, and select the most suitable 3D reconstruction technique and method is important for MR catalogs.

3.2 Related work

This section presents a literature review of the commonly adopted 3D reconstruction technique of Structure from Motion, as well as the most relevant datasets for 3D reconstruction.

3.2.1 Structure from Motion

The SfM pipeline allows the reconstruction of three-dimensional structures starting from a series of images acquired from different observation points. As it can be seen from Figure 3.2, a typical SfM pipeline comprises different processing steps, each of which tackles a different problem. Each step can exploit different algorithms to solve the problem at hand, and thus many different SfM pipelines can be built.

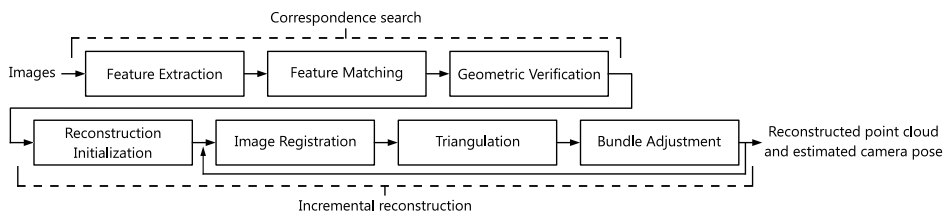


Figure 3.2: Incremental Structure from Motion pipeline.

In particular, incremental SfM is a sequential pipeline that consists of

a first phase of correspondence search between images and a second phase of iterative incremental reconstruction. The correspondence search phase is composed of three sequential steps: *Feature Extraction*, *Feature Matching*, and *Geometric Verification*. This phase takes as input the image set and generates as output the so-called *Scene Graph* (or *View Graph*) that represents relations between geometrically verified images. The iterative reconstruction phase is composed of an initialization step followed by three reconstruction steps: *Image Registration*, *Triangulation*, and *Bundle Adjustment*. Using the scene graph, it generates an estimation of the camera pose for each image and a 3D reconstruction as a sparse point cloud.

3.2.1.1 SfM building blocks

This section describes the building blocks of a typical incremental SfM pipeline illustrating the problem that each of them addresses and the possible solutions exploited.

Feature Extraction: For each image given in input to the pipeline, a collection of local features is created to describe the points of interest of the image (keypoints). For feature extraction different solutions are available; the choice of the algorithm influences the robustness of the features and the efficiency of the matching phase. Once keypoints and their description are obtained, the correspondences of these points in different images can be searched by the next step.

Feature Matching: The keypoints and features obtained through Feature Extraction are used to determine which images portray common parts of the scene and are therefore at least partially overlapping. If two points in different images have the same description, then those points can be considered as being the same in the scene with respect to the appearance. If two images have a set of points in common, then it is possible to state that they portray a common part of the scene. Different strategies can be used to efficiently compute matches between images; solutions adopted by SfM implementations are reported in Table 3.1. The output of this phase is

a set of images overlapping at least in pairs and the set of correspondences between features.

Geometric Verification: This phase of analysis is necessary because the previous matching phase only verifies that pairs of images apparently have points in common; it is not guaranteed that found matches are actual correspondences of 3D points in the scene, and outliers could be included. It is necessary to find a geometric transformation that correctly maps a sufficient number of points in common between two images. If this happens, the two images are considered geometrically verified, thus meaning that the points are also corresponding to the geometry of the scene. Depending on the spatial configuration used to acquire the images, different methods can be used to describe their geometric relationship. A homography can be used to describe the transformation between two images of a camera that acquires a planar scene. Instead, the epipolar geometry allows the description of the movement of a camera through the essential matrix E if the intrinsic calibration parameters of the camera are known; alternatively, if the parameters are unknown, it is possible to use the uncalibrated fundamental matrix F . Algorithms used for geometric verification are reported in Table 3.1. Since the correspondences obtained from the matching phase are often contaminated by outliers, it is necessary to use robust estimation techniques such as RANSAC (RANdom SAmple Consensus) [80] during the geometry verification process [120, 196]. Instead of RANSAC, some of its optimizations can be used to reduce execution times. Refer to Table 3.1 for a list of possible robust estimation methods. The output of this phase of the pipeline is the so-called *Scene Graph*, a graph whose nodes represent images and edges join the pairs of images that are considered geometrically verified.

Reconstruction Initialization: The initialization of the incremental reconstruction is an important phase because a bad initialization leads to a poor reconstruction of the 3D model. To obtain a good reconstruction it is preferable to start from a dense region of the scene graph so that the redundancy of the correspondences provides a solid base. If the process starts from an area with few images, the Bundle Adjustment process does not have sufficient information to refine the position of the reconstructed camera poses

and points; this leads to an accumulation of errors and a bad final result. For the initialization of the reconstruction, a pair of geometrically verified images is chosen in a dense area of the scene graph. If more than one pair of images can be used as a starting point, the one with the most geometrically verified matching points is chosen. The points in common with the two images are used as the first points of the reconstructed cloud; they are also used to establish the pose of the first two cameras. Subsequently, the *Image Registration*, *Triangulation*, and *Bundle Adjustment* steps add iteratively new points to the reconstruction considering a new image at a time.

Image Registration: Image registration is the first step of incremental reconstruction. In this phase a new image is added to the reconstruction and is thus identified as *registered image*. For the newly registered image the pose of the camera (position and rotation) that has acquired it must be calculated; this can be achieved using the correspondence with the known 3D points of the reconstruction. Therefore, this step takes advantage of the 2D-3D correspondence between the keypoints of the newly added image and the 3D reconstruction points that are associated with the keypoints of the previously registered images. To estimate the camera pose it is necessary to define the position in terms of 3D coordinates of the reference world coordinates system and the rotation (pitch, roll, and yaw axes), for a total of six degrees of freedom. This is possible by solving the *Perspective-n-Point* (PnP) problem. Various algorithms can be used to solve the PnP problem (see Table 3.1). Often outliers are present in the 2D-3D correspondences; the aforementioned algorithms are used in conjunction with RANSAC (or its variants) to obtain a robust estimate of the camera pose. The newly recorded image has not yet contributed to the addition of new points; this will be done by the triangulation phase.

Triangulation: The previous step identifies a new image that certainly observes points in common with the 3D point cloud reconstructed so far. The new registered image may observe further new points that can be added to the 3D reconstruction if they are observed by at least one previously registered image. A triangulation process defines the 3D coordinates of the new points that can be added to the reconstruction and thus generate a more dense point

cloud. The triangulation phase takes a pair of registered images with points in common to estimate the respective camera poses. Then, it tries to estimate the 3D coordinates of each point in common between the two images. To solve the problem of triangulation, an epipolar constraint is placed. It is necessary that the positions from which the images were acquired allow identification of the position of acquisition of the counterpart in the image; these points are called epipoles. In the ideal case, it is possible to use the epipolar lines to define the epipolar plane on which lies the point whose position is to be estimated. However, because of the inaccuracies in the previous phases of the pipeline, it is possible that the point does not lie in the exact intersection of the epipolar lines; this error is known as a reprojection error. To solve this problem, special algorithms that take into account the inaccuracy are necessary. Algorithms used by SfM pipelines are listed in Table 3.1.

Bundle Adjustment: Since the estimation of camera poses and the triangulation can generate inaccuracies in the reconstruction it is necessary to adopt a method to minimize the accumulation of such errors. The purpose of the *Bundle Adjustment* (BA) [292] phase is to prevent inaccuracies in the estimation of the camera pose to propagate in the triangulation of cloud's points and vice versa. BA can therefore be formulated as the refinement of the reconstruction that produces optimal values for the 3D reconstructed points and the calibration parameters of the cameras. The algorithm used for BA is Levenberg-Marquardt (LM), also known as Damped Least-Squares; it allows the resolution of the least squares method for the non-linear case. Various implementations exist as shown in Table 3.1. This phase has a high computational cost and must be executed for each image that is added to the reconstruction. To reduce processing time BA can be executed only locally (i.e., only for a small number of images/cameras, the most connected ones); BA is executed globally on all images only when the rebuilt point cloud has grown by at least a certain percentage since the last time global BA was made.

3.2.1.2 Incremental SfM pipelines

Over the years many different implementations of the SfM pipeline were proposed. This subsection focuses on the most popular ones with publicly available source code that could allow customization of the pipeline itself. Among the available noteworthy pipelines are COLMAP, Theia, OpenMVG, VisualSFM, Bundler, and MVE. The following briefly describes each pipeline, while Table 3.1 details their implementations with the algorithms used in each processing block.

COLMAP [259]—is an open-source implementation of the incremental SfM and MVS pipeline. The main objective of its creators is to provide a general-purpose solution usable to reconstruct any scene introducing enhancements in robustness, accuracy, and scalability. The C++ implementation also comes with an intuitive graphical interface that allows the configuration of pipeline parameters. It is also possible to export the sparse reconstruction for different MVS pipelines.

Theia [282]—is an incremental and global SfM open-source library. It includes many algorithms commonly used for feature detection, matching, pose estimation, and 3D reconstruction. Furthermore, it is possible to extend the library with new algorithms using its software interfaces. Implementation is in the form of a C++ library. Obtained sparse reconstruction can be exported into Bundler or VisualSFM NVM file format that can be used by most MVS pipelines.

OpenMVG [197]—is an open-source library to solve Multiple View Geometry problems. It provides an implementation of the SfM pipeline for both incremental and global cases. Different options are provided for feature detection, matching, pose estimation, and 3D reconstruction. It is also possible to use geographic data and GPS coordinates for the pose estimation phase. The library is written in C++ and can be included in a bigger project or can be compiled in multiple executables, each one for a specific set of algorithms. Sample code to run SfM is also included. Sparse reconstruction can be exported in different file formats for different MVS pipelines.

VisualSFM [317]—implements the incremental SfM pipeline. Compared

to other solutions, this is less flexible because only one set of algorithms can be used to make reconstructions. The software provides an intuitive graphical user interface that allows SfM configuration and execution. Reconstructions can be exported in VisualSfM’s NVM format or Bundler format. It is also possible to execute the dense reconstruction steps using CMVS/PMVS directly from the user interface.

Bundler [270]—is one of the first incremental SfM pipeline implementations of success. It also defines a Bundler ‘out’ format that is commonly used as an exchange file between SfM and MVS pipelines.

MVE [83] (Multi-View Environment)—is an incremental SfM implementation. It is designed to allow multi-scale scene reconstruction, comes with a graphical user interface, and also includes an MVS pipeline implementation.

Linear SfM [335]—is a new approach to the SfM reconstruction that decouples the linear and nonlinear components. The proposed algorithm starts with small reconstructions based on Bundle Adjustment that is afterward joined hierarchically.

Table 3.1: Incremental SfM pipelines algorithm comparison.

	Feature Extraction	Feature Matching	Geometric Verification	Image Registration	Triangulation	Bundle Adjustment	Robust Estimation
COLMAP	SIFT [176]	Exhaustive	4 Point for Homography [120]	P3P [88]	sampling-based DLT [259]	Multicore BA [318]	RANSAC [80]
		Sequential	5 Point Relative Pose [277]	EPnP [156]		Ceres Solver [4]	PROSAC [55]
		Vocabulary Tree [260]	7 Point for F-matrix [120]				LO-RANSAC [56]
		Spatial [259]	8 Point for F-matrix [120]				
		Transitive [259]					
OpenMVG	SIFT [176]	Brute force	affine transformation	6 Point DLT [120]	linear (DLT) [120]	Ceres Solver [4]	Max-Consensus
	AKAZE [8]	ANN [198]	4 Point for Homography [120]	P3P [88]			RANSAC [80]
		Cascade Hashing [53]	8 Point for F-matrix [120]	EPnP [156]			LMed [246]
			7 Point for F-matrix [120]				AC-Ransac [195]
		5 Point Relative Pose [277]					
Theia	SIFT [176]	Brute force	4 Point for Homography [120]	P3P [88]	linear (DLT) [120]	Ceres Solver [4]	RANSAC [80]
		Cascade Hashing [53]	5 Point Relative Pose [277]	PNP (DLS) [123]	2-view [166]		PROSAC [55]
			8 Point for F-matrix [120]	P4P [43]	Midpoint [121]		Arrsac [235]
				P5P [151]	N-view [120]		Evsac [81]
						LMed [246]	
VisualSfM	SIFT [176]	Exhaustive Sequential Preemptive [317]	n/a	n/a	n/a	Multicore BA [318]	RANSAC [80]
Bundler	SIFT [176]	ANN [13]	8 Point for F-matrix [120]	DLT based [120]	N-view [120]	SBA [175] Ceres Solver [4]	RANSAC [80]
MVE	SIFT [176] + SURF [20]	Low-res + exhaustive [83] Cascade Hashing	8 Point for F-matrix [120]	P3P [88]	linear (DLT) [120]	own LM BA	RANSAC [80]

Commercial software also exists, usually providing full implementations that allow sparse and dense reconstruction. Some examples are Agisoft PhotoScan and Metashape¹, Capturing Reality RealityCapture², Autodesk

¹<http://www.agisoft.com/>

²<https://www.capturingreality.com/>

ReCap³.

3.2.2 3D reconstruction datasets

Over the past two decades, the need to evaluate new computer vision and photogrammetry algorithms have motivated the research community toward the creation of 2D and 3D datasets for different scenarios (e.g., indoor, outdoor, laboratory, urban, buildings) and tasks (such as image matching, image retrieval, structure from motion, and SLAM).

As a tool for scientists, a benchmark dataset is a set of data to evaluate or compare the performance of sensors, platforms, or processing algorithms [16] against high-quality and accurate ground truth. Nonetheless, the acquisition of a sufficient amount of data is still a challenge. Barriers to their realization are related to the costs and time for obtaining data diversity (like scale or scene) and the collection of precise annotations and accurate and reliable ground truth.

Driven by the achievements and pending research issues in the 3D reconstruction sector [24, 242], many scientific initiatives have been recently proposed to evaluate the current status of available processing methods while boosting further investigations, both in the photogrammetric and computer vision research fields. These activities have encouraged developers and users to deliver comparative performance analyses focusing, in particular, on image-based 3D reconstruction [32, 139, 261].

In recent years, advanced deep learning-based algorithms for extracting complex information and features from visual data have been effectively applied in many domains and scenarios, such as image analysis, remote sensing, computer vision, and geoscience research, outperforming standard approaches or opening to new applications not possible with classical methods. However, applying these techniques implies the availability of a large amount of data for training the underlying models. Moreover, real data collected and targeted for a given task are rarely complete and heterogeneous enough (in

³<https://www.autodesk.com/products/recap/overview>

Table 3.2: Comparison of the most popular benchmark datasets and the proposed datasets (in bold).

Name	Setting	Real/Synthetic	Resolution	Image format	Videos	SfM	SLAM	Stereo	MVS	Depth	GCP	Lighting
Middlebury stereo	Indoor	Real	2-6 Mpx	fixed	-	-	-	✓	-	✓	-	varying
Middlebury MVS	Lab	Real	0.3 Mpx	fixed	-	✓	-	-	✓	-	-	fixed
DTU	Lab	Real	2 Mpx	fixed	-	✓	-	-	✓	-	-	varying
KITTI	Streets	Real	0.5 Mpx	fixed	✓	-	✓	✓	✓	✓	-	fixed
Strecha	Monuments	Real	6 Mpx	fixed	-	✓	-	-	✓	-	-	fixed
Tanks and Temples	Indoor/Outdoor	Real	8 Mpx	fixed	✓	✓	✓	-	✓	-	-	fixed
3DOMcity	Lab	Real	24 Mpx	varying	-	✓	-	-	✓	-	✓	fixed
ETH3D	Indoor/Outdoor	Real	0.4-24 Mpx	varying	✓	✓	✓	✓	✓	✓	-	fixed
DIODE	Indoor/Outdoor	Real	0.8 Mpx	fixed	-	-	-	-	-	✓	-	varying
BlendedMVS	Multi-scale	Blended	3 Mpx	fixed	-	✓	-	-	✓	✓	-	varying
IVL-SYNTHSFM	Objects	Synthetic	2 Mpx	varying	-	✓	-	-	✓	-	-	fixed
IVL-SYNTHSFM-v2	Objects outdoor	Synthetic	2 Mpx	varying	-	✓	-	-	✓	-	-	varying
ENRICH	Multi-scale outdoor	Synthetic	24 Mpx	varying	-	✓	-	-	✓	✓	✓	varying

terms of, e.g., acquisition condition, point of view, signal distortion) to enable the design of robust and flexible algorithms and approaches. Finally, the training set must be annotated, and this process is frequently time-consuming and resource-intensive. Synthetic heterogeneous and annotated datasets have proved to be an effective and efficient way to overcome the current limitations of real data [208, 291, 325].

Here are presented the most popular public datasets and benchmark datasets available for the photogrammetric and computer vision research communities (Figure 3.3). Table 3.2 summarizes their characteristics. For each dataset it reports the acquisition setting, its source, image properties, covered tasks, and lighting conditions.

- **Middlebury stereo**⁴ [256]. The dataset contains 32 + 24 stereo scenes (published in 2014 and 2021, respectively) to evaluate stereo algorithms on 6 and 2-megapixel images. Each scene is composed of a single stereo pair with substantial exposure variations, while scenes and cameras are static. The ground truth consists of accurate depth maps.
- **Middlebury MVS**⁵ [263]. The dataset, designed for evaluating multi-view stereo reconstruction algorithms, consists of undistorted images (640x480 pixels) of a plaster Greek temple and a dinosaur. The ground truth is a laser-scanner model (not released), while image orientations are provided.

⁴<https://vision.middlebury.edu/stereo/data/>

⁵<https://vision.middlebury.edu/mview/>



Figure 3.3: Samples from the most popular benchmark datasets reported in Table 3.2.

- **DTU Robot Image Data Sets**⁶. It contains two different collections of scenes, one designed to evaluate local features [2], the other for multi-view-stereo (MVS) investigations [136]. Images (2 megapixels) of miniatures were acquired, varying the illumination conditions. A structured light scanner was mounted on the same arm, and both the camera poses and the geometric model are provided. The trajectory of

⁶<http://roboimagedata.compute.dtu.dk/>

the images follows a circular path for all objects.

- **KITTI**⁷ [99]. This benchmark dataset is used in the context of autonomous driving. The data derive from several devices mounted on a car: two grayscale and two color cameras, a laser scanner, and an inertial navigation system (GPS/IMU). The goal was to provide training and testing images for different computer vision tasks, such as stereo matching, SLAM, 3D object detection, and depth prediction.
- **Strecha**⁸ [278]. This benchmark dataset is designed to compare the reliability of passive 3D reconstruction methods with active stereo systems. Data consists of LiDAR and camera acquisitions of outdoor scenes up to 6 megapixels. The authors have provided camera poses and calibrations and the laser-scanner model.
- **Tanks and Temples**⁹ [150]. It is a benchmark dataset for image-based 3D reconstruction algorithms. The data consists of real outdoor scenes divided into training and test sets derived from 4K videos (acquired with two rolling shutter and one global-shutter cameras). Laser scanner point clouds are provided as ground truth, as well as the reconstruction and camera pose obtained by processing the images with an “out of the box” approach based on COLMAP [259].
- **3DOMcity**¹⁰ [219]. It is a multipurpose and high-resolution (6, 016x4, 016 pixels) benchmark dataset, including 420 nadir and oblique aerial images, to assess the performance of the image-based pipeline for 3D urban reconstruction and 3D data classification. Laser scanner point clouds and reference measurements are provided to evaluate image orientation, dense image matching, and point cloud classification results.
- **ETH3D**¹¹ [262]. This dataset is composed of multi-sensors low and high-resolution images and videos for MVS investigations, with scenes

⁷<http://www.cvlibs.net/datasets/kitti/>

⁸<http://cvlab.epfl.ch/data>

⁹<https://www.tanksandtemples.org/>

¹⁰<https://3dom.fbk.eu/3domcity-benchmark>

¹¹<http://www.eth3d.net>

acquired both indoor and outdoor and laser-scanner data as ground truth. Moreover, a benchmark is available dedicated to the evaluation of SLAM algorithms.

- **DIODE**¹² [300]. It is a dataset specifically created for the depth estimation task. It contains color images with accurate laser-scanned depth measurements of indoor and outdoor scenes. It also includes validity masks for the ground truth depth scans and surfaces normal vector ground truth.
- **BlendedMVS**¹³ [325]. Similarly to ENRICH, it provides synthetic sets of images consisting of about 17,000 rendered images with a max resolution of 2,048x1,536 pixels representing 113 different scenes, both aerial and terrestrial. The peculiarity of this dataset is that the rendered images are obtained from a linear combination of low-pass and high-pass filters applied to the original and rendered images to preserve the realism of lights. This procedure implies that no new synthetic views can be generated in the dataset. BlendedMVG is a superset of BlendedMVS, expanded with additional 389 scenes for a total of about 110,000 rendered images.

The tasks covered by the presented benchmark datasets are mainly related to evaluating the performance of the entire SfM pipeline, focusing, on multi-view stereo reconstruction algorithms (Middlebury MVS, DTU, Strecha, KITTI, 3DOM city, Tanks and Temples, and ETH3D). In addition, KITTI and ETH3D propose methods for evaluating SLAM or Visual Odometry techniques. The strength of KITTI is the long image sequences for autonomous driving applications. Tanks and Temples use only video sequences for SfM reconstructions, while Strecha offers, in addition to standard datasets with calibrated cameras, also datasets with uncalibrated cameras, e.g., for reconstructions from internet photos. DIODE and BlendedMVS are designed to train algorithms for depth estimation. KITTI also allows evaluating algorithms for object detection, while 3DOMcity deals with classification

¹²<https://diode-dataset.org/>

¹³<https://github.com/YoYo000/BlendedMVS>

problems. BlendedMVS is the only dataset with accurate ground truth from synthetic models and image generation. Middlebury stereo, DTU, DIODE, and BlendedMVS present environments with very different lighting conditions within the same scene to test their effects on image orientation and final 3D reconstruction.

The methods and tools for the generation of datasets and the datasets themselves that are presented in this chapter are intended to complement existing benchmark datasets, most of which offer low-resolution images and scenes of limited size. BlendedMVS also proposes multi-scale datasets but with quite low-resolution images and only upright. The proposed datasets jointly present high-resolution and multi-scale images, camera rotations, and accurate ground truth of poses and 3D models. Accuracy is ensured by synthetic image generation, while the use of 3D reality-based surveys and 3D synthetic models guarantees texture realism.

3.3 Evaluation method for SfM 3D reconstruction

Once a 3D reconstruction has been performed using the SfM and MVS pipelines, it is possible to evaluate the quality of the results obtained by comparing them to a ground truth with the same data representation. An evaluation method applicable to the reconstructions obtained from real and synthetic datasets is here defined. This method requires the ground truth geometry of the model to be reconstructed and the ground truth camera pose for each image. The proposed evaluation method is composed of four phases: (i) alignment and registration, (ii) evaluation of sparse point cloud, (iii) evaluation of camera pose, (iv) evaluation of dense point cloud.

Another approach to the evaluation of the SfM reconstructions is the one presented by Tefera et al. [284]. The authors designed a Web application that can visualize reconstruction statistics, such as minimum, maximum and average intersection angles, point redundancy, and density. All mentioned

statistics do not require a ground truth.

3.3.1 Alignment and registration

Since the reconstruction and the ground truth use different reference coordinate systems (RCSs), it is necessary to find the correct alignment between the two. The translation, rotation, and scale factors to align the two RCS can be defined using a rigid transformation matrix T . The adopted procedure finds this matrix by aligning the reconstructed sparse point cloud to the ground truth geometry using a two-step process. A first phase of coarse alignment and a second phase of fine registration allow to overlap in the best possible way the reconstruction to the ground truth. Alignment and registration steps generate two transformation matrices T_1 and T_2 of size 4×4 in homogeneous coordinates. By multiplying the matrices to each other in the order in which they were identified, it is possible to obtain the global alignment matrix $T = T_2 \cdot T_1$. This matrix is applied to the reconstructed clouds (sparse and dense) and also to the estimated camera poses to obtain the reconstruction aligned and registered with the ground truth. The ground truth can present itself as a dense points cloud or a mesh. Alignment algorithms work only with point clouds, so in the case where the ground truth is a mesh, a cloud of sampled points is used to bring the problem back to the alignment of two point clouds.

Alignment: To increase the probability of success of the *Fine registration* step (Section 3.3.1) and to reduce the processing time, it is necessary to find a good alignment of the reconstructed point cloud with the ground truth. This operation can be performed manually by defining the parameters of rotation, translation, and scale or, more conveniently, by specifying pairs of corresponding points to be aligned by a specific algorithm (i.e. Horn [126]). This algorithm uses three or more points of correspondence between the reconstructed cloud and the ground truth to estimate the transformation necessary to align the specified matching points. The method proposed by Horn estimates the translation vector by defining and aligning the barycenters

of the two point clouds. The scaling factors are defined by looking for the scale transformation that minimizes the positioning error between the specified matching points. Finally, the rotation that allows the best alignment is estimated using unit quaternions from which the rotation matrix can be extracted. The algorithm then returns the transformation matrix T_1 which is the composition of translation, rotation, and scaling.

Fine registration: After aligning the reconstruction to the ground truth, it is possible to refine such alignment using a process of fine registration. The algorithm used for this phase is Iterative Closest Point (ICP) [27, 52]. It takes the two point clouds and a criterion for stopping the iterations as input, then it produces a rigid transformation matrix T_2 that allows better alignment.

The stopping criterion is usually a threshold to be reached in the decrease of the RMSE measure. For very large point clouds it is also useful to also limit the number of iterations allowed to the algorithm. Furthermore, ICP does not work well if the point cloud to be registered and the reference cloud are very different, for example when one cloud includes portions that are not present in the other. In this case, it is first necessary to clean the clouds so that both represent the same portion of a scene or object.

3.3.2 Evaluation of sparse point cloud

The sparse point cloud generated by SfM can be evaluated in comparison to the ground truth of the reconstruction. The evaluation considers the distance between the reconstructed points and the geometry of the ground truth. Once the reconstruction is aligned with the ground truth, it is possible to proceed with the evaluation of the reconstructed point cloud, calculating the distance between the reconstructed points and the ground truth.

If the ground truth is available as a dense point cloud, the distance can be evaluated by calculating the Euclidean distance. For each 3D point of the cloud to be compared, the nearest point is searched in the reference cloud calculating the Euclidean distance. Octree [188] data structures can be used

to partition the three-dimensional space and speed up the calculation. Once the distance values are obtained for all points in the cloud, the mean value and standard deviation are calculated.

If the ground truth is available as a mesh, the distance is calculated between a reconstructed point and the nearest point on the triangles of the mesh. This can be done using the algorithm defined by David Eberly [71]. Given a point of the reconstructed point cloud, for each triangle of the mesh the algorithm searches the point with the smallest square distance. Among all the selected points (one for each triangle) the one with the smallest square distance is chosen and the square root of this value is returned. This calculation is repeated for each point of the reconstructed cloud. Even in this case octree data structures can be used to partition the three-dimensional space and speed up the computation. Once distance values are obtained for all points in the cloud, the mean value and standard deviation are computed.

In both cases, the reconstructed cloud must contain only points relative to objects included in the ground truth model used for comparison. Usually, the ground truth includes only the main object of the reconstruction, ignoring the other elements visible in the dataset's images. If the reconstruction includes parts of the scene that do not belong to the ground truth, the distance calculation will be distorted. To overcome this problem, it is possible to cut out the cloud of points of the reconstruction, manually eliminating the parts in excess before evaluating the distance. If this is not possible (mainly because the separation between the objects of interest and those not relevant is not simply identifiable), then the same result can be achieved by specifying a maximum distance allowed for the evaluation of the reconstruction. If a reconstruction point is evaluated with a greater distance from the ground truth than allowed, it is discarded so that it does not affect the overall assessment.

3.3.3 Evaluation of camera pose

In addition to the sparse points cloud, the SfM pipeline also generates information about the camera poses. The pose of each camera can be compared to the corresponding ground truth. In particular, the method defined here provides information on the distance between the positions and the difference in orientation between each pair of ground truth and estimated camera pose. Ideally, if a camera is reconstructed in the same position as its ground truth, it observes the same points. Consequently, its orientation is the same as that of the ground truth. However, in real cases, it is possible to observe slight differences between the orientations, and for this reason, an evaluation is provided.

Position evaluation: The position of a reconstructed camera is evaluated by calculating the Euclidean distance between the reconstructed position and the corresponding ground truth camera position. Such values can also be used to calculate average distance and standard deviation.

Orientation evaluation: The differences in orientation of the cameras are evaluated using the angle of the rotation necessary for the relative transformation that, applied to the reconstructed camera, brings it to the same orientation as the corresponding ground truth camera. The camera orientation can be defined using a unit quaternion. Therefore, it is possible to define \mathbf{q}_{GT} as the camera ground truth orientation and \mathbf{q}_E as the reconstructed camera orientation. The relative transformation that aligns the reconstructed camera at the same orientation of the ground truth is defined by the quaternion \mathbf{q}_R with components w, x, y, z that is calculated as follows:

$$\mathbf{q}_R = \mathbf{q}_E^{-1} \cdot \mathbf{q}_{GT} \quad (3.1)$$

where \mathbf{q}_E^{-1} is the inverse quaternion of \mathbf{q}_E calculated by Equation 3.2 where \mathbf{q}_E^* is the conjugate of \mathbf{q}_E and $\|\mathbf{q}_E\|$ is the norm.

$$\mathbf{q}_E^{-1} = \frac{\mathbf{q}_E^*}{\|\mathbf{q}_E\|^2} \quad (3.2)$$

By substituting in Equation 3.1 the term \mathbf{q}_E^{-1} with his definition, the equation becomes:

$$\mathbf{q}_R = \frac{\mathbf{q}_E^*}{\|\mathbf{q}_E\|^2} \cdot \mathbf{q}_{GT} \quad (3.3)$$

Being rotations expressed with unit quaternions, the norm of \mathbf{q}_E is always 1 accordingly the equation can be simplified obtaining:

$$\mathbf{q}_R = \mathbf{q}_E^* \cdot \mathbf{q}_{GT} \quad (3.4)$$

Quaternion \mathbf{q}_R represents the rotation transformation necessary to change the orientation of the reconstructed camera so that it is the same as the ground truth. This can be expressed by defining a rotation axis and the angle necessary to rotate the camera around it. This rotation angle can be used as a quality measure of the reconstructed camera rotation. If the orientation of the reconstructed camera is the same as the ground truth camera, the rotation angle of the defined transformation is 0; when the orientation of the reconstructed camera is different from that of the ground truth, the value of the rotation angle necessary to align the orientation of the camera also increases.

The representation of \mathbf{q}_R in terms of axes \mathbf{a} (vector with components x, y, z) and rotation angle α is defined as follows:

$$\mathbf{q}_R = \cos\left(\frac{\alpha}{2}\right) + i \mathbf{a}_x \sin\left(\frac{\alpha}{2}\right) + j \mathbf{a}_y \sin\left(\frac{\alpha}{2}\right) + k \mathbf{a}_z \sin\left(\frac{\alpha}{2}\right) \quad (3.5)$$

Angle α is expressed in radians and the rotation axis can be extracted from the quaternion using Equations 3.6 and 3.7. The identified angle is always positive.

$$\alpha = 2 \cdot \arccos(\mathbf{q}_{Rw}) \quad (3.6)$$

$$\mathbf{a}_x = \frac{\mathbf{q}_{Rx}}{\sqrt{1 - \mathbf{q}_{Rw}^2}} \quad \mathbf{a}_y = \frac{\mathbf{q}_{Ry}}{\sqrt{1 - \mathbf{q}_{Rw}^2}} \quad \mathbf{a}_z = \frac{\mathbf{q}_{Rz}}{\sqrt{1 - \mathbf{q}_{Rw}^2}} \quad (3.7)$$

Using this representation particular attention should be paid when the rotation angle is 0° . When this happens the rotation axis is arbitrary, and the result is the same whichever is chosen. The quaternion is in the form $\mathbf{q} = 1+i0+j0+k0$,

and consequently division by 0 must be avoided when applying Equation 3.7. To solve the problem an arbitrary axis with unitary norm can be chosen; in this way, there is no need to compute a rotation axis, and the length is still unitary. Angle α from Equation 3.6 can be converted from radians to α_{deg} expressed in degrees. This angle can vary from 0° to 360° ; it also must be taken into account that α_{deg} is a rotation around the axis of direction \mathbf{a} or a rotation of $-\alpha_{deg}$ around the opposite direction axis. Moreover, a rotation greater than 180° around the \mathbf{a} axis can also be expressed as a rotation of $-(360 - \alpha_{deg})$ degrees around the same axis. To correctly compute the difference of orientations the smallest angle must be considered, independently of its direction; therefore in the $\alpha_{deg} > 180$ case the difference between the camera's orientations is computed as $360 - \alpha_{deg}$. The differences in orientations measured through angle α can also be used to calculate the average distance value and the standard deviation.

3.3.4 Evaluation of dense point cloud

The Multi-View Stereo (MVS) pipeline reconstructs the dense points cloud of the scene observed by the set of images. This cloud of points can be evaluated in comparison to the ground truth of the object. The evaluation takes place in terms of the distance between the reconstructed points and the geometry of the ground truth. Once the dense reconstruction is registered in the best possible way with the ground truth, it is possible to proceed with the evaluation of the reconstructed cloud by calculating the distance between the reconstructed points and the ground truth. This evaluation can be done in the same way used for the sparse point cloud, as illustrated in Section 3.3.2.

3.4 SfM Flow: synthetic data generation [182]

In the literature, evaluation of 3D reconstruction pipelines is often performed on data specifically acquired to test the designed pipeline and algorithms [275]. The data is not always made available to the research community making it difficult to perform comparisons. Although available datasets exist (see Section 3.2.2), they may lack some characteristics that are needed to highlight the peculiarities of different 3D reconstruction solutions. The need for ad-hoc real scenes as reference models for the evaluation process is restrictive and not always practical.

For these reasons *SfM Flow* is proposed, an add-on for Blender [35] that allows rapid and efficient evaluation and comparison of different 3D reconstruction pipelines. *SfM Flow* allows the generation of datasets to stress the pipelines under different conditions of lighting, multiple camera effects, and scene complexity [32, 181, 183]. The data are generated from synthetic scenes that provide a known geometry and allow an easy, and precise, evaluation of the reconstruction.

To the best of our knowledge, SyB3R [158] is the only similar tool supporting synthetic dataset creation for 3D reconstruction. Recently, a few rendering engines started providing API to interact with them [234, 299], mainly for machine learning tasks, and none of them is specifically designed for 3D reconstruction. Existing synthetic datasets are usually generated using rendering engines and application-specific automation scripts that lack flexibility. To exploit the features of *SfM Flow*, it is usually necessary to use multiple software to manually build and render a synthetic dataset, run a 3D reconstruction pipeline and perform a reconstruction evaluation. *SfM Flow* supports the complete workflow in a single package.

SfM Flow is an add-on for Blender that focuses on providing an easy-to-use toolkit for the evaluation of 3D reconstructions performed starting from images. This add-on covers both the steps required for image generation and reconstruction evaluation. To facilitate the use of the tool, each phase of the evaluation is available in the user interface as a separate tool. The images are rendered from a custom virtual 3D scene, thus allowing the

simulation of a variety of acquisition setups without requiring the use of dedicated hardware for the acquisition of real scenes. The *SfM Flow* add-on is publicly available at https://github.com/davidemarelli/sfm_flow, a complete wiki is also accessible at https://github.com/davidemarelli/sfm_flow/wiki. Details about SfM Flow’s software architecture and an example of step-by-step usage are available in Appendix A and B.

3.4.1 Software functionalities

The main functionalities of *SfM Flow* can be grouped by scope as scene setup, data generation, 3D reconstruction pipeline execution, and 3D reconstruction evaluation. *SfM Flow* provides tools for the complete workflow. To offer higher flexibility, each group of features also works standalone. This section illustrates the functionalities in the order in which they are used for a complete workflow.

Scene initialization The first set of tools concerns the 3D scene initialization or rather the operations that support the user during the definition of the 3D scene that will be used for the subsequent steps of data generation and reconstruction evaluation. In this phase, a virtual 3D scene is created by defining the portrayed objects, the environment setting, the lighting setup, as well as the image acquisition viewpoints. After choosing the main subject of the scene, the environment can be set up in different ways by selecting a scene type and a lighting condition through the “Initialize current scene” functionality. The available scene types are *floor* and *hemisphere*. The first one adds a concrete-looking floor to the scene. The latter includes the scene in a smooth hemisphere with the objects placed on its floor. The lighting options available are *sky & sun* and *point lights*. The first one creates a procedural sky and places a sunlamp in the scene to simulate an outdoor setup. The *point lights* option places four point-lights around the scene to provide uniform illumination. It is also possible to skip one or both the configuration of scene type and light, and instead use an existing or a custom setup. Finally, if the

initialize camera flag is enabled, the intrinsic camera calibration parameters are set to the default values. In any case, the rendering engine is switched to Cycles, and some render default values are applied. Objects added to the scene by this toolset are placed in a new collection named “SfM_Environment”.

After defining the scene, Ground Control Points (GCPs) can be manually added to it. *SfM Flow* provides several commonly used patterns for GCPs. Sample GCPs are visible in Figure 3.4. The placement of GCPs is guided by the add-on, which automatically rotates them in alignment with the closest surface.

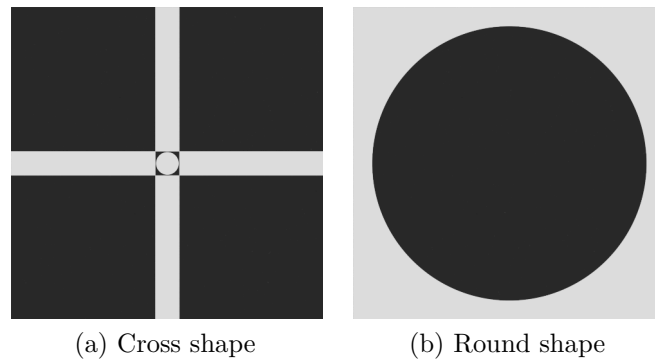


Figure 3.4: Example of Ground Control Point patterns.

Multiple cameras can be used to acquire images of the same scene. *SfM Flow* allows the provision of different camera characteristics (resolution, focal length, etc.) and different acquisition paths. A camera can be manually defined by the user or selected from a list of commercial cameras whose parameters are set by the add-on. The cameras can also be configured to perform single-camera acquisitions or multi-camera acquisitions, thus simulating configurations required for aerial imaging. The add-on also provides aid to ensure the desired Ground Sample Distance (GSD) when configuring a camera.

The “Animate camera” functionality simplifies the camera animation process by providing tools for the camera path setup to allow image acquisition portraying the scene from different viewpoints. The type of camera animation and its duration can be customized. It is also possible to use random camera

positions around the computed points to simulate a non-perfect acquisition setup. Multiple animations can be chained to obtain a complex acquisition setup. Predefined animation types include: aerial, helix, hemisphere, circle, and multiple circles moving the target viewpoint upwards. *SfM Flow* also provides a way to visualize all the acquisition positions of the camera to show at a glance the animation of the camera in the 3D viewport.

Similarly, if the scene is initialized with the *Sky & Sun* lighting setup, it is possible to animate the sun constraining it to follow a semi-circular path around the scene. The sun animation can be randomized to simulate acquisition at different hours. Thanks to the flexibility of Blender, the outcome of each of these operations can be further manually manipulated to create a custom setup to fit specific needs.

Data generation The second group of tools allows the generation of synthetic images and additional data necessary for the execution and evaluation of the 3D reconstruction. A custom rendering tool generates the synthetic images and camera ground truth information. Available image output formats are JPEG, PNG, BMP, and AVI. It is also possible to apply camera effects such as motion blur and depth of field (DoF) during image rendering. The motion blur effect can be applied to random frames using user-specified probability and shutter time. For a realistic DoF simulation, the camera focus distance is automatically set to the first object intersection along the camera’s look-at direction during the camera animation setup. EXIF metadata are set for each image saved as JPEG or PNG file. In addition to RGB images, the *SfM Flow* add-on also produces depth maps containing metric depth for each pixel in the camera’s field of view. This depth information is saved as an OpenEXR file. A colored preview of the depth information is also provided as PNG files.

The rendering process also generates four Comma-Separated Values (CSV). A file named “scene.csv” describes the acquisition setup using the following fields:

1. **scene_name**: string, name of the acquisition setup, same as the 3D model name

2. `images_count`: integer, number of images in each set, always 100
3. `unit_system`: string, measurement unit system, always METRIC
4. `unit_length`: string, length unit, always METERS
5. `scene_center_x,y,z`: three floats, coordinate of the scene's center
6. `scene_ground_center_x,y,z`: three floats, coordinate of the scene's ground center
7. `scene_width,depth,height`: three floats, size of the scene along X, Y and Z axes
8. `mean_cam_dist_center`: float, mean camera distance from the center of the scene
9. `mean_cam_dist_obj`: float, mean camera distance from the object's surface, computed as the distance between the camera and the first point of intersection with the object along the camera's look-at direction
10. `mean_cam_height`: float, mean camera height from scene's ground

The ("cameras.csv") file contains information about camera position, camera rotation, camera look-at direction, depth of field, motion blur, and sun lighting position (if any) for each image.

1. `label`: string, the filename of the image the entry refers to, including the extension.
2. `position_x,y,z`: three float numbers representing the global position of the camera.
3. `omega, phi, kappa`: Omega, Phi, Kappa angles defining the rotation of the camera. Three floats, representing angles in radians.
4. `yaw, pitch, roll`: rotation of the camera using Yaw, Pitch, Roll angles. Three floats, representing angles in radians.

5. `rotation_w,x,y,z`: four floats, representing the camera rotation quaternion.
6. `lookat_x,y,z`: three floats, representing the camera look-at direction vector.
7. `depth_of_field`: boolean, 'True' if image rendered with depth of field enabled, 'False' otherwise
8. `motion_blur`: boolean, 'True' if image rendered with motion blur enabled, 'False' otherwise
9. `sun_azimuth`: float, azimuth angle in radians of the sunlamp illuminating the scene
10. `sun_inclination`: float, inclination angle in radians of the sunlamp illuminating the scene

The values are defined according to a global coordinate reference system, having X growing right/east, Y growing forward/north, and Z growing upward/zenith. Omega, Phi, and Kappa are counterclockwise (CCW) local rotations along the X, Y, and Z axis, applied in the following order $R = R_x \cdot R_y \cdot R_z$. Pitch and Roll are CCW local rotations along the X and Y-axis respectively; Yaw is a clockwise (CW) local rotation along the Z-axis. The order of application is $R = R_z \cdot R_x \cdot R_y$. In any case, the parameters define the world-space rotation and position transformations defining the mapping of the camera space to the global coordinate reference system. Thus, a camera with a 0 translation and a 0° rotation along all axes is placed at the origin and aligned to the global coordinate system (looking along the -Z axis with its up direction aligned with +Y and right direction aligned to +X).

The third file “`gcp_list.csv`” describes the 3D location of the GCPs used in the scene. The reported fields are:

1. `gcp_name`: string, unique identifier of the GCP.
2. `x_east`, `y_north`, `z_altitude`: three floats, representing the global coordinates of the center of the GCP in the scene.

3. **type**: string, defines the shape of the GCP either *cross* or *round*.

The last file “gcp_images_list.csv” contains the GCPs visibility information for each rendered image. The reported fields are:

1. **image_name**: string, the filename of the image that portrays the GCP.
2. **gcp_name**: string, unique identifier of the GCP.
3. **image_x,y**: two floats, X and Y coordinate of the center of the GCP in the image (0,0 is the top left corner of the top left pixel in the image, X grows right and Y grows downwards).

It is also possible to export the geometry ground truth as a Wavefront OBJ file. The exported file excludes by default the environment objects that are part of the “SfM_Environment” collection. It is also possible to export only the current object selection as geometry ground truth.

Pipeline execution 3D reconstruction pipelines can be run directly from the user interface. *SfM Flow* provides direct support for popular incremental SfM pipelines such as COLMAP [259], OpenMVG [197], Theia [282], and VisualSFM [316]; commands for other custom pipelines are configurable in the user preferences. Before the execution, the user can specify a reconstruction workspace in which the add-on will create a separate folder for each pipeline where to save the reconstruction and the execution log.

Reconstruction evaluation The reconstruction evaluation is carried out following the procedure defined in Section 3.3. After importing a reconstruction through *SfM Flow*, it is possible to proceed in the evaluation process by registering the reconstructed point cloud to the scene’s geometry. It can be done using either the integrated Iterative Closest Point (ICP) [27] implementation or a registration matrix provided by an external tool. The ICP-based algorithm uses by default only 25% of the point cloud for the alignment process and allows a maximum number of iterations defined as 1% of the size of the point cloud.

A point cloud filtering functionality takes into account the presence in the reconstruction of points that are not part of the ground truth. This tool can filter the point cloud before the ICP registration, discarding points that are more distant than a given threshold from the ground truth. Once the point cloud has eventually been filtered, the alignment tool performs ICP registration of the remaining points.

Finally, the reconstruction evaluation tool generates an evaluation report that contains information about the reconstructed point cloud and the camera poses evaluation. The first part is an evaluation of the geometry in terms of the Euclidean distance between the reconstructed 3D points and a dense point cloud sampled over the ground truth of the virtual 3D model(s). This evaluation can be performed either on the filtered point cloud or the full one. Currently, *SfM Flow* provides only the commonly used cloud-to-cloud euclidean distance metric. However, other metrics (cloud accuracy, completeness, point density, and roughness) may be more suitable depending on the use case. The reconstruction evaluation module is designed to be easily extendable to fit specific needs by implementing other metrics.

The second part evaluates camera poses in terms of camera position distance and orientation differences. The position is evaluated using the Euclidean distance between the reconstructed camera position and the ground truth one. The orientation difference is computed using the angle of the rotation transformation needed to align the reconstructed camera to its corresponding ground truth, forcing them to the same position. Finally, the camera orientation is evaluated as the non-oriented angle between look-at vectors of each pair of reconstructed and ground truth camera pose. The camera pose evaluation also reports the percentage of images used by the reconstruction pipeline. All evaluations report the minimum, maximum, mean, and standard deviation values of each metric considered.

Command line execution The rendering process of complex scenes is resource-intensive and thus often performed on servers. With this scenario in mind, *SfM Flow* comes with additional command line parameters to enable execution from scripts. Such parameters allow rendering with all the possible

combinations of effects described in Section 3.4.1 and post-rendering export of addition ground truth files.

3.4.2 Impact

SfM Flow provides a fast way to test and stress the 3D reconstruction pipelines using synthetic images, and allow the simulation of a variety of scene setups. By using synthetic data it is possible to overcome the limitations imposed by the need for real 3D model acquisition, which is tricky, requires dedicated hardware, and complex setup making it not suitable in many situations. *SfM Flow* is intended for use by researchers and final users for the evaluation of the existing 3D reconstruction pipelines as well as to provide support for the development of new 3D reconstruction methods. The availability of *SfM Flow* as an add-on for Blender integrates its functionalities with a well-known and widely adopted 3D modeling software, making it easily usable by experts in the 3D reconstruction field as well as by newcomers. Finally, *SfM Flow* is written in Python, a popular programming language that encourages further development including the addition of new functionalities and its adaptation to specific needs. Version 1.0.0 of *SfM Flow* was used to create the IVL-SYNTHSFM-v2 [181] dataset; its preliminary version was previously used to develop the IVL-SYNTHSFM dataset [32]. Version 1.1.0, which is still under development, was used to create the ENRICH dataset [183].

3.5 Proposed synthetic datasets for 3D reconstruction evaluation

As stated in previous sections, the evaluation of 3D reconstruction pipelines requires some datasets of source images and associated ground truth. Over the years, various datasets of real-world objects have been created (see Section

3.2). Those usually contain the ground truth of the object as a dense point cloud, acquired through high-accuracy laser scanners. In some cases, the ground truth is made available as a three-dimensional mesh generated starting from a scanner acquisition or a high-quality reconstruction obtained directly from the images that compose the dataset. In any case, the accuracy of the ground truth depends on the quality of the instrumentation used and the process used to acquire it. The assumption that must be made to use the ground truth so generated is that it is more precise than the reconstruction generated by the pipelines. Otherwise, having a low-quality ground truth, it would not be possible to evaluate the accuracy of the reconstructed model. Usually, these datasets do not report the ground truth of the camera poses and thus do not allow for evaluation of the pose of reconstructed cameras. The generation of such datasets encounters limitations due to the equipment or the scene to be captured itself, making it complex to generate a set of images that fully comply with the guidelines. Moreover, it is difficult to find available datasets that include model ground truth, and even when it is available its quality is low, and occluded surfaces are missing.

To overcome the problems in creating real datasets is possible to use virtual 3D models to generate synthetic datasets with good image quality, intrinsic parameters for each image, and optimal 3D model ground truth. Concerning real datasets usually acquired with physical imaging devices, synthetic datasets make it possible to have accurate and infinitely precise ground truths. We can generate synthetic datasets by capturing images of virtual 3D models employing rendering software. For such purposes, Blender [35] and the *SfM Flow* add-on (see Section 3.4) are employed. Firstly the subject of the dataset needs to be chosen; for optimal results, the 3D model must have a highly detailed geometry and texture. The model must then be placed in a scene where lights and other objects can be included. A camera is then added, and all its intrinsic calibration parameters must be set. Such a camera is then animated to observe the scene from different viewpoints; each frame of the animation will be used as an image of the dataset. Once everything is set, the images can be rendered, and thanks to *SfM Flow* additional metadata and ground truth information can be exported

from the 3D virtual scene setup. This approach was followed to propose three new synthetic datasets. In detail,

1. **IVL-SYNTHSFM** [32] – contains images and ground truth information of five single-object 3D scenes. It aims to support the evaluation of 3D reconstruction methods when reconstructing a single object in an isolated environment without any other reference in the scene.
2. **IVL-SYNTHSFM-v2** [181] – is an extension of the previous dataset that also provides a basic environment around the objects. It aims to stress the 3D reconstruction pipelines introducing strong light variations between the acquisitions, depth of field, and motion blur artifacts.
3. **ENRICH** [183] – contains three sub-datasets aiming to support the evaluation of 3D reconstruction methods at different scales. It includes an aerial view of a city, a ground view of a square, and a ground view of a statue. Multiple cameras and lighting conditions are included.

Table 3.3: Comparison of the proposed datasets.

Name	Setting	Real/Synthetic	Resolution	Image format	SfM	MVS	Depth	GCPs	Lighting	Camera effects
IVL-SYNTHSFM	Objects	Synthetic	2 Mpx	varying	✓	✓	–	–	constant	–
IVL-SYNTHSFM-v2	Objects outdoor	Synthetic	2 Mpx	varying	✓	✓	–	–	varying	✓
ENRICH	Multi-scale outdoor	Synthetic	24 Mpx	varying	✓	✓	✓	✓	varying	–

The three datasets aim to enable the evaluation of different aspects of 3D geometry reconstruction. A quick comparison of the characteristics of the datasets is provided in Table 3.3.

The IVL-SYNTHSFM dataset provides images of a single object acquired with a white background to simulate a controlled environment. Different objects are provided to evaluate the overall 3D reconstruction pipeline performances w.r.t. different object sizes, geometrical complexity, and texture. The IVL-SYNTHSFM-v2 is an extension of the previous dataset, which introduces camera effects (i.e. motion blur and depth of field), variable lighting conditions, and a non-uniform outdoor-like background. This dataset relaxes the controlled environment assumption of its previous version. Both sets allow evaluation of 3D reconstruction pipelines in the context of MR catalogs for the

acquisition of 3D models of real objects to be virtually shown in a catalog (e.g., AR product configurators, MR virtual tours, MR publishing features, Virtual try-on, see Section 2.2). The characteristics of the datasets allow to evaluate the impact of different acquisition configurations and pipeline parameters on the final result. The ENRICH dataset aims instead to enable evaluation of the impact of changes in single steps of the pipeline, as well as different acquisition setups at various scales. It is best suited for the evaluation of 3D reconstruction methods for the creation of 3D virtual environments. This is again relevant in the creation of 3D virtual environments for MR catalogs (e.g., MR tourist guides, Augmented and Virtual shops, see Section 2.2). All of them provide pixel-precise ground truth and additional information.

3.5.1 The IVL-SYNTHSFM dataset [32]

The goal of this dataset is to make it possible to evaluate the ability to reconstruct a single 3D object when no other elements are included in the scene. The dataset has been made publicly available for download at <http://www.ivl.disco.unimib.it/activities/evaluating-the-performance-of-structure-from-motion-pipelines/>. The images in this dataset are generated from the five 3D models shown in Figure 3.5.

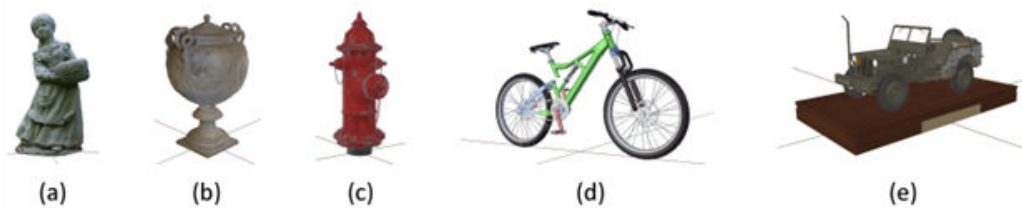


Figure 3.5: 3D models used for synthetic data generation: (a) Statue. (b) Empire Vase. (c) Hydrant. (d) Bicycle. (e) Jeep.

Five synthetic datasets of different 3D models (Figure 3.5) have been generated using the tool described in Section 3.4:

3.5 Proposed synthetic datasets for 3D reconstruction evaluation

- Statue¹⁴ — set of images about a statue of height 10.01 m, composed of 121 images;
- Empire Vase¹⁵ — set of images about an ancient vase of height 0.92 m, composed of 86 images;
- Hydrant¹⁶ — set of images about an hydrant of height 1.00 m, composed of 66 images;
- Bicycle¹⁷ — set of images about a bicycle of height 2.66 m, composed of 86 images;
- Jeep¹⁸ — set of images about a jeep of height 2.48 m, composed of 141 images.

The selected 3D models (Figure 3.5) were chosen based on the different levels of geometry complexity and texture detail which translates into different levels of complexity for the reconstruction process. The Statue model is composed of 60k vertices, 295k for the Vase, 9k for the Hydrant, 300k for the Bicycle, and 2335k for the Jeep model.

All the images of each dataset have been acquired at resolution 1920x1080px using a virtual camera with a 35mm focal length and 32x18mm sensor. To achieve this, each model is placed in a reference scene and is rendered using a virtual camera moving in a circle around the object. The reference scene does not include any other object and provides a white uniform background for the main object. Lighting is provided by four point-lights to generate shadows on the object itself. An example of the sequence of operations needed to set up the virtual scene is visible in Figure 3.6. This dataset has been created using a preliminary version of the *SfM Flow* add-on which allowed basic scene setup operations and ground truth export from the virtual scene.

¹⁴<https://free3d.com/3d-model/statue-92429.html>

¹⁵<https://www.blendswap.com/blends/view/90518>

¹⁶<https://www.blendswap.com/blends/view/87541>

¹⁷<https://www.blendswap.com/blends/view/67563>

¹⁸<https://www.blendswap.com/blends/view/82687>

3.5 Proposed synthetic datasets for 3D reconstruction evaluation

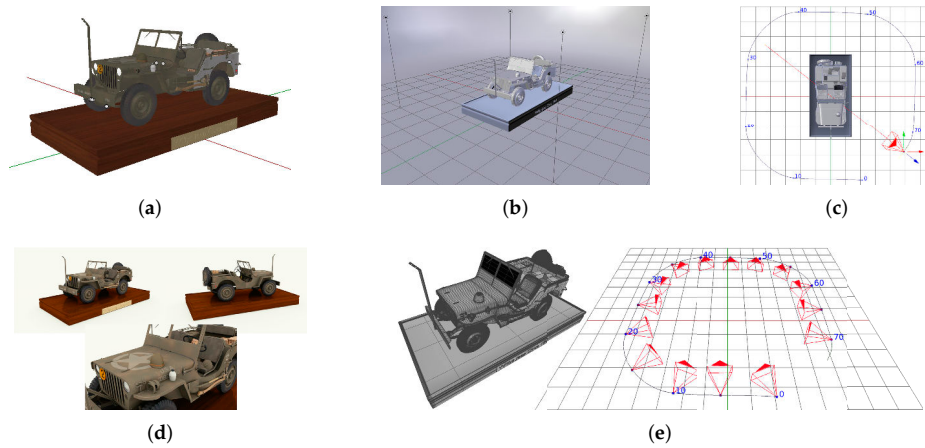


Figure 3.6: Example of synthetic dataset generation steps. (a) 3D model. (b) Scene setup. (c) Camera motion around the object. (d) Images rendering. (e) 3D model geometry and camera pose ground truth export.

The data can be used to evaluate and compare 3D reconstructions of single objects from multiple images obtained using various techniques. The data is of interest to researchers who would like to test and compare various 3D reconstruction methods to check the results of different approaches to the reconstruction of single objects. It can be used to assess the performance of state-of-the-art methods as well as evaluate and compare new techniques. The data can be used to determine how a 3D reconstruction method reacts when used on images of objects with differences in size, geometry, and texture details. The data contains information about camera intrinsic and extrinsic calibration parameters that allow precise camera positioning, reconstruction estimation, and evaluation. It is highly relevant for the evaluation of reconstructions made by techniques that assume unknown camera poses (e.g. Structure from Motion) and reconstructions pipelines that require known camera poses, such as Multi-View Stereo (MVS).

3.5.2 The IVL-SYNTHSFM-v2 dataset [181]

The IVL-SYNTHSFM-v2 dataset is an extension of the previous IVL-SYNTHSFM dataset. While maintaining the same goal of the IVL-SYNTHSFM dataset, it also tries to test the robustness of the reconstruction pipelines on different image acquisition setups. It also aims to allow evaluation of the impact of variations in illumination conditions, depth of field, and motion blur on the reconstruction pipelines. The dataset is publicly available for download at <https://doi.org/10.17632/fnxy8z8894>. The images in this dataset are generated from the same five 3D models used for IVL-SYNTHSFM (Figure 3.5). Each model is placed in a reference scene and is rendered under different lighting and camera conditions. Eight scenes are created for each object, and images are acquired from different viewpoints. Each scene is composed of a set of 100 captured images. The list of the image sets, along with the acquisition setup, is shown in Table 3.4.

All the images rendered for each 3D scene are available as JPG files of resolution 1920x1080 pixels. The images were acquired using a perspective virtual camera with a 35mm focal length and 18x32mm sensor; this information can also be found in the EXIF metadata (version 2.3) of each image.

The dataset was created using Blender as 3D modeling and rendering software with the aid of the *SfM Flow* add-on. For each scene portraying a single object, a 3D model is placed in the center of the scene leaning on a plane, and a sunlamp lights up the environment. The floor is textured with a

Table 3.4: List of available images sets for each object in the dataset.

Set name	Lighting setup	Depth of field	Motion blur
Fs	Sun, fixed position	No	No
fs-dof	Sun, fixed position	Yes, on all images	No
fs-mb	Sun, fixed position	No	Yes, on random images
fs-dof-mb	Sun, fixed position	Yes, on all images	Yes, on random images
Ms	Sun, random position	No	No
ms-dof	Sun, random position	Yes, on all images	No
ms-mb	Sun, random position	No	Yes, on random images
ms-dof-mb	Sun, random position	Yes, on all images	Yes, on random images

3.5 Proposed synthetic datasets for 3D reconstruction evaluation

concrete-looking material, and a sky with procedural clouds is created. The scene is then observed from different viewpoints by a moving perspective camera. The camera moves in a circle around the vertical axis at the scene center to obtain complete coverage of the object. Depending on the complexity and size of the object, the movement can be a single circle or two circles at different heights. To simulate realistic manual acquisition the camera position is randomized by 5% of the acquisition points sampled on the movement circle. A sample scene setup is visible in Figure 3.7.

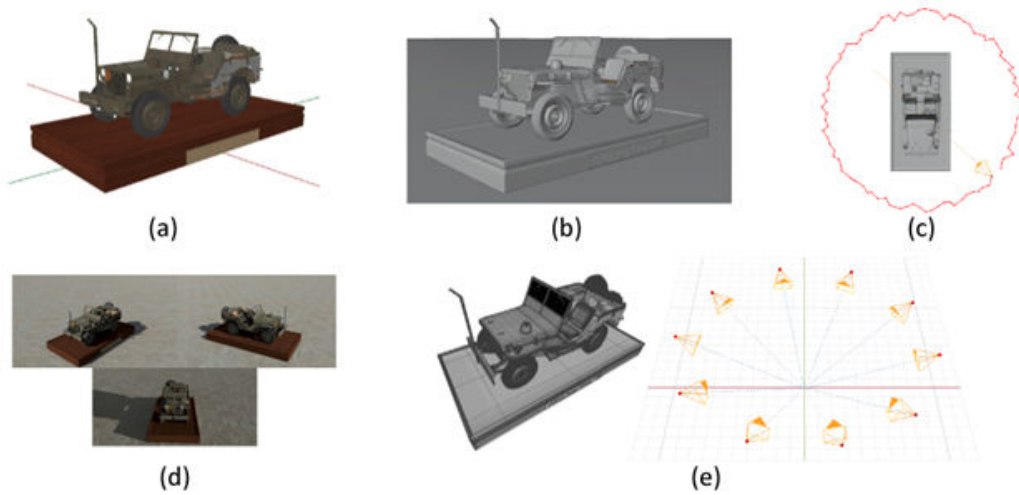


Figure 3.7: Example of data generation steps for the Jeep model: (a) 3D model. (b) Scene setup with main object, floor surface and lights. (c) Camera animation around the object. (d) Images rendering. (e) 3D model geometry and camera pose ground truth export.

For each of these scenes portraying different objects, eight sets of images are acquired under different lighting, depth of field, and motion blur. For the sets that make use of a moving sun, the sunlamp is placed at a random position for each image; this is intended to simulate the acquisition during different hours of the day. The sun’s position is randomized along a semicircular path and kept consistent across different sets of the same target object. The depth of field is applied to all images of the images sets that make use of it. Finally, in the sets that make use of motion blur the effect is introduced randomly on approximately 33% of the images. The different setups for each object can be

used to evaluate the performances of reconstruction pipelines under different light conditions and the robustness to the depth of field and motion blur. Rendering of images uses Cycles, the Blender’s path-tracing render engine, that simulates physics-based light interactions and allows the generation of photo-realistic images. Samples of rendered images are visible in Figure 3.8.



Figure 3.8: Sample of rendered images of the Jeep model. (a) image from the ‘fs’ set. (b) image from the ‘fs-dof-mb’ set. (c, d) images from the ‘ms’ set.

3.5.3 The ENRICH dataset [183]

ENRICH is a new multi-purpose synthetic benchmark dataset created to (i) complement existing close-range and aerial datasets, (ii) boost investigations and analyses in the 3D reconstruction process, and (iii) evaluate photogrammetric and computer vision algorithms. To this end, the dataset comprises three collections of outdoor images capturing an urban area taken from above, a city square, and a statue. Compared to existing benchmark datasets, ENRICH offers higher-resolution images that are rendered with different lighting conditions (clear blue sky, cloudy sky, sunrise, etc.) and camera orientation (landscape and portrait). ENRICH can be exploited to benchmark algorithms under variable and realistic conditions with data acquired at diverse scales, different cameras, and lighting setups. Unlike many benchmarks and challenges, such as the Image Matching Challenge [139], images with rotations from 0 to 180 degrees are included in the dataset to encourage the scientific community to propose solutions that also manage rotations, a property of local features that is fundamental in many photogrammetric applications.

The contribution is twofold:

- to introduce the ENRICH dataset and its characteristics, consisting of three 2D and 3D synthetic and multi-scale outdoor datasets: an urban area (ENRICH-Aerial), a square (ENRICH-Square), and a statue (ENRICH-Statue). The benchmark includes, for the first time, high-resolution rendered images, depth maps, camera parameters, absolute orientation information, Ground Control Points (GCPs), and 3D models as ground truth data, allowing multiple investigations in the fields of photogrammetry and computer vision.
- to show the usefulness of the ENRICH dataset by performing several processing tests exploring some steps of the photogrammetric 3D reconstruction pipeline. Three example analyses are addressed with the ENRICH datasets: (i) the contribution of new deep learning-based local features for image matching and the evaluation of different Structure from Motion (SfM) pipelines; (ii) the influence of GCPs number and distribution in aerial mapping applications; (iii) the use of neural networks for monocular depth estimation.

The ENRICH benchmark dataset consists of three synthetic datasets generated from 3D scenes reproducing different scenarios, levels of detail, resolution, scale, lighting condition, and field of view (FoV): ENRICH-Aerial, ENRICH-Square, and ENRICH-Statue. Most of the 3D models used to compose the scenes come from real objects, assuring realistic textures.

The ENRICH-Aerial dataset is generated from an aerial image block of the city of Launceston, Australia [154]. The ENRICH-Square and the ENRICH-Statue are two ground-level datasets, capturing, respectively, a square surrounded by monumental buildings and a statue placed in its center. A virtual camera, based on the specifications of the Nikon D750 DSLR full-frame camera (sensor size 35.9x24mm, pixel size 5.95 μ m) with an image resolution of 6,016x4,016px (24MP) is used to acquire images of the scenes, and create the corresponding datasets. The virtual camera then acquires ideal images without lens distortions (pinhole camera model).

In all the scenes, GCPs were manually inserted in flat areas and well distributed at different heights. An overview of the acquisition setup for each

3.5 Proposed synthetic datasets for 3D reconstruction evaluation

one of the three datasets is given in Table 3.5.

Table 3.5: Summary of the acquisition setup of each dataset in the ENRICH benchmark.

Dataset	Camera	Focal Length	Images	Orientation	Lighting setup	GSD
ENRICH-Aerial	nadir	35mm / 5,882px	60	Landscape	Uniform light	2.5cm
	forward	70mm / 11,764px	60	Landscape	Uniform light	1.8cm
	backward	70mm / 11,764px	60	Landscape	Uniform light	1.8cm
	right	70mm / 11,764px	60	Landscape	Uniform light	1.8cm
	left	70mm / 11,764px	60	Landscape	Uniform light	1.8cm
	nadir 2	35mm / 5,882px	39	Landscape	Uniform light	3.0cm
ENRICH-Square	camera 1	35mm / 5,882px	50	Landscape & Portrait	Partly cloudy	0.8cm
	camera 2	50mm / 8,403px	50	Landscape	Clear sky	0.5cm
	camera 3	35mm / 5,882px	50	Landscape & Portrait	Sunrise	0.8cm
	camera 4	35mm / 5,882px	50	Landscape	Clear sky	1.0cm
ENRICH-Statue	camera 1	50mm / 8,403px	50	Landscape	Partly cloudy	0.69mm
	camera 2	35mm / 5,882px	50	Portrait	Clear sky	0.64mm
	camera 3	50mm / 8,403px	50	Landscape	Sunrise	0.70mm
	camera 4	35mm / 5,882px	50	Portrait	Cloudy	0.64mm

ENRICH builds on previous knowledge to generate the new datasets by integrating new functionalities in previously developed tools specifically tailored for the multi-scale tasks. In particular new functionalities have been introduced to support multiple camera paths and configurations, automatic export of GCPs coordinates and image visibility, and depth data.

Compared to the previous synthetic dataset IVL-SYNTHSfM, created mainly to evaluate SfM pipelines, ENRICH offers a more diverse set of scenes acquired with a variable range of cameras and scales. This makes ENRICH a multi-purpose dataset exploitable for different photogrammetry and computer vision tasks, including SfM applications. Additional details on 3D scenes, acquisition, and data available for each dataset are provided in the following subsection.

3.5.3.1 Data generation method

Data were generated using the popular 3D modeling and rendering software Blender [35] with the aid of the SfM Flow add-on [182]. This add-on has been enhanced to support multi-camera configurations as well as GCPs placement and export of their ground truth 3D position and visibility in the images.



Figure 3.9: Orthographic view of the GCPs placement on the ENRICH-Aerial dataset. Cross and round shapes as in Figure 3.4.

For the ENRICH-Aerial dataset, a Blender scene was created importing the 3D mesh and textures of a 3D Lidar scan of the city of Launceston [154]. A total of 26 GCPs of 50x50cm are positioned in the scene on flat or almost-flat surfaces at different elevations (Figure 3.9), with a cross (see Figure 3.4a) or a circular pattern (see Figure 3.4b). The size of each GCP is defined to have the cross’s thickness visible in at least 4 pixels. Each GCP is guaranteed to be visible in at least ten images. This GCPs configuration allows us to have them uniformly distributed in the scene and visible in many images. This allows researchers to select which GCPs to use for their experiments and still have targets available to be used as Check Points (see Section 3.6.3).

The acquisition is performed simulating a typical oblique aerial camera with five views (see Figure 3.10a-e): one nadir and four oblique views (forward, backward, left, and right). The nadir camera has a focal length of 35mm, whereas that of the oblique cameras is 70mm. The oblique cameras have an angle of 45° w.r.t. the nadir direction. The five cameras are rigidly mounted on a virtual flying platform at the same altitude, with the oblique ones having

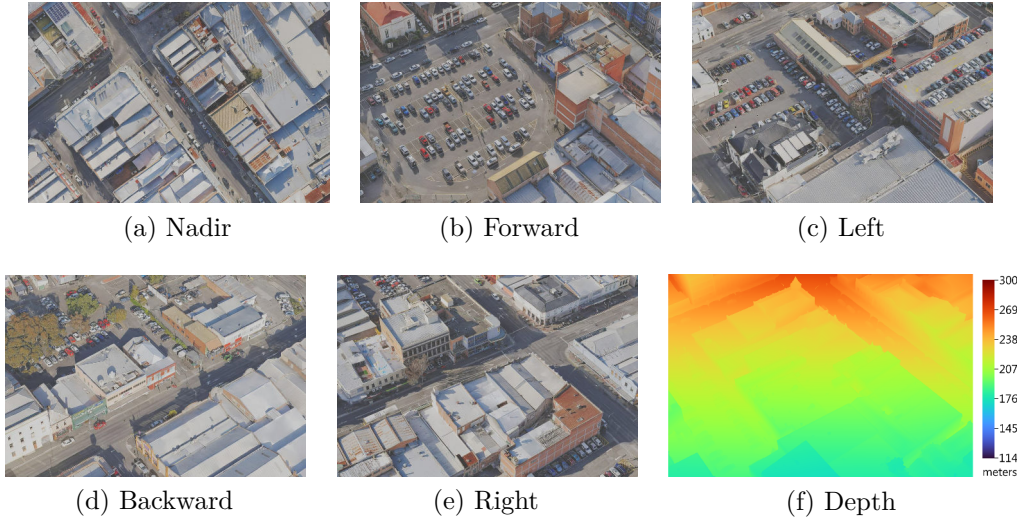


Figure 3.10: Sample images from the ENRICH-Aerial dataset.

a 20cm padding from the nadir camera in their viewing direction. The image acquisitions followed six parallel strips, with 10 acquisition points in each track, providing a total of 300 images. A second acquisition, orthogonal to the first one, is performed using only the nadir camera (nadir-2). This path consists of three parallel strips with 13 images each, for a total of 39 images. In both paths, the image overlap for the nadir images is 80% along the track and 60% across it, respectively. The flying heights are approximately 150m and 175m above the ground. These camera paths mimic real acquisition setups that allow detailed 3D model generation [249]. The first nadir camera has an average Ground Sample Distance (GSD) of 2.5cm, while the second nadir camera has a GSD of 3.0cm. The GSD of the oblique cameras ranges from 1.2 to 2.4cm (1.8cm at an average distance of 213m). The paths followed by the cameras are visible in Figure 3.11. In the ENRICH-Aerial dataset, the scene is illuminated only by a global environment white light since the diffuse texture of the 3D Launceston model already incorporates shadows. The shadows are embedded in the model due to the use of multiple images to build it. The blending of those images, acquired at different times, produced some artifacts in the textures. While this has no impact on the evaluation of geometry-related tasks, it may limit the usability of ENRICH-Aerial on



Figure 3.11: Camera paths on the ENRICH-Aerial dataset. In red the path followed by the first nadir and oblique cameras, in green the path followed by the second nadir camera.

texture-related tasks, such as image blending, shadow removal, and de-lighting. The images were generated employing Blender’s Eevee raster render engine that focuses on rendering speed while achieving Physically Based Rendering of materials.

In the ENRICH-Square dataset, several 3D models were used to build the virtual scene. It comprises 3D meshes of monumental buildings surrounding the square, statues, and trees. The tallest building is 27m high. The meshes of the buildings were generated using photogrammetry software such as Agisoft Metashape by the original authors; trees and walls were instead 3D modeled. In some cases, 3D model editing was required to solve geometry issues in the meshes (e.g. holes on the facades due to occlusions or dark areas). The whole square is surrounded by a hilly landscape model, providing a background for some far portions of the scene (i.e. behind the walls). Cross pattern GCPs of size 15x15cm are positioned on the facades of the buildings at different heights (see Figure 3.12); a total of 54 GCPs are available, and each one is

3.5 Proposed synthetic datasets for 3D reconstruction evaluation

visible in at least 16 images. This GCPs placement guarantees that they are uniformly distributed in the scene and visible in a large number of images.

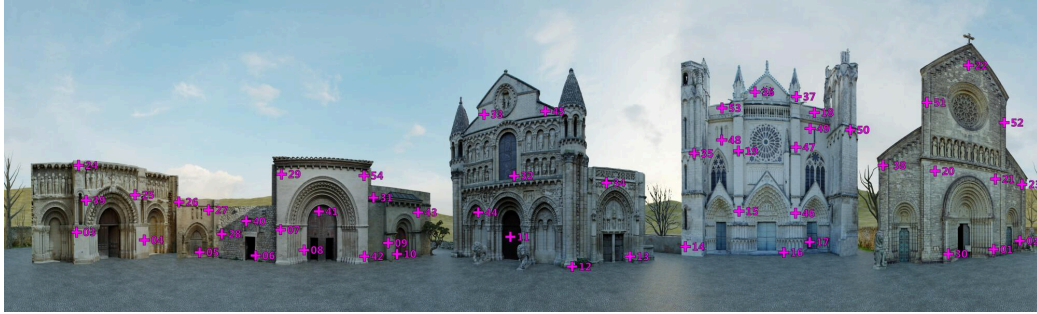


Figure 3.12: Equirectangular projection of the ENRICH-Square scene showing GCPs placement.

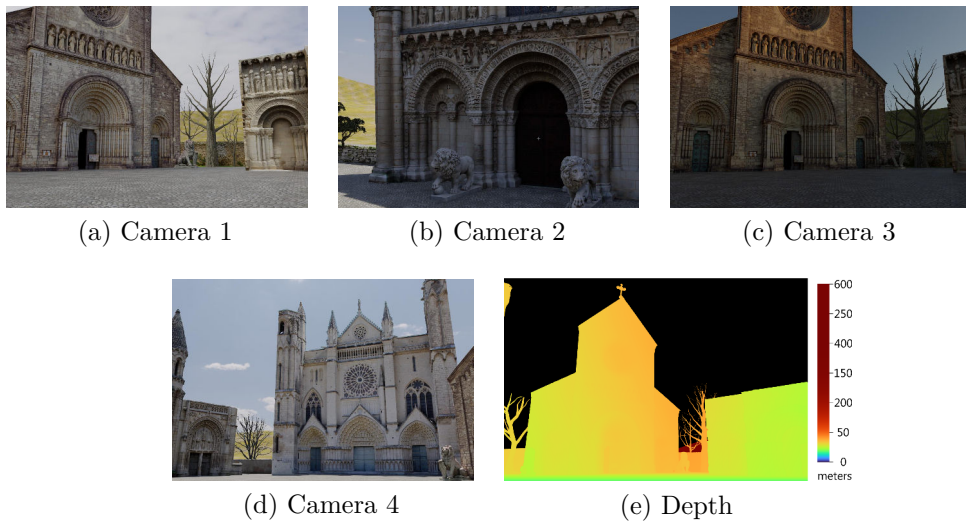


Figure 3.13: Sample images from the ENRICH-Square dataset.

Images are captured by four cameras (Figure 3.13a-d), each of them providing 50 images for a total of 200. Images for Camera1 are acquired following a circular path of a 5m radius around the center of the square with the camera looking toward the center of the circle; a first revolution provides 25 landscape images (1m height above ground), while the second further 25 portrait images (1.9m height above ground). The second camera follows two different circles, looking directly toward the buildings; while all images are landscape, the

3.5 Proposed synthetic datasets for 3D reconstruction evaluation

first 25 are acquired at 3.4m height from the ground, and with a circle of radius 2m, the last 25 following a circle of radius 6.25m at 4m height. The third camera uses the same configuration as the first one, but the acquisition poses slightly differ in position and orientation. The fourth camera follows the border of the square taking pictures of its opposite side from 1.3m above the ground. An overview of the paths followed by the cameras is visible in Figure 3.14. These camera paths have been chosen to cover all of the facades and provide overlap between images acquired by the same camera as well as across different cameras. Cameras 1, 3, and 4 use a focal length of 35 mm, whilst Camera2 has a 50mm focal length. The average GSD and depth for the cameras are, respectively: 8mm @ 46m, 5mm @ 38m, 8mm @ 46m, and 10mm @ 61m. Different high dynamic range image (HRDI) maps were used for lighting the scene. Camera1 images are captured in a partly cloudy sky. Cameras 2 and 4 acquisitions use clear sky conditions. Camera3 images are acquired at sunrise, thus with a predominant orange color and strong shadows. Considering the availability of roughness and normals maps for different 3D models (in addition to the diffusive color component), Blender’s Cycles path tracing engine has been used to render photorealistic images of the scene.

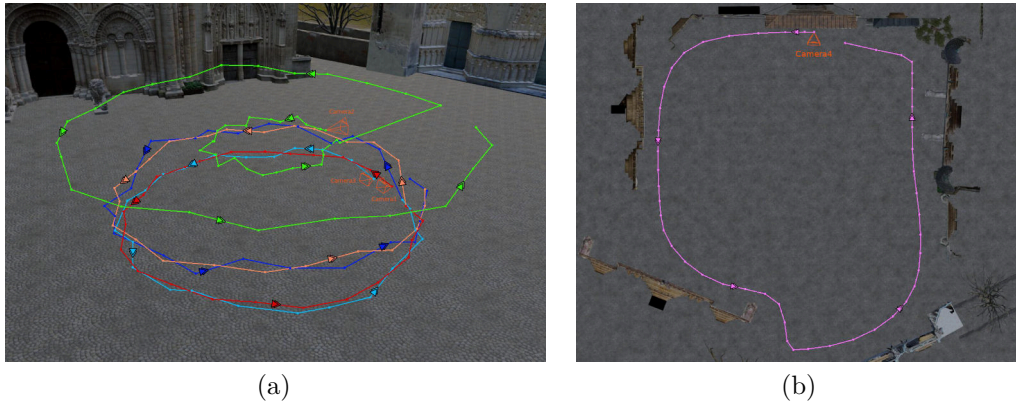


Figure 3.14: Camera paths on the ENRICH-Square dataset. (a) Camera 1, 2, and 3. The path of cameras 1 and 3 change color where the camera rotates from landscape to portrait, red to orange and cyan to blue respectively. (b) Path of the 4th camera.

The ENRICH-Statue dataset uses the same virtual setup of ENRICH-

Square, with an additional 2m high 3D statue of a hunter (textured mesh generated through photogrammetry) placed at the center of the square. Cross pattern GCPs of size 2x2cm are uniformly placed directly on the statue and its basement (Figure 3.15). While their quantity is lower than the ones used in the ENRICH-Square dataset, each of them appears in at least 62 images.



Figure 3.15: GCPs locations on the ENRICH-Statue dataset.

In this dataset, four cameras are used to acquire 200 pictures of the statue (50 images each). Some sample images are visible in Figure 3.16a-b. Camera1 and Camera3 captured landscape images rotating around the statue (radius 3.75m), looking at it slightly from the bottom (average height from ground 0.3m). Camera2 and Camera4 rotated around the statue (radius 2.25m), looking at it in portrait orientation from slightly above (height 1.9m). The path followed by each camera is visible in Figure 3.17. The path and orientations of the cameras ensure that the whole surface of the statue is covered in the image while also providing the overlap needed for the 3D reconstruction tasks. The average distance of the statue from the camera is 5.8m for the first and third cameras and 3.8m for the second and fourth, thus providing a GSD on the statue of 0.69mm, 0.70mm, 0.64mm, and 0.64mm, respectively. As in the ENRICH-Square dataset, different HRDIs were used for lighting the scene. The whole scene is illuminated by a partly cloudy sky for images acquired by the first camera, and with sunrise lighting for those by the third camera. For Camera2 and Camera4 the light is provided by a sunny sky and a cloudy sky, respectively. Blender’s Cycles path tracing engine has been used to render photorealistic images of the scene.

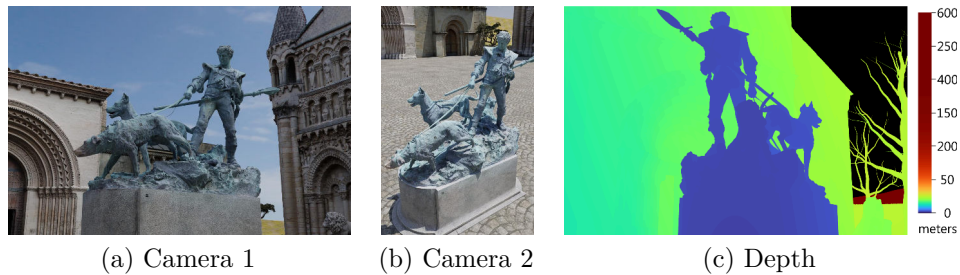


Figure 3.16: Sample images from the ENRICH-Statue dataset.

All the images have been rendered using an Nvidia Quadro RTX 6000 GPU. The GPU was awarded by Nvidia as part of the Academic Hardware Grant Program 2021. For each dataset, additional depth information is obtained directly from the Rendering Layers of Blender (Figures 3.10f, 3.13e, 3.16c).

3.6 Experimental results

The datasets proposed in Section 3.5 enable researchers to test and compare algorithms for different photogrammetric and computer vision applications. The next subsections present some experiments and possible usage of the datasets for testing new algorithms and solutions in some photogrammetric and computer-vision open research topics:

1. Evaluation of different SfM pipelines including out-of-the-box solutions and usage of different local features algorithms (Section 3.6.1);
2. Effect of the ground control points number and spatial distribution on 3D accuracy (Section 3.6.3);
3. Monocular depth estimation for outdoor architectural scenarios (Section 3.6.4).

The proposed tasks reflect some recent interests of the research community, focused on exploring new solutions for image orientation and 3D scene

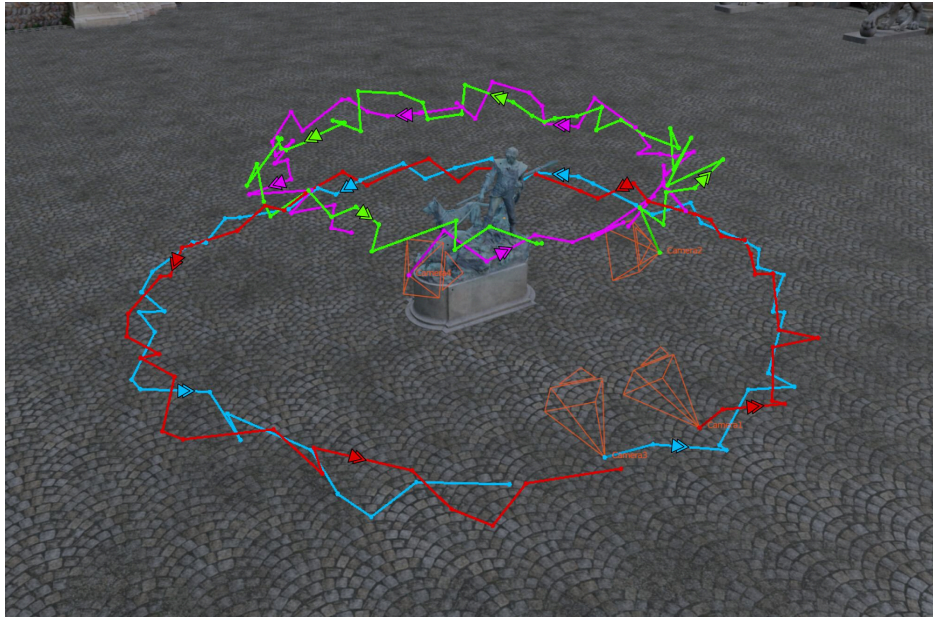


Figure 3.17: Camera paths on the ENRICH-Statue dataset. The paths followed by cameras 1-3 are shown in red, green, cyan, and purple respectively.

reconstruction and quality assessment in large-scale mapping.

The first one considers the contribution of neural networks for image matching in SfM. The SfM pipeline is very robust, but a high overlap between images and constant lighting conditions are typically needed for the 3D scene reconstruction. New matching algorithms based on neural networks have been proposed to overcome these limitations and obtain robust matches even in very challenging conditions. However, further investigations on their behavior in different survey scenarios are still needed, considering their different performances with multi-temporal [74, 179], wide-baseline [25, 50], or aerial datasets [225, 243]. The experiments focus on local features, but further SfM tasks have been recently revisited. As an example, new geometric verification approaches have been proposed in alternatives to RANSAC [57, 328], or end-to-end deep learning-based methods which handle the entire SfM pipeline. Section 3.6.1 offers an overview of several new deep learning-based local features, tested in challenging conditions, like strong variations in the viewpoint, light, scale, and rotation.

The second task deepens the effects of the GCPs configurations in aerial tri-

angulation (AT) and image orientation, a topic not fully explored. Especially in aerial mapping projects, the quality of the final products is highly correlated to their quantity and distribution within the scene. Most of the available SfM software estimates the exterior orientation parameters within a free-network bundle adjustment, followed by a 3D similarity transformation to move from an arbitrary to a real-world coordinate system. Investigations on the effects of GCPs spatial distribution on the 3D accuracy have been extensively proposed in the last years for UAV image blocks [90, 212, 296, 301] while a few studies have focused on their influence in the airborne case [101, 217]. The influence of several GCPs configurations in an aerial mapping project is demonstrated in Section 3.6.3.

Finally, in the state-of-the-art, various methods for the Monocular Depth Estimation (MDE) [9, 28, 155, 289, 312] task exists. Even in this case, the ENRICH dataset can represent a useful dataset for testing. These algorithms can predict a depth map starting from a single RGB image, and in the most challenging case, no prior knowledge about the scene or camera parameters is provided. These methods are usually trained on task-specific datasets such as KITTI [99] (autonomous driving) or NYUv2 [268] (indoor environments). While training on such datasets is suitable for a given application, it limits the ability of the models to generalize the depth estimation on different scenes. Recently, some approaches [162, 237] have improved the generalizability by using datasets depicting various scenes, from indoor environments to aerial views. The efficacy of MDE methods on new unseen data is investigated in Section 3.6.4.

3.6.1 Evaluation of SfM pipelines

The SfM pipeline consists of numerous tasks (see Section 3.2.1) that can be addressed by different algorithms. Evaluation of the most popular SfM pipelines (Section 3.6.1.1) is performed using the IVL-SYNTHSFM dataset. In addition, the ENRICH datasets offer the opportunity to assess the efficacy of different local features within the SfM pipelines (Sections 3.6.1.2 and 3.6.1.3).

In the last few years, new local features based on neural networks have appeared in addition to traditional methods. However, their performances, limits, and potentials for photogrammetric applications are still an open research topic. Scenes characteristics (like texture types, illumination, scale, and camera rotation) are critical elements in conditioning the algorithm’s performance. For these reasons, the tests consider both deep learning-based features and traditional ones.

The images and ground truth of the ENRICH-Statue and ENRICH-Square datasets were used for the experiments. The two scenes represent typical terrestrial photogrammetric surveys, both in terms of surveyed objects and the camera network. The available images are distortion-free and can be modeled as a pinhole camera with a known principal distance. Therefore, the evaluation relies on the accuracy of the checkpoints and external orientation parameters, neglecting considerations about internal orientation. The checkpoints are well-distributed targets on the scene whose 3D coordinates have been synthetically generated. In addition, the datasets offer images with relevant variations in scale, illumination, angle of view, and camera rotations, which are challenging situations for local feature extractors.

The datasets are here compared by using three different SfM pipelines:

1. **COLMAP + RootSIFT** [259], an open-source SfM software, available both with a command line interface and a graphical user interface that allows the user to customize many SfM parameters. This test uses the build-in RootSIFT [12] as local feature, followed by brute-force matching with near neighborhood ratio threshold set to 0.80, incremental reconstruction (resection-intersection), and local/global bundle adjustment.
2. **COLMAP + Deep learning local feature.** Several deep learning-based feature extractors have been tested, importing their keypoints and descriptors in COLMAP. Image matching and orientation have been performed with the default options, as in the previous method.
3. **Metashape**, the commercial software developed by Agisoft with its proprietary SfM pipeline implementation as reference.

For the comparison, 8,000 local features were extracted on images resized to 1,500x1,000 pixels because of the high computational performance required by the deep learning-based methods. Tiling images is the alternative approach for dealing with full-size images, as proposed in [243].

When comparing different SfM pipelines, several metrics are possible. Often, the bundle statistics obtained downstream the SfM pipeline are used, [261], such as the number of correctly registered images, the number of triangulated 3D points, the mean track length (MTL) or multiplicity, the mean reprojection error (MRE), and the mean observations per image, which is the mean number of correct tie points. Previous works have highlighted how these metrics are often inconsistent with the actual accuracy of the reconstruction evaluated in the object space [24, 242]. Therefore, the results are also evaluated in the object space by using the root mean square error (RMSE) computed on a few well-recognizable object points (the targets provided by the ENRICH benchmark) and computing the RMSE on the center of projection (COP) of the cameras. Note that currently, COLMAP handles the bundle block adjustments in a free network. For this reason, the ground control points have been used to compute only the Helmert transformation and get a scaled model, and not to add constraints to the bundle adjustment.

3.6.1.1 The IVL-SYNTHSFM dataset

In this section popular SfM pipelines are evaluated and compared through the use of the IVL-SYNTHSFM datasets. In addition to the synthetic datasets, a real one is also used: Ignatius (Figure 3.18f) from the “Tanks and Temples” collection [150], whose 263 images have been acquired at a resolution of 1920x1080px. The physical height of the statue is 2.51 meters.

Among all the SfM pipelines listed in Section 3.2.1.2, here are compared the reconstruction results of COLMAP, Theia, OpenMVG, and VisualSfM because each one is a reference implementation. In particular, VisualSfM and COLMAP represent two remarkable developments of the incremental

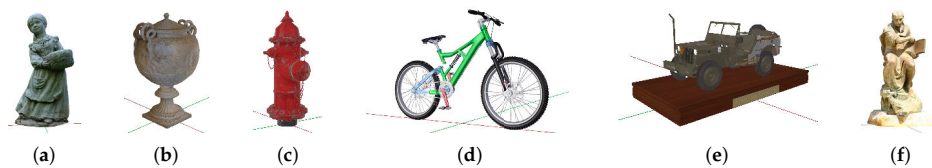


Figure 3.18: 3D models used for synthetic dataset generation and Ignatius ground truth: (a) Statue. (b) Empire Vase. (c) Hydrant. (d) Bicycle. (e) Jeep. (f) Ignatius.

SfM pipeline with improvements in accuracy and performance compared to previous state-of-the-art implementations. Theia and OpenMVG are instead two ready-to-use SfM and multi-view geometry libraries that implement reconstruction algorithms and allow to build SfM pipelines that meet specific needs.

Evaluation of dense 3D reconstructions is performed by pairing the chosen SfM pipeline with CMVS/PMVS [84] as the MSV reference algorithm because it is a widely used state-of-the-art implementation and is also natively supported by all the SfM pipelines considered. The use of a single MVS pipeline with the same configuration parameters for all the reconstructions allows us to evaluate and compare the dense results based on the quality of the sparse SfM reconstruction. In such a way, no other variables affect the reconstruction process. Here are reported the results of evaluations by using the method described in Section 3.3. Results are reported in Tables 3.6–3.9 and some examples of the reconstructed dense point cloud are visible in Figure 3.19.

Table 3.6: SfM cloud evaluation results. \bar{x} is the average distance of the point cloud from the ground truth and s its standard deviation. N_p is the number of reconstructed points.

Model	COLMAP			OpenMVG			Theia			VisualSFM		
	\bar{x} [m]	s [m]	N_p	\bar{x} [m]	s [m]	N_p	\bar{x} [m]	s [m]	N_p	\bar{x} [m]	s [m]	N_p
Statue	0.034	0.223	9k	0.057	0.267	4k	0.020	0.039	8k	0.185	0.236	6k
Empire Vase	0.005	0.152	8k	0.013	0.191	2k	0.002	0.005	8k	0.007	0.013	5k
Bicycle	0.042	0.365	5k	0.156	1.705	7k	0.027	0.086	2k	0.056	0.796	4k
Hydrant	0.206	0.300	2k	–	–	28	0.045	0.123	89	0.029	0.032	1k
Jeep	0.053	1.058	6k	0.057	0.686	4k	0.012	0.016	8k	0.055	0.124	5k
Ignatius	0.009	0.021	23k	0.013	0.032	12k	0.023	0.022	10k	0.054	0.124	14k

3.6 Experimental results

Table 3.7: SfM camera pose evaluation results. N_c is the percentage of used cameras. \bar{x} is the mean distance from ground truth of reconstructed camera positions and s_x its standard deviation. \bar{r} is the mean rotation difference from ground truth of reconstructed camera orientations and s_r its standard deviation. n.a. means value not available.

Model	COLMAP					OpenMVG					Theia				VisualSfM					
	N_c	\bar{x} [m]	s_x [m]	\bar{r} [°]	s_r [m]	N_c	\bar{x} [m]	s_x [m]	\bar{r} [°]	s_r [m]	N_c	\bar{x} [m]	s_x [m]	\bar{r} [°]	s_r [m]	N_c	\bar{x} [m]	s_x [m]	\bar{r} [°]	s_r [m]
Statue	100	0.08	0.01	0.04	0.05	100	0.27	0.03	0.47	0.22	100	1.86	0.09	0.45	0.22	100	1.45	0.91	3.55	2.88
E. Vase	100	0.01	0.01	0.51	0.05	83	0.78	1.62	32.19	64.89	100	0.13	0.07	0.91	0.35	94	0.15	0.14	4.91	5.07
Bicycle	88	0.04	0.02	0.25	0.19	94	0.60	1.09	7.00	12.53	37	0.60	0.03	1.10	0.31	47	0.27	0.14	1.32	1.03
Hydrant	82	2.63	2.09	72.28	64.98	3	-	-	-	-	6	3.49	0.29	174.27	0.51	80	2.43	1.76	66.29	58.90
Jeep	63	0.04	0.02	0.26	0.11	92	0.24	1.32	4.80	26.33	95	0.43	0.42	1.33	5.67	83	1.02	2.79	9.68	22.84
Ignatius	100	n.a.	n.a.	n.a.	n.a.	100	n.a.	n.a.	n.a.	n.a.	100	n.a.	n.a.	n.a.	n.a.	100	n.a.	n.a.	n.a.	n.a.

Table 3.8: MVS cloud evaluation results. \bar{x} is the average distance of the point cloud from the ground truth and s its standard deviation. N_p is the number of reconstructed points.

Model	COLMAP			OpenMVG			Theia			VisualSfM		
	\bar{x} [m]	s [m]	N_p	\bar{x} [m]	s [m]	N_p	\bar{x} [m]	s [m]	N_p	\bar{x} [m]	s [m]	N_p
Statue	0.009	0.023	75k	0.008	0.027	86k	0.010	0.011	84k	0.065	0.049	76k
Empire Vase	0.001	0.001	390k	0.001	0.004	246k	0.002	0.002	356k	0.005	0.007	240k
Bicycle	0.013	0.012	74k	0.062	0.146	69k	0.018	0.020	46k	0.021	0.025	44k
Hydrant	0.008	0.017	42k	-	-	-	0.080	0.147	11k	0.008	0.014	40k
Jeep	0.010	0.016	236k	0.008	0.016	471k	0.014	0.019	448k	0.048	0.056	281k
Ignatius	0.004	0.004	155k	0.003	0.004	161k	0.018	0.019	109k	0.017	0.031	76k

Table 3.9: Pipelines execution times in seconds and peak memory usage in MB.

Model	COLMAP				OpenMVG				Theia				VisualSfM			
	SfM		MVS		SfM		MVS		SfM		MVS		SfM		MVS	
	t [s]	RAM	t [s]	RAM	t [s]	RAM	t [s]	RAM	t [s]	RAM	t [s]	RAM	t [s]	RAM	t [s]	RAM
Statue	59	897	86	1062	43	1359	115	1300	196	1984	98	1249	86	1406	144	1452
Empire Vase	53	897	154	2101	28	628	130	1734	129	1988	134	1926	62	1226	159	2095
Bicycle	98	896	117	1356	57	1467	146	1720	63	1722	58	548	64	1226	68	641
Hydrant	19	894	55	793	16	1547	-	-	17	2048	3	1249	36	997	56	1452
Jeep	38	897	121	1812	69	1550	275	3209	213	2083	280	3293	109	1406	254	3078
Ignatius	1225	1825	430	5082	401	1555	494	5926	992	2588	484	5626	1639	2381	345	4742

SfM pipelines have generated sufficient information to allow dense reconstruction on all datasets except the Hydrant one. That dataset has a low geometric complexity, a high level of symmetry, and an almost uniform texture; for these reasons, SfM pipelines were not able to find enough correspondence

between images and thus cannot generate a good reconstruction. The worst result was obtained with the OpenMVG pipeline and was not possible to run the MVS pipeline. COLMAP is the pipeline that achieves better results on average; even when it does not generate the best reconstruction it achieves good results.


















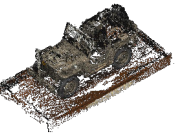
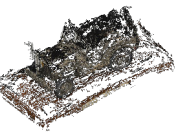




	COLMAP	OpenMVG	Theia	VisualSFM
Statue				
Empire Vase				
Bicycle				
Hydrant		n.a.		
Jeep				
Ignatius				

Figure 3.19: Example of dense point clouds using CMVS/PMVS on different SfM reconstructions.

Results for dataset Ignatius do not include camera pose evaluation because no information about camera pose ground truth is included in the dataset. This real dataset includes many elements besides the main object of the reconstruction; for this reason, the reconstructed clouds have a high number of points that do not belong to the statue and thus must be removed. An evaluation of system resources usage was also done and COLMAP is efficient also in this aspect; using as many resources as possible can complete the reconstruction in less time than the other pipelines.

The SfM reconstruction generated by pipeline Theia for the Statue dataset shows that the obtained sparse point cloud is the best for that dataset, but the camera poses are not accurate. These imprecisions are relative to camera positioning and not the camera rotation estimation which is always accurate. Further analysis shows that all the camera positions are estimated further away from the ground truth but in the correct viewing direction; for this reason, the camera orientation is correct. Because the position error is constant and applies to all the cameras, it still allows the reconstruction of an accurate point cloud.

3.6.1.2 The ENRICH-Statue dataset

The ENRICH-Statue dataset presents images both in landscape and portrait orientation, requiring rotation invariant local features. The LF-Net [214] deep learning approach has been chosen, an end-to-end convolutional neural network among the few local features trained to be invariant to rotations. Its results are compared against RootSIFT and Metashape.

In Table 3.10, the RMSE for the three pipelines are shown beside some bundle statistics reported for completeness. All the tested methods obtained an RMSE on the GCPs of about one-third of the GSD (0.69-0.64mm). As expected, worse results have been achieved for the 3D coordinates of the COPs, with an RMSE value similar to the GSD. Therefore, for this dataset, no significant differences among the three tested methods were found, apart from a slightly worse behavior of RootSIFT. In Figure 3.20a is reported

Table 3.10: Bundle statistics and RMSE on GCPs and COPs for different SfM pipelines using the ENRICH-Statue dataset.

Method	RMSE on GCPs [mm]	RMSE on COPs [mm]	MTL	MRE [pix]	Total keypoints
COLMAP + RootSIFT	0.23	1.63	4.6	0.65	86,525
COLMAP + LF-Net	0.16	0.69	4.5	0.43	99,140
Metashape	0.14	0.89	3.6	0.65	77,994

the camera network and the sparse point cloud processed by the RootSIFT implementation of COLMAP.

After the block orientation, the dense cloud was reconstructed for each method and compared in terms of cloud-to-cloud (C2C) distance with the point cloud extracted from the reference 3D models. Figure 3.20b shows an example of a dense point cloud obtained with COLMAP+RootSIFT, with an achieved average Cloud-to-Cloud distance of 1.34mm and a standard deviation of 0.74mm. Comparable results in terms of C2C distance were obtained for COLMAP+LF-Net and Agisoft Metashape. This first test thus shows small differences in the performance of hand-crafted and deep learning-based local features both in terms of RMSE on checkpoints and on the dense cloud accuracy.

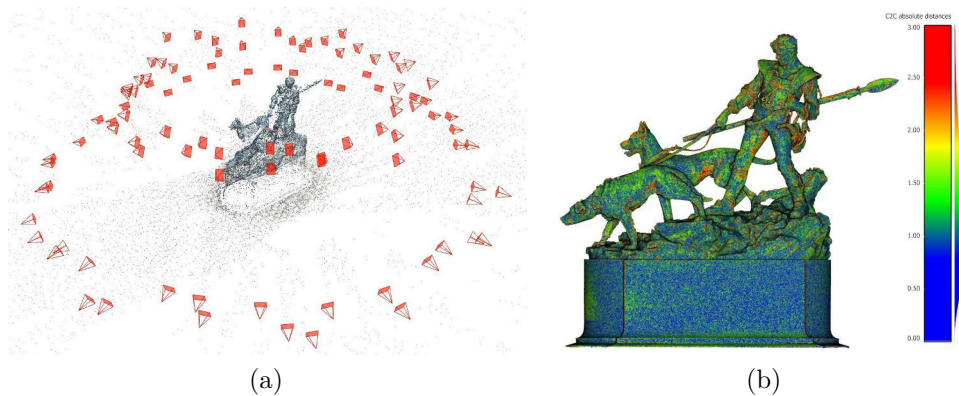


Figure 3.20: Results obtained with COLMAP + RootSift. (a) Sparse cloud and camera network. (b) Dense cloud with C2C absolute distance in millimeters.

3.6.1.3 The ENRICH-Square dataset

This section further deepens the local features comparison, using the ENRICH-Square dataset. Among the various deep learning-based methods, other rotational invariant features were chosen beside LF-Net: KeyNet + AffNet + HardNet [18, 192, 233] and RoRD [222]. KeyNet, AffNet, and HardNet are respectively a deep learning-based detector, orientation estimator, and descriptor. In the rest of the text, we will refer to this combination simply as KeyNet. Rotation invariance is necessary to optimize the sensor calibration, although many deep learning-based methods still neglect this aspect. ALIKE [336] is the most recent keypoint extractor, and it has been included in the tests, although it is not trained to be invariant to rotations.

Results are reported in Table 3.11 in terms of RMSE on the targets distributed on the scene. In this test, RootSIFT still represents the state-of-the-art approach, with an RMSE of an order of magnitude lower than KeyNet, while all the other methods failed to register the entire image block due to relevant scale variation and sensor rotation. Agisoft Metashape performed similarly to RootSIFT. Figure 3.21 shows the camera network and a closer view of a rectangular target (white cross on black background).

COLMAP provides further processing statistics, also reported in Table 3.11. It is worth noting the excessively high mean reprojection error (MRE) of RoRD, equal to 1.298 pixels, and how KeyNet has the highest mean track length (MTL), even if it extracted fewer keypoints (6,775 kpts, less than the required 8,000).

Figures 3.22, 3.23, 3.24, and 3.25 show a few image pairs representing interesting challenging conditions for the local features.

Table 3.11: RMSE and statistics for the comparison of deep learning-based and hand-crafted local features with the ENRICH-Square dataset.

Local Features	RMSE [cm] on CPs	MRE / ST.DEV [pix] on CPs	MTL	Mean observations per image	Total keypoints	MRE [pix] on tie points
COLMAP + RootSIFT	0.333	0.122 / 0.015	8.47	5,436	7,989	0.306
COLMAP + KeyNetAffNetHardNet	1.523	0.515 / 0.067	9.37	4,864	6,775	0.680
COLMAP + LF-Net	Failed to register all images		5.59	4,489	8,000	0.647
COLMAP + RoRD	Failed to register all images		6.37	4,892	7,275	1.298
COLMAP + ALIKE	Failed to register all images		6.81	4,198	8,000	0.689
Metashape	0.286	0.132 / 0.013	4.66	–	8,000	0.428

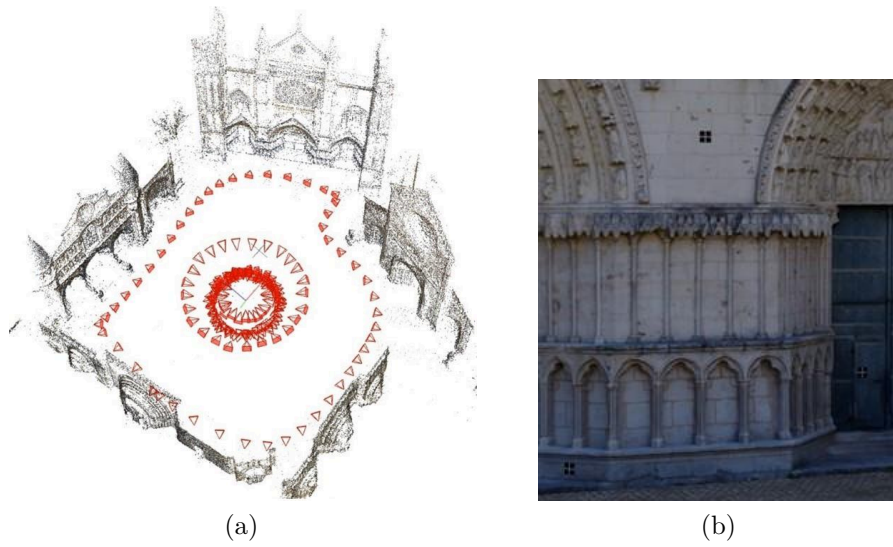


Figure 3.21: Example of the ENRICH-Square sparse reconstruction performed with RootSIFT and COLMAP (a), and the detail of three white crosses on a black background used to materialize the targets (b).

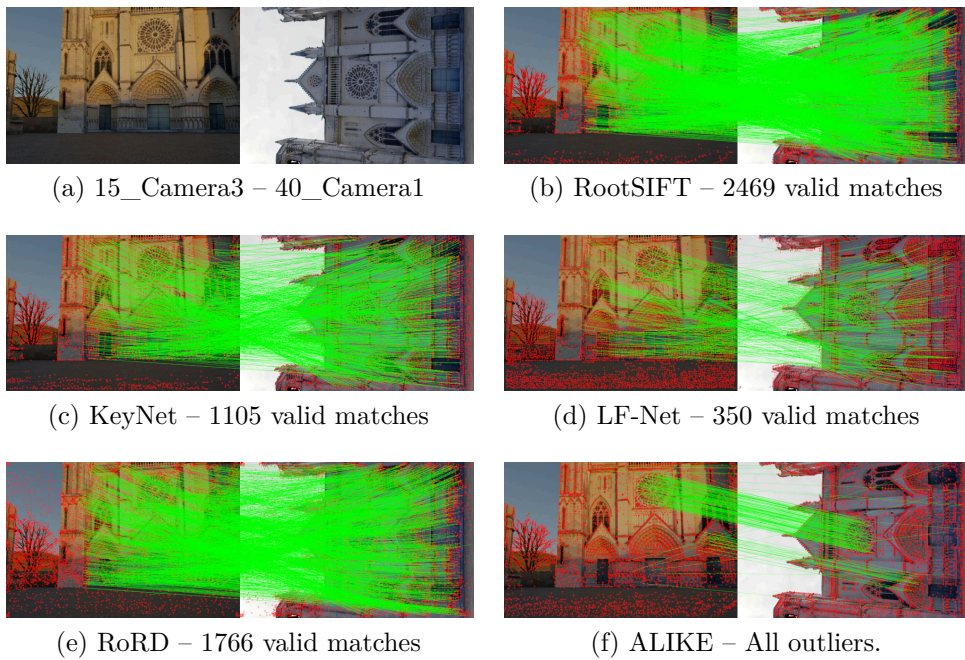


Figure 3.22: Tie point extraction under rotation and illumination changes.

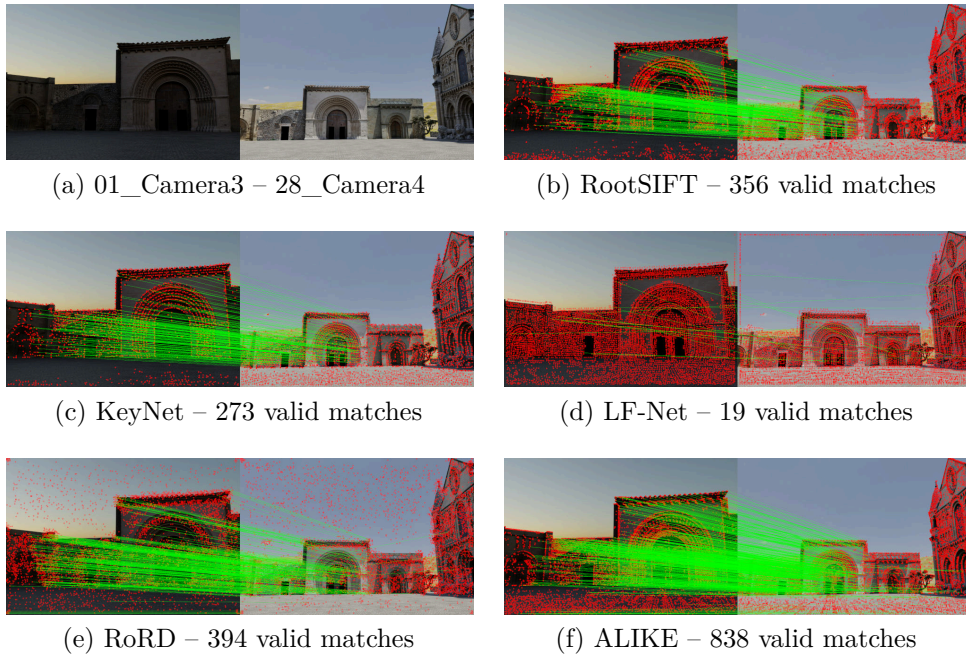


Figure 3.23: Tie point extraction under scale and illumination changes.

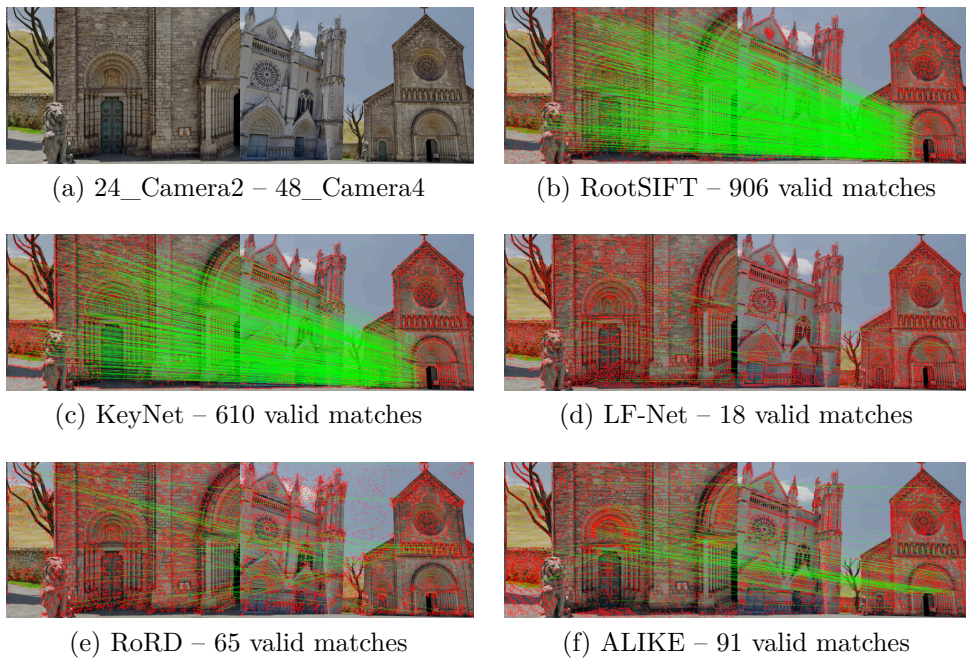


Figure 3.24: Tie point extraction under large scale changes.

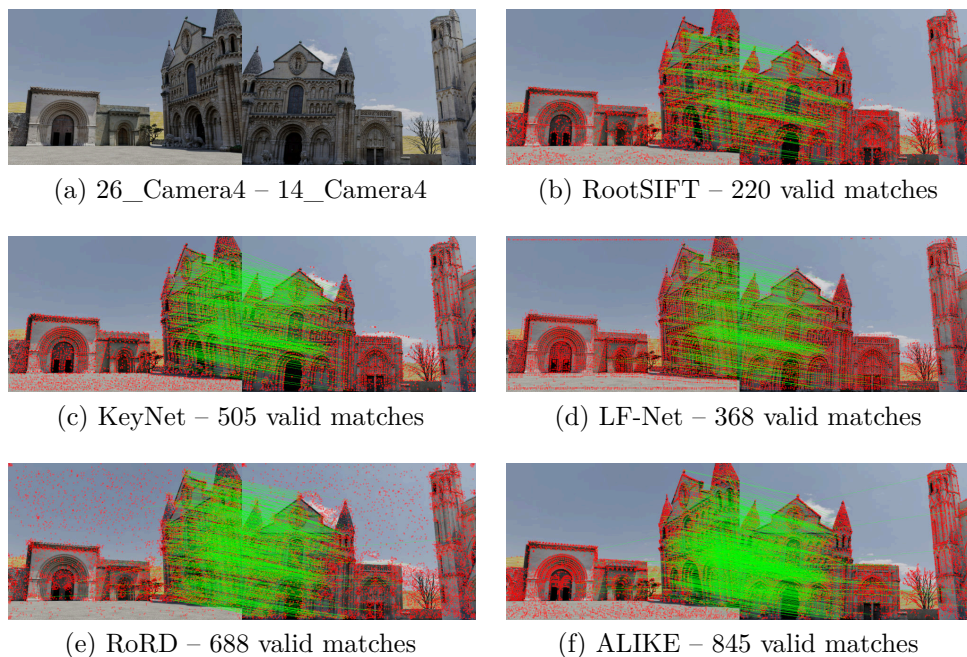


Figure 3.25: Tie point extraction under large view changes.

Figure 3.22 shows the case of a 90 degrees rotation and lighting changes. Except for ALIKE, all others methods handled sensor rotations properly, showing that ALIKE is not trained for rotation invariance. Note how the number of LF-Net valid matches is significantly lower than the other methods. Figure 3.23 shows a significant scale variation and changes in illumination, adequately addressed by all the methods with the LF-Net exception. In Figure 3.23e can also be noted that RoRD is the only method to find keypoints in homogeneous sky areas. Figure 3.24 remarkably accentuates the scaling factor with respect to Figure 3.23, demonstrating that the scale is a further critical factor for learning-based methods, in addition to the rotation invariance [25, 242]. Finally, Figure 3.25 reports the case of an extreme perspective variation and shows the good performance of deep learning-based methods in managing these situations.

This section showed how the ENRICH dataset provides a variety of challenging conditions useful for testing new SfM algorithms, such as local detectors and descriptors. In particular, it appears that the new deep learning-

based local features sometimes fail to orient the whole image block or are less accurate than classical methods such as RootSIFT, because of strong rotations or scale variations.

3.6.2 The effects of depth of field, motion blur, and light changes on camera registration

This section explores the impact of Depth of Field (DoF), Motion Blur (MB), and variations of light intensity and position on the alignment of camera pose in the SfM pipeline. Evaluation is performed using the IVL-SYNTHSFM-v2 dataset, which provides challenging data for those effects in both cameras and lighting.

Experiments on the IVL-SYNTHSFM dataset (see Section 3.6.1.1) showed that the best results (in terms of both point cloud and camera alignment) are obtained using the SfM pipeline implementation provided by COLMAP. For this reason, COLMAP has been selected to run this experiment. The decision is further motivated by the fact that the IVL-SYNTHSFM-v2 dataset is a direct revision and extension of the previously used IVL-SYNTHSFM and maintains the same single object subjects.

Tables from 3.12 to 3.16 report the results of the experiment. For each subject and each combination of effects (set name, see Table 3.4), the statistics of reconstruction provided by COLMAP are reported. In detail, the number of registered images, keypoints, Mean Track Length (MTL), mean observations per image, and Mean Reprojection Error (MRE).

Across all subjects, except the Statue, the introduction of light intensities and position variations (**ms**, **ms-*** sets) drastically reduces the number of registered images as well as the mean observations per image and the MTL. The presence of a non-uniform background in the images simplifies the task of reconstruction for all subjects and allows to correctly register all images in the fixed light and no camera effect case (**fs**). However, the Hydrant remains the most critical subject due to its size, filling of images, symmetric geometry, and

Table 3.12: Effects on the Bicycle dataset.

Set name	Total number of images	Registered images	Total keypoints	MTL	Mean observations per image	MRE [pix]
fs	100	100	15,016	5.546	832.88	0.619
fs-dof	100	100	14,820	5.582	827.37	0.623
fs-mb	100	100	14,427	5.580	805.10	0.640
fs-dof-mb	100	100	14,383	5.584	803.20	0.642
ms	100	44	2,940	4.347	290.48	0.939
ms-dof	100	44	2,947	4.343	290.89	0.944
ms-mb	100	44	3,438	4.288	335.11	0.971
ms-dof-mb	100	42	2,879	4.378	300.14	0.945

Table 3.13: Effects on the Empire Vase dataset.

Set name	Total number of images	Registered images	Total keypoints	MTL	Mean observations per image	MRE [pix]
fs	100	100	32,693	5.640	1,843.91	0.391
fs-dof	100	100	41,789	5.698	2,381.26	0.403
fs-mb	100	100	26,225	5.534	1,451.45	0.460
fs-dof-mb	100	100	36,566	5.460	1,996.79	0.432
ms	100	33	7,239	3.735	819.36	0.394
ms-dof	100	47	13,659	3.556	1,033.51	0.370
ms-mb	100	57	3,694	3.231	209.42	0.504
ms-dof-mb	100	40	9,609	3.589	862.35	0.381

Table 3.14: Effects on the Hydrant dataset.

Set name	Total number of images	Registered images [†]	Total keypoints	MTL	Mean observations per image	MRE [pix]
fs	100	100	48,029	6.773	3,253.13	0.366
fs-dof *	100	100	56,830	6.733	3,826.87	0.404
fs-mb	100	100	46,607	6.542	3,049.28	0.406
fs-dof-mb *, [§]	100	100	53,707	6.467	3,473.62	0.431
ms	100	96 (17)	28,807	4.035	1,211.04	0.397
ms-dof	100	96 (27)	33,492	4.001	1,395.99	0.394
ms-mb	100	94 (40)	24,586	3.972	1,039.10	0.419
ms-dof-mb	100	96 (40)	26,999	3.876	1,090.21	0.440

[†] In brackets the images registered with wrong pose;

* Sequential matching used instead of exhaustive matching;

[§] Manually fixed initial image pair.

Table 3.15: Effects on the Jeep dataset.

Set name	Total number of images	Registered images	Total keypoints	MTL	Mean observations per image	MRE [pix]
fs	100	100	26,327	4.945	1,301.98	0.417
fs-dof	100	100	25,654	4.944	1,268.49	0.430
fs-mb	100	100	25,107	4.875	1,224.14	0.443
fs-dof-mb	100	100	23,679	4.912	1,163.12	0.466
ms	100	68	5,415	3.202	255.00	0.449
ms-dof	100	92	7,112	3.285	253.99	0.482
ms-mb	100	57	3,694	3.231	209.42	0.504
ms-dof-mb	100	49	3,505	3.151	181.03	0.666

Table 3.16: Effects on the Statue dataset.

Set name	Total number of images	Registered images	Total keypoints	MTL	Mean observations per image	MRE [pix]
fs	100	100	9,417	7.484	704.79	0.580
fs-dof	100	100	9,349	7.505	701.70	0.583
fs-mb	100	100	8,763	7.734	677.77	0.614
fs-dof-mb	100	100	8,826	7.680	677.89	0.614
ms	100	100	6,111	6.035	368.82	0.621
ms-dof	100	100	6,112	6.020	368.00	0.619
ms-mb	100	100	6,162	5.971	367.96	0.634
ms-dof-mb	100	100	6,066	5.987	363.20	0.634

uniform texture. On this subject, images fail to register in various cases due to the combination of DoF, MB, and characteristics of the subject. Moreover, in the **fs-dof-mb** it was necessary to perform reconstruction by using sequential matching and fixing the initial pair of images. The **fs-dof** set required the use of sequential matching instead of the exhaustive one. Furthermore, on the **ms*** sets almost all images were registered but several of them were aligned in the wrong locations and orientations leading to unusable point clouds. Results are visible in Figure 3.26.

In a few cases, the ***-dof-mb** sets provide MREs that are slightly better than the ***-mb** set but slightly worse than the ***-dof** set. This behavior is traceable to two different reasons. In some cases, the number of images correctly registered in the ***-dof-mb** is lower than in the other ones thus the error is computed on a more robust registered images block. In other cases the number of registered images is similar and the lower error is due to the

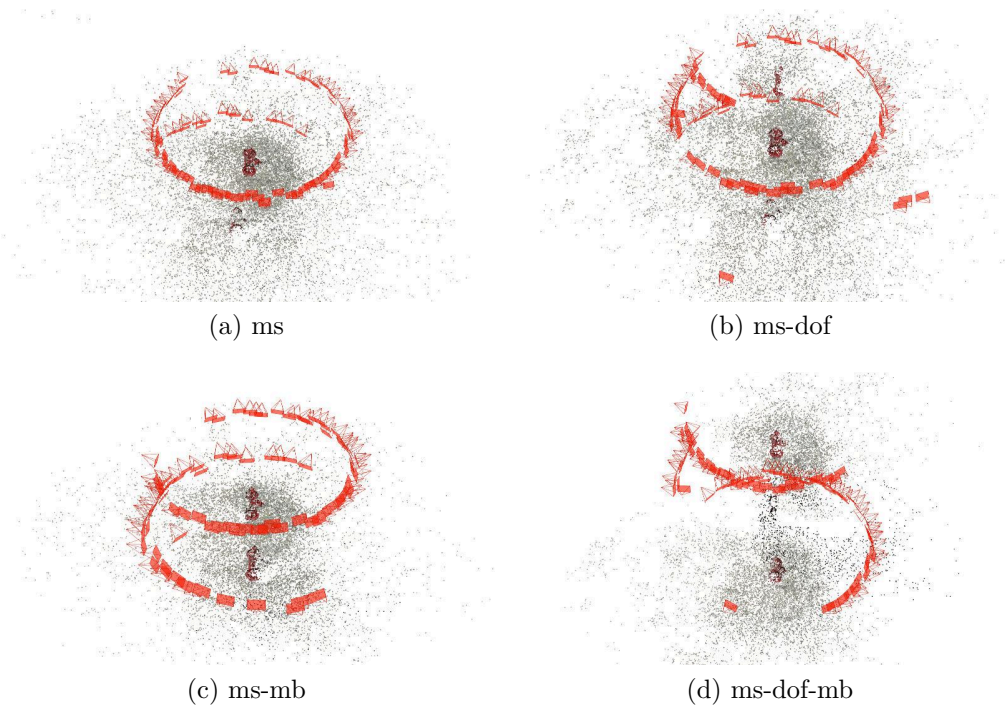


Figure 3.26: Results of image registration on the IVL-SYNTHSFM-v2 – Hydrant dataset.

effects of DOF that reduces the mismatch of keypoints located on background elements thus enforcing alignment using more points belonging to the object and leading to a robustly registered image block.

3.6.3 The effects of Ground Control Points spatial distribution on the 3D accuracy

This section examines the impact of Ground Control Points (GCPs) number and spatial distribution within the scene on 3D accuracy, exploiting the ENRICH-Aerial dataset.

In aerial triangulation (AT), GCPs are used to georeference data and optimize camera orientation by providing additional information for refining the bundle adjustment. Their configuration (number and distribution) affects the

reconstruction results. Most of the recent and available automatic processing solutions for exterior orientation initially adopt a free-network approach for the bundle adjustment and GCPs coordinates are then introduced for georeferencing data and refining the orientation results. This approach was followed in the tests for the exterior orientation, while no further considerations were made for the interior orientation being the images without distortions.

Different GCPs configurations were investigated for processing the block of 300 oblique and nadir images at the original image size (6,016x4,016 pixels). In the reported experiments, a variable and increasing number of targets (four to twelve) with different distributions were used as GCPs, while the rest as Check Points (CPs). Both the image and spatial coordinates provided in the dataset were employed. Tested configurations and distribution schemes (Figure 3.27) are defined as follows:

- Configuration 1: four GCPs aligned on the shorter side of the block;
- Configuration 2: four GCPs aligned along the diagonal of the block;
- Configuration 3: four GCPs distributed on the edges of the block;
- Configurations 4 to 11: an increasing number of GCPs (from five to twelve), distributed on the edges, and progressively adding points inside and outside the central area of the block.

In order to assess the achieved accuracy for each configuration, the root mean square error (RMSE) between the ground truth and the estimated 3D coordinates after the bundle adjustment was calculated for the CPs. RMSE of planimetric (R_x and R_y), vertical (R_z), and 3D (R_{xyz}) errors are reported in Table 3.17 and visualized in Figure 3.28.

As expected, the worst results on the CPs occurred when four GCPs were arranged along one line (Configurations 1 and 2), with a significant improvement of metrics when the same quantity is uniformly displaced on the edges of the block (Configuration 3). Increasing the number of GCPs and their spread distribution within the area generates less relevant changes in error metrics. While Configuration 4 confirms that adding a point in

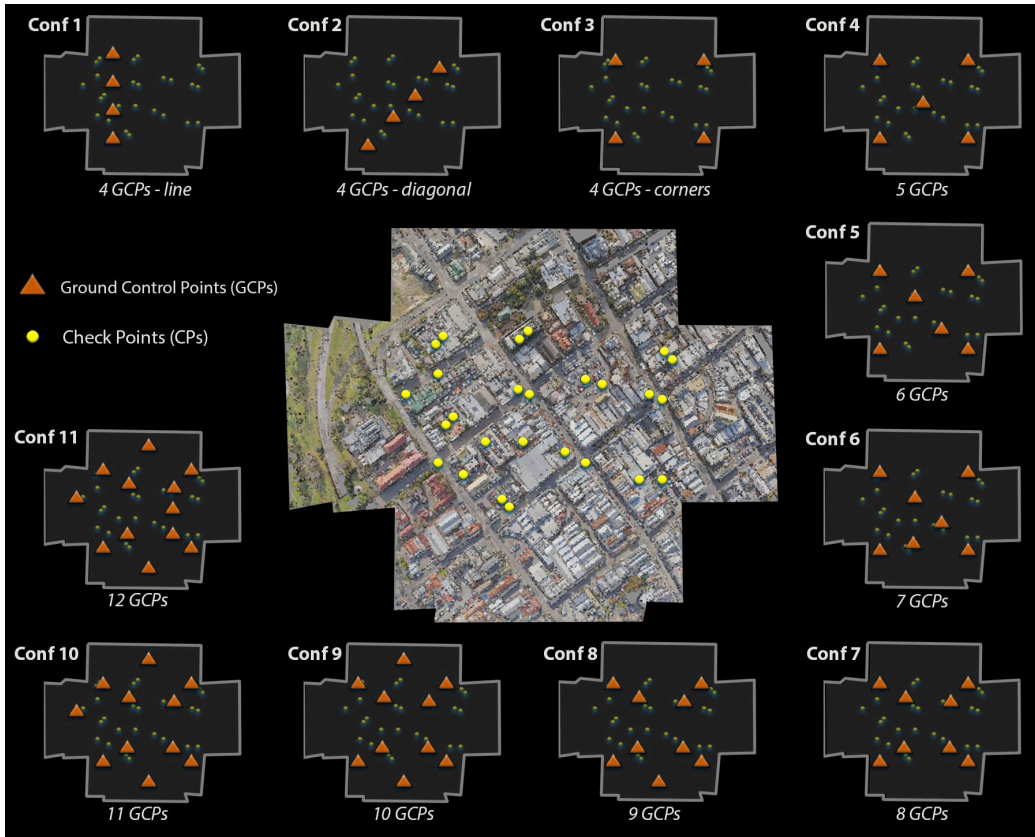


Figure 3.27: Schemes of the eleven GCPs configurations, varying in the number and distribution of ground and check points.

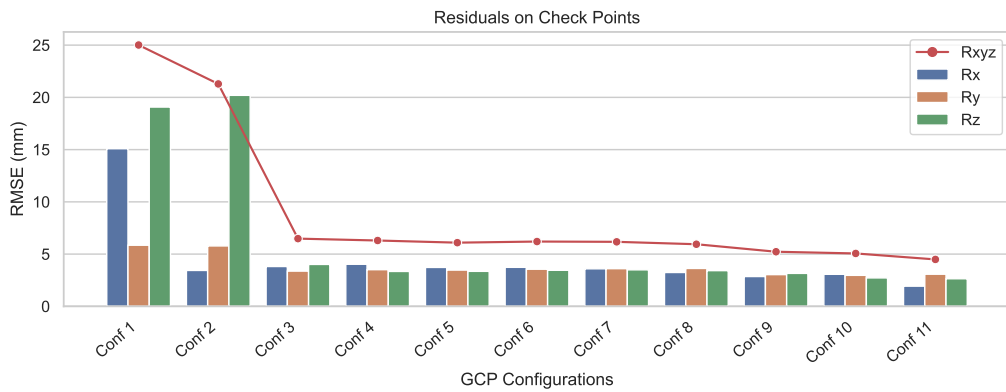


Figure 3.28: RMSE residuals on Check Points (CPs) with eleven GCPs configurations.

Table 3.17: RMSE of the planimetric (Rx and Ry), vertical (Rz) and global (Rx,y,z) residuals on Check Points (CPs) with the eleven Ground Control Points (GCPs) configurations.

	Rx [mm]	Ry [mm]	Rz [mm]	Rxyz [mm]
Conf 1 (4 GCPs line)	15.085	5.849	19.073	25.011
Conf 2 (4 GCPs diagonal)	3.436	5.782	20.195	21.286
Conf 3 (4 GCPs edges)	3.812	3.372	4.012	6.481
Conf 4 (5 GCPs)	4.028	3.509	3.340	6.300
Conf 5 (6 GCPs)	3.725	3.470	3.351	6.095
Conf 6 (7 GCPs)	3.730	3.553	3.452	6.201
Conf 7 (8 GCPs)	3.592	3.606	3.494	6.174
Conf 8 (9 GCPs)	3.240	3.627	3.417	5.944
Conf 9 (10 GCPs)	2.853	3.036	3.155	5.227
Conf 10 (11 GCPs)	3.067	2.967	2.722	5.062
Conf 11 (12 GCPs)	1.931	3.075	2.641	4.490

the middle of the block increases the altimetric accuracy, a more marked improvement is visible only in the last configuration. Furthermore, the eleven sparse point clouds were compared with the reference mesh model provided in the ENRICH-Aerial dataset for a more in-depth analysis and quality assessment of the results achieved with the different GCPs configurations. The orthogonal distance between each point and the corresponding triangle surface for each configuration scenario returned no significant differences in the standard deviation (for all the cases, around 0.1 meters). In contrast, more relevant divergences are evident when comparing the average distances, as shown in Figure 3.29. Also for this case, the worst metrics are related to the first two configurations, with a clear improvement in the other scenarios. Tests with the ENRICH-Aerial dataset show that 3D accuracy is more affected by the GCPs' spatial distribution with respect to their number. Four GCPs distributed on the edges of the aerial block are already sufficient for a clear improvement of the error metrics.

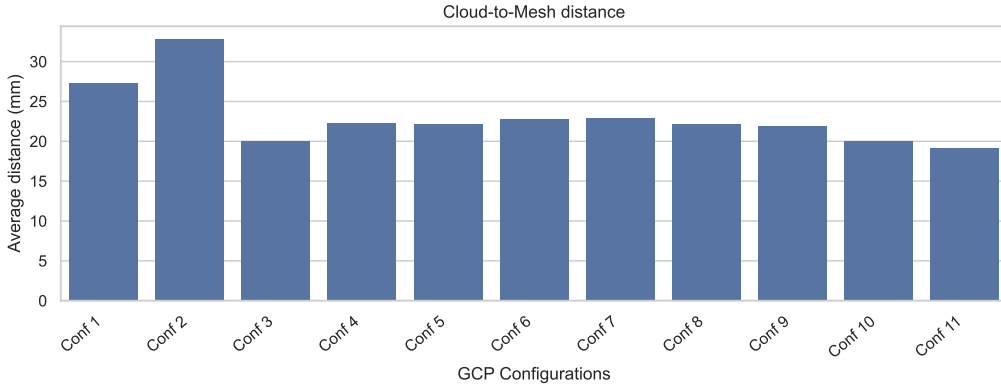


Figure 3.29: Average Cloud-to-Mesh distance (mm) between the eleven sparse point clouds computed in different GCPs configuration scenarios and the reference mesh model provided in the ENRICH-Aerial dataset.

3.6.4 Monocular depth estimation

Monocular Depth Estimation (MDE) is the task of estimating the distance of the surface depicted by each pixel in a single RGB image. This task is of interest to many fields, most notably 3D scene reconstruction, autonomous driving, and Augmented Reality. MegaDepth [162] and Dense Prediction Transformers (DPT) [237] have been selected among the multitude of MDE methods. MegaDepth proposes a large depth dataset built by using structure from motion and multi-view stereo on internet photo collections, seeking to learn to predict monocular depth with high accuracy and generalizability. It depicts different city landmarks, including buildings, statues, and squares. In addition to the dataset, the authors train different architectures obtaining the best depth estimation results with the hourglass architecture proposed by [51]. DPT proposes the use of dense vision transformers for the prediction of monocular depth, showing an improvement of up to 28% in relative performance when compared to a state-of-the-art fully-convolutional network. In addition to the architecture, the authors also propose the use of a meta-dataset to train the neural network. This dataset is composed of 10 different depth datasets, each including different scene types ranging from indoor to outdoor and even aerial views.

All of the available images of each ENRICH dataset have been used for the

evaluation of MegaDepth and DPT. Results of both methods are evaluated following the procedure defined by [236]. The images of the dataset are resized to match the input size required by each approach. The prediction is resized to the ground truth resolution using nearest-neighbor interpolation before the evaluation. Pixels belonging to the sky in the ground truth are used as a mask to exclude invalid portions from the prediction as well as the ground truth during evaluation. Since the prediction and the ground truth may differ in scale and shift, the prediction is first aligned before measuring errors; the alignment is performed in the inverse-depth space based on the least-squares criterion. Here are provided evaluation results using a depth cap suitable for each dataset as well as non-capped predictions. Such depth cap is used to include too far pixels in the exclusion mask.

The first metric used for the evaluation is the mean absolute value of the relative error $AbsRel = (1/M) \sum_{i=1}^M |z_i - z_i^*|/z_i^*$ in depth-space. M denotes the number of valid pixels (not masked), z is the predicted relative depth, and z^* is the ground truth absolute depth. The second metric is the percentage of pixels with $\delta = \max\left(\frac{z_i}{z_i^*}, \frac{z_i^*}{z_i}\right) > \theta$ in depth space. θ defines a threshold for the evaluation, and a common value is $\theta=1.25$, which considers wrong pixels only those whose difference in depth is more than 25% of the ground truth value.

The experiments used the codes and the pre-trained models provided by the authors. For MegaDepth and DPT, the best-generalization and the DPT-large models are respectively used. Examples of depth predictions as well as ground truth are visible in Figure 3.30. Tables 3.18 and 3.19 report evaluation results on the ENRICH-Statue and ENRICH-Square respectively. For both datasets, the predictions are evaluated with a 70m depth cap and with uncapped depths. The two methods were tested with portrait images either rotated or not in order to evaluate the effect of image orientation on depth estimation. For both datasets and in all the experiment configurations, the best results are obtained by DPT. Table 3.20 reports the results of evaluation on the ENRICH-Aerial dataset. On this dataset, evaluation was performed with uncapped depth ground truth. Since $\delta > 1.25$ allows an error up to 37m for the nadir cameras and 54m for the oblique cameras at the average depths, it is also reported the $\delta > 1.05$ value, which allows an error up

Table 3.18: Evaluation of the depth estimation on ENRICH-Square.

Method	Portrait as landscape				Portrait as portrait			
	depth cap 70m		no depth cap		depth cap 70m		no depth cap	
	AbsRel	$\delta > 1.25$	AbsRel	$\delta > 1.25$	AbsRel	$\delta > 1.25$	AbsRel	$\delta > 1.25$
MegaDepth	0.111	13.986%	0.211	16.918%	0.108	13.041%	0.208	16.290%
DPT-large	0.087	10.337%	0.135	13.369%	0.085	9.869%	0.134	12.975%

Table 3.19: Evaluation of the depth estimation on ENRICH-Statue.

Method	Portrait as landscape				Portrait as portrait			
	depth cap 70m		no depth cap		depth cap 70m		no depth cap	
	AbsRel	$\delta > 1.25$	AbsRel	$\delta > 1.25$	AbsRel	$\delta > 1.25$	AbsRel	$\delta > 1.25$
MegaDepth	0.577	86.278%	1.217	86.530%	0.459	71.832%	1.227	72.055%
DPT-large	0.244	40.358%	0.287	41.635%	0.146	20.133%	0.188	21.155%

to 7m for the nadir cameras and 10m for the oblique cameras.

DPT-large achieved lower errors than MegaDepth on all three datasets. Both methods achieve the best results on the ENRICH-Square dataset. This is mainly related to the scene setup that resembles some of the data used for the training of the Neural Networks. The ENRICH-Statue dataset is challenging for both methods: while MegaDepth fails to estimate the correct relative depth order of the elements in the scene, DPT-large has difficulties in correctly identifying and separating the foreground statue from the background elements. The evaluation on the ENRICH-Aerial dataset shows low errors for the $\delta > 1.25$, but this error increases significantly when the threshold is reduced to 1.05. Even in this case, DPT-large provides depth estimation with a lower error, and the result of MegaDepth on the aerial view appears flat and fails to highlight the buildings from the ground. Finally, while the depth cap of the predictions has a limited impact on the evaluation, the correct rotation of the

Table 3.20: Evaluation of the depth estimation on ENRICH-Aerial.

Method	AbsRel	$\delta > 1.25$	$\delta > 1.05$
MegaDepth	0.039	0.060%	27.708%
DPT-large	0.017	0.001%	4.824%

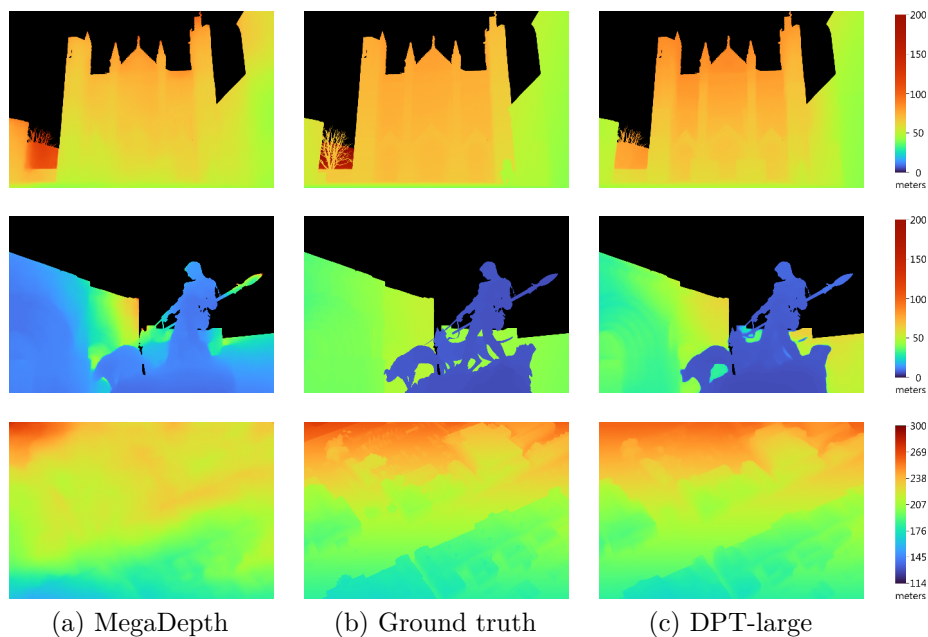


Figure 3.30: Example of predicted depth maps for the ENRICH-Square, ENRICH-Statue, and ENRICH-Aerial datasets.

portrait images significantly influences the depth estimation and evaluation. Providing the ENRICH-Statue images in the wrong orientation notably affects the results of DPT-large.

3.6.5 Impact

Mixed Reality catalogs may require the acquisition of small to large to wide objects. Experiments have shown how challenging synthetic datasets can be used to boost the testing of methods and algorithms designed for various application domains and their ability to fit a specific one. The proposed datasets feature images with different formats, cameras, environmental and acquisition conditions. Having pixel-precise ground truth information about GCP coordinates, depth maps, and 3D models allows for an accurate evaluation of various steps and aspects of 3D reconstruction pipelines, as well as numerous downstream computer vision tasks. The evaluation of SfM

pipelines showed how the scale, complexity, and texture details impact 3D reconstruction accuracy and performance. In addition, it was possible to evaluate the impact of a single step of the reconstruction pipeline (i.e. the local features) on the whole reconstruction process, thus allowing us to choose the best solution for a specific reconstruction setup. The availability of a high number of accurate GCPs provides sufficient information to also experiment with their quantity and placement. Synthetic GCPs allow quick simulation of different configurations which have a considerable impact on the quality of the 3D reconstruction. Those types of evaluations are of great importance in the creation of an MR catalog since they find use in the creation of 3D objects for rendering or 3D environment reconstruction and understanding. For this reason, the availability of multi-scale data with different geometrical and texture complexities is crucial to evaluate the performances of methods and algorithms under various conditions and gathering insights on their strengths and limits. Furthermore, MR catalogs may take advantage of the increasing availability of monocular CNN-based depth estimation methods to handle occlusion and virtual object blending with the real world. The experiment showed that, since the prediction is a relative depth (pixel ordering, not metric distances), correct depth cap and image orientation have a significant impact on the depth estimation.

4

Material Appearance Acquisition

Material appearance acquisition is the task of characterizing the optical properties of a surface and is a topic of interest in both computer vision and computer graphics. This topic is also relevant in the creation of Mixed Reality catalogs since it allows to reproduce in the virtual world elements that mimic real world-surfaces. Being able to virtually reproduce the appearance of a surface enables providing catalogs of object materials (e.g. a textile catalog presented in Section 5.2) as well as virtually replacing materials of real objects.

The appearance of a surface depends on how it absorbs, reflects, and transmits the incident light energy. These characteristics define for example evident differences between materials such as plastic, metals, and glass. But are also responsible for slight variations of appearance between surfaces belonging to the same category of materials, think for example the different colors and shininess of different plastic objects. For each material, a different amount of light is reflected in multiple directions. The distribution of this reflection is responsible for the human perception of different materials. The reflectance properties may be represented using a Bidirectional Reflectance Distribution Function (BRDF) that defines how light is reflected by a specific surface for each possible pair of the incident and reflected light directions using a set of parameters. Over the years, several algorithms for measuring the material properties of real-world surfaces have been proposed. Older methods require a large amount of data, expensive measurement devices, and long

acquisition times while newer ones are able to produce an accurate material representation using fewer data, cheaper devices, and shorter computation times.

This chapter presents the concepts and methods behind the material appearance acquisition and reviews its state of the art as well. It also presents the design and development of a portable low cost device for the acquisition of a planar surface's material appearance that can be later used to provide realistic renderings of real surfaces. This device has been successfully used for the creation of a prototype Mixed Reality catalog for textiles (see Section 5.2). The prototype has been used to validate the device and its value in acquiring material representations for use in an MR catalog.

The chapter is organized as follows. Section 4.1 introduces the concept of the Bidirectional Reflectance Distribution Function. Section 4.2 reviews the existing literature on material appearance acquisition. An in-depth review of the Photometric Stereo approach is provided in Section 4.3. Section 4.4 presents the design and methods of the proposed material appearance acquisition device. Section 4.5 presents the obtained results. Finally, Sections 4.6 and 4.7 illustrate the known limitations and impact of the device.

4.1 The Bidirectional Reflectance Distribution Function

The appearance of a material is defined by its physical properties, and in particular by the way that it reflects, absorbs, and transmits light. In the case of opaque materials, the incident light is mostly scattered by the interior in a small volume. A variable portion of that light is absorbed by the material. In this category we find wood, stone, plastic, ceramic, and fabrics. The scattering reflects the light back to the environment and makes the material appear with its diffusive color (the prominent color of the material). Some opaque materials may have a broader scattering where the light travels inside a larger volume of such materials, those are translucent materials. Other

4.1 The Bidirectional Reflectance Distribution Function

kinds of materials present different properties, such as metals where all the light is reflected (usually with high specularity) and not absorbed, transparent materials where most of the light is refracted through the material without scattering, and layered materials where a reflective surface is covered by a top clear coat.

The appearance of materials can be modeled by a function. In the case of opaque materials, this function is called the Bidirectional Reflectance Distribution Function (BRDF) [206]. The BRDF models the amount of light energy reflected from any incoming direction into any outgoing direction for any given wavelength at a specific point p of a surface. It is thus a 4D function that maps any pair of directions over the upper unit hemisphere to a non-negative real number:

$$f_r(\omega_i, \omega_o) : \Omega^+ \times \Omega^+ \longrightarrow R^+ \quad (4.1)$$

where f_r is the BRDF, Ω^+ is the solid angle representing the upper unit hemisphere, and ω_i and ω_o are unit-length vectors representing the directions of incoming and reflected light respectively. The concept of BRDF was first introduced by Nicodemus [297] in the field of radiometry as:

$$f_r(\omega_i, \omega_o) = \frac{dL(\omega_o)}{dE(\omega_i)} \quad (4.2)$$

where $L(\omega_o)$ is the radiance along the reflected light direction, $E(\omega_i)$ is the irradiance incoming along the incident direction. The radiance is thus the amount of light energy that bounces off the surface along the direction ω_o . In general, we define radiance as the light energy traveling along a ray through space. The irradiance is instead defined as the density of light energy arriving at the surface from all directions.

The notation used in this document is geometrically illustrated in Figure 4.1. In addition to the already mentioned variables, \mathbf{n} is the normal at a specific point P on the surface, \mathbf{t} is the tangent vector perpendicular to the normal, \mathbf{h} is the half-vector.

In computer graphics the BRDF is used to compute the amount of radiance that is reflected for any pair of incoming and outgoing directions as in Equation

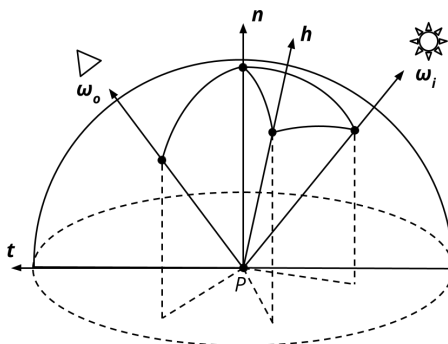


Figure 4.1: Geometry of the BRDF notation.

4.3, where $\cos \theta_i$ is used to convert the incident radiance into the incident irradiance.

$$L(\omega_o) = f_r(\omega_i, \omega_o)L(\omega_i) \cos \theta_i \quad (4.3)$$

In the field of radiometry colors are represented by wavelength. In computer graphics, instead of the wavelength, colors are usually represented by tristimulus signals (e.g. the RGB color space).

It should be noted that materials and their BRDFs can be classified into *Isotropic* and *Anisotropic*. Isotropic materials are the ones whose reflection is independent of the orientation of the surface, and thus the reflectance properties are invariant to rotations of the surface around \mathbf{n} . Anisotropic materials present instead of reflection changes w.r.t. the rotation around \mathbf{n} . This category includes brushed metals, slate, satin, wood, and hair.

It is also important to note that the BRDF can characterize only the appearance of opaque materials. For example, transparent and translucent materials require more general scattering functions and material models. Those materials are usually described by a Bidirectional Scattering Distribution Function (BSDF) or a variation of it. The BSDF is composed of two other functions: the BRDF which describes the reflectance of the material, and the BTDF (Bidirectional Transmittance Function), which describes its transmittance properties of it. This chapter considers only the BRDF and the materials that it can represent.

4.1.1 BRDF models

Observing the differences between real-world materials is easy to observe how broad the appearance of materials is and ranges from the shininess of polished surfaces to the matt appearance of other surfaces. These materials present a different kind of scattering such as mirror, glossy, diffuse, and Lambertian. Each of these types of scattering can be characterized by the BRDF and several parametric models for this characterization exists.

In computer graphics BRDF models try to represent the characteristics of the materials using a limited set of parameters. The parameters are selected to model salient characteristics of a material's appearance. These BRDF models are either *Phenomenological* or *Physically-based*. In any case, the models try to be as precise as possible in the reproduction of the material properties while being also concise and inexpensive to evaluate.

4.1.1.1 Phenomenological models

Phenomenological models are based on reflectance data, which is approximated by analytical formulas to reproduce characteristics of real-world materials. Here are briefly described some of the most important phenomenological models.

The most simple phenomenological model is the *Mirror scattering*. A mirror-link surface can be represented through the following computational model:

$$L(\omega_o) = \begin{cases} \rho L(-\omega_i), & \text{if } \omega_o = \omega_i - 2(\omega_i \cdot \mathbf{n})\mathbf{n} \text{ and } \omega_i \cdot \mathbf{n} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (4.4)$$

This represents the ideal mirror where all the light is reflected along the mirror direction of ω_i . Since in the reality, some energy is absorbed by the surface a constant ρ defines the reflectivity of the surface as a number between 0 and 1.

Another early BRDF model is the Lambertian model based on Lambert’s Law [153]. When illuminated, Lambertian surfaces present the same outgoing radiance in every reflected direction. The outgoing radiance also varies linearly with the irradiance. The BRDF is thus constant and defined as:

$$f_r(\omega_i, \omega_o) = \frac{k_d}{\pi} \quad (4.5)$$

where k_d is the parameter representing the albedo (or base color) of the material. In the real world Lambertian surfaces do not exist, however this model can be used to represent materials that present an high level of subsurface scattering, thus an approximation of matte surfaces.

Non-Lambertian surfaces have been modeled using the Phong model [228]. It is a method based on the cosine law that approximates the reflectance properties of slightly rough materials. This model captures the specular reflections and is defined as:

$$f_r(\omega_i, \omega_o) = k_s(\omega_o \cdot \omega_i)^n \quad (4.6)$$

where k_s is a specular constant that defines the magnitude and color of the reflection, and n is a scalar that controls the shape of the specular highlight. Typically the Phong model is used together with the Lambertian BRDF to model the specular and diffusive components respectively. The main limitation of this model is that it does not take into consideration energy conservation. To solve the problem of energy conservation a variation of the model known as the Blinn-Phong reflection model [36] has been proposed:

$$f_r(\omega_i, \omega_o) = k_s(\mathbf{n} \cdot \mathbf{h})^n \quad (4.7)$$

Instead of the ω_i reflection vector it uses the halfway vector \mathbf{h} and the normal direction \mathbf{n} . While being inexpensive to compute, the Phong and Blinn-Phong models lack visual fidelity. The Blinn-Phong has been used as the standard-de-facto shading model in many render engines until they were able to use more plausible physically-based models.

4.1 The Bidirectional Reflectance Distribution Function

The Ward model [311] was designed to fit measured isotropic (Equation 4.8) and anisotropic (Equation 4.9) BRDFs. It can represent a wide variety of materials combining diffuse and specular reflections, the latter is modeled through Gaussian distributions.

$$f_r(\omega_{\mathbf{i}}, \omega_{\mathbf{o}}) = \frac{k_d}{\pi} + \frac{k_s}{\sqrt{\cos \theta_i \cos \theta_o}} \cdot \frac{e^{-\tan^2 \left(\frac{\theta_h}{\alpha^2} \right)}}{4\pi\alpha^2} \quad (4.8)$$

$$f_r(\omega_{\mathbf{i}}, \omega_{\mathbf{o}}) = \frac{k_d}{\pi} + \frac{k_s}{\sqrt{\cos \theta_i \cos \theta_o}} \cdot \frac{e^{-\tan^2 \theta_h \left(\frac{\cos^2 \phi_h}{\alpha_x^2} + \frac{\sin^2 \phi_h}{\alpha_y^2} \right)}}{4\pi\alpha_x\alpha_y} \quad (4.9)$$

In the above equations θ_i , θ_o , θ_h are respectively the angles between vectors $\omega_{\mathbf{i}}$, $\omega_{\mathbf{o}}$, \mathbf{h} and the normal direction \mathbf{n} . ϕ_h is the azimuth angle of the half vector projected into the surface plane. α is the standard deviation of the surface slope. Instead, anisotropic materials use α_x , α_y to control the Gaussian lobe in the principal directions of the anisotropy. In both cases, the normalization factors ($4\pi\alpha^2$, $4\pi\alpha_x\alpha_y$) ensure that the distribution will integrate easily and predictably over the hemisphere.

The Lafortune [152] model is an empirical model based on Phong and designed to represent surface reflectance as a combination of cosine lobes. The function is defined as:

$$f_r(\omega_{\mathbf{i}}, \omega_{\mathbf{o}}) = \sum_{j=1}^N k_s^j [C_x^j \omega_{ix} \omega_{ox} + C_y^j \omega_{iy} \omega_{oy} + C_z^j \omega_{iz} \omega_{oz}]^{n_j} \quad (4.10)$$

where N is the number of lobes, where C_x^j , C_y^j , C_z^j control the magnitude and direction of the j^{th} cosine lobe. The main intuition behind this formulation is that the reflection lobes of real materials are not centered on the mirror reflection. Using this model its parameters can be adjusted to represent a wide variety of materials using a large enough number of lobes.

4.1.1.2 Physically-based models

This section describes the most relevant physically-based models. Those models are designed to follow the laws of physics and optics considering any surface as rough at the micro-scale. At this fine scale, the surface can be described as made up of microfacets, each of which acts as a perfect mirror reflector. Different materials present thus different distribution of these microfacets in terms of their size and orientation.

The concept of microfacets was first adopted by the Torrance-Sparrow [288] and later extended by the Cook-Torrance [60] model. According to the microfacet formulation, the incident light along ω_i at a point P when looking at micro-scale can be the subject of three different events as shown in Figure 4.2. It can be reflected into the environment by a microfacet, can be blocked by a close microfacet before arriving at P (shadowing), or can be blocked by a microfacet after being reflected from the facet at P (masking).

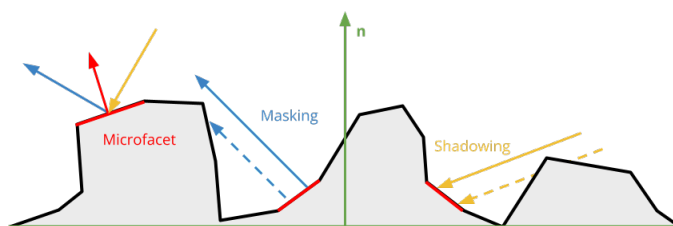


Figure 4.2: Cook-Torrance BRDF microfacets.

It thus considers that the amount of light reflected along ω_o is related to the orientation of the microfacets, their reflectance, and the amount of shadowed or masked rays. The Cook-Torrance BRDF uses Lambertian reflections for the diffusive component of the material and models the specular component as follows:

$$f_r(\omega_i, \omega_o) = \frac{FDG}{4(\mathbf{n} \cdot \omega_i)(\mathbf{n} \cdot \omega_o)} \quad (4.11)$$

where F is the Fresnel term that describes the light scattering of individual microfacets, D defines the microfacets' distribution, and G is the geometry

attenuation term that accounts for both shadowing and masking.

The Fresnel terms represent the reflection of a single microfacet as:

$$F = \frac{(g - c)^2}{2(g + c)^2} \left(1 + \frac{(c(g + c) - 1)^2}{(c(g - c) + 1)^2} \right) \quad (4.12)$$

where $c = \omega_o \cdot \mathbf{h}$ and $g = \eta^2 + c^2 - 1$ given η the index of refraction. A common approximation of the Fresnel effect is provided by the Schlick's fresnel term [257]:

$$F = \frac{(\eta - 1)^2}{(\eta + 1)^2} + \left(1 - \frac{(\eta - 1)^2}{(\eta + 1)^2} \right) (1 - \omega_o \cdot \mathbf{h}) \quad (4.13)$$

The statistical distribution of the microfacets D that controls the shape of the specular highlight is usually defined as a Gaussian as proposed by Blinn [36] or a Beckmann distribution [21]. Finally the geometry attenuation term G is defined as:

$$G = \min \left(1, \frac{2(\mathbf{n} \cdot \mathbf{h})(\mathbf{n} \cdot \omega_o)}{\omega_o \cdot \mathbf{h}}, \frac{2(\mathbf{n} \cdot \mathbf{h})(\mathbf{n} \cdot \omega_i)}{\omega_o \cdot \mathbf{h}} \right) \quad (4.14)$$

The Oren-Nayar [216] BRDF improved the use of microfacets for rough materials. The proposal is modeled by microfacets arranged in symmetrical V-cavities that act as Lambertian reflectors (instead of the Fresnel reflectors used in Cook-Torrance). This model still accounts for masking, shadowing, and inter-reflections. However, it is not suitable for glossy surfaces. The model is also heavy to compute but the authors proposed a simplification that doesn't consider inter-reflections:

$$\begin{aligned} f_r(\omega_i, \omega_o) &= \frac{k_d}{4\pi} (A + B \max(0, \cos(\phi_r - \phi_i)) \sin \alpha \tan \beta) \\ A &= 1 - 0.5 \frac{\sigma^2}{\sigma^2 + 0.33} \\ B &= 0.45 \frac{\sigma^2}{\sigma^2 + 0.09} \end{aligned} \quad (4.15)$$

where σ is the surface roughness, $\alpha = \max(\theta_o, \theta_i)$ and $\beta = \min(\theta_o, \theta_i)$.

The microfacets theory introduced by Cook-Torrance has been later further extended by Walter et al. [306]. In their work, a new microfacet distribution D function better fits measured data of real materials. Rough surfaces and transmission are also better represented. The new D function is named GGX and is defined as:

$$D = \frac{\alpha_g^2 \chi^+(\mathbf{h} \cdot \mathbf{n})}{\pi \cos^4 \theta_h (\alpha_g^2 + \tan^2 \theta_h)} \quad (4.16)$$

where α_g^2 is a width parameter, χ^+ is the positive characteristic function (which equals one if its argument is greater than 0 and zero otherwise). This distribution presents stronger tails than the Beckmann and Phong distributions and thus tends to have more shadowing. In addition to the distribution, a new bidirectional shadowing and masking term is derived from D and approximated as the product of two monodirectional shadowing terms G_1 :

$$G_1(\omega, \mathbf{h}) = \chi^+ \left(\frac{\omega \cdot \mathbf{h}}{\omega \cdot \mathbf{n}} \right) \frac{2}{1 + \sqrt{1 + \alpha_g^2 \tan^2 \theta_\omega}} \quad (4.17)$$

$$G \approx G_1(\omega_i, \mathbf{h}) G_1(\omega_o, \mathbf{h})$$

The GGX distribution is nowadays commonly used in many render engines. However, it fails to capture the highlights of extremely polished surfaces.

4.1.2 The Spatially Varying BRDF

The BRDF is meant to provide information about how light is scattered by the material at a single point on the surface. However, it can be used to represent the material properties of uniform surfaces. In the real world most objects are not made of uniform materials and light is not scattered in the same way across the surface. To solve this problem the Spatially Varying Bidirectional Reflectance Distribution Function (SVBRDF) [297] models the reflectance by adding a dependency on the surface position, making it a six-

dimensional function defined as in Equation 4.18. u and v defines a coordinate vector (u, v) that encode the position on a parameterized 2D space.

$$f_r(u, v, \omega_{\mathbf{i}}, \omega_{\mathbf{o}}) : [0, 1] \times [0, 1] \times \Omega^+ \times \Omega^+ \longrightarrow R^+ \quad (4.18)$$

Using this formulation makes clear that the SVBRDF is the representation of the unique four-dimensional BRDF at each position of the surface.

Although adding a spatial dimension enables the modeling of real-life surfaces, it still has the same limitations as the BRDF in terms of the kinds of material that it can represent. The SVBRDF is thus suitable for opaque materials that are also non-light emitters and static.

4.1.3 Texture maps

The standard approach for storing the SVBRDF parameters of a material is to use one or more images (texture maps). The process of applying the 2D texture maps onto a 3D object's surface is called *texture mapping*. Instead, the process of unrolling the 3D surface over the 2D image space is known as *UV unwrap* where (u, v) defines the texture 2D coordinate space varying in the range $[0, 1]$. It is thus necessary to also store the corresponding mapping between the 3D surface point and the 2D texture image. In the most simple case, a single texture map can be used to encode the diffusive RGB albedo of a BRDF. However, the concept of texture maps is easily usable to represent any other property of the material, such as surface orientations (normal map), surface roughness, surface smoothness, glossiness, metalness, displacement, and ambient occlusion. Texture maps are dependent on the material model used to store and render the surface. Those textures can be generated by means of automated approaches (derived by acquired data or procedurally generated) as well as created by 3D artists. The manual generation of the textures representing a material requires identifying the key characteristics of such material, and those have to be correctly encoded into the necessary texture maps. Finally, in the case of a 3D object, texture mapping should be

provided as well.

Commonly used texture maps are a diffusive albedo map, a normal map, a roughness map, and a specular albedo map. The diffusive albedo map is usually defined as an sRGB image where each pixel represents the diffusive component of the surface as an RGB color value. It should be noted that while the colors are gamma encoded, no assumption or correction is usually performed on the illuminant since this map encodes the color as diffused by the surface when lit by an ideal uniform emitter light. The normal map is again a three-channel image but its content is no longer color information but is instead the normal direction vector to the surface for each pixel. The values contained in this map may be positive or negative directions on the three axes (XYZ) and thus the 0 value is centered in the range represented by the image format (e.g. 128 for 8-bit images). Due to the surfaces being mainly flat, these maps assume a periwinkle-like color when viewed as images. It has to be noted that two main encoding conventions exist. The OpenGL and the DirectX formats whose difference is the opposite orientation of the green (or Y) channel. The roughness map is instead a one-channel texture in which the value of each pixel defines how rough a surface is at its micro-geometry level (range from 0=smooth to 1=rough). While this roughness is a linear value its effect on the surfaces is not perceived as linear by the human eye. To solve this problem render engines use the so-called perceptual roughness which maps to the linear roughness as $\sigma = \textit{perceptualRoughness}^2$ and provides a more linear perception to the users. In some cases instead of a roughness texture map a smoothness map is used and is defined as the reverse of the roughness (0=rough, 1=smooth). The specular map instead provides information regarding the specularity of the surface and thus helps in defining the specular highlights. It can be a one-channel texture (which defines the intensity of the highlight) or a three-channel texture (which defines the intensity and color of the highlights). Dark pixels signify that the light is mostly scattered while bright pixels mark pixels that present specular highlights (less scattering).

4.1.4 Rendering

The process of synthesis of an image starting from a 3D scene is known as rendering. Render engines are computer software that takes as input the computer representation of the 3D geometry of a scene (which may consist of a single object or a full environment), the material appearance parameters of such geometry, the illumination sources, and a virtual camera. The engine then computes the image depicting the scene according to the virtual setup.

Existing rendering engines that produce RGB images as output can be categorized as raster or global-illumination renderers. The goal of rasterization is to compute the mapping from the scene's geometry to image pixels. Various strategies can be used to assign a color to each pixel. Typically raster-based engines consider only local lighting (light that is transported directly from a source, reflected by a single surface, and directed into the camera). The radiance of each pixel can be thus obtained by approximating the rendering equation as:

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \sum_{j=1}^N L_i^j f_r(x, \omega_i^j, \omega_o)(n \cdot \omega_i^j) \quad (4.19)$$

where x is a point over the surface, L_o is the radiance from point x along the direction ω_o , L_e is the radiance emitted by the material, N is the number of light sources, L_i^j is the radiance emitted by the j^{th} light source, ω_i^j is the unit-vector pointing towards such light, and f_r is the BRDF of the material at point x . This formulation allows the raster algorithms to render images at real-time frame rates, thus making them appropriate for applications like video games and virtual reality. However, this comes at the cost of missing light effects (e.g., global illumination, multiple reflections) but modern raster engines are using techniques to approximate these affects keeping the real-time performances.

Global-illumination render engines aims to accurately simulate the way light is transported through the scene before reaching the virtual camera. They compute the radiance that travels along rays that intersect each pixel over the image. Due to the reciprocal nature of light, tracking the light

paths from the sources is equivalent to tracing them from the camera into the scene. This allows for simulating of the way the light is reflected, absorbed, transmitted, and refracted by the surfaces before reaching the light sources. The radiance received by each pixel can be thus computed by solving the rendering equation [140]:

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega^+} L_i(x, \omega_i) f_r(x, \omega_i, \omega_o) \cos \theta_i d\omega_i \quad (4.20)$$

where L_i is the incident radiance at point x along the ω_i direction. The integration is done over the space of incoming light directions, Ω^+ since we consider only opaque surfaces. Common algorithms for global illumination rendering are RayTracing, PathTracing, and Radiosity. One advantage of global illumination over rasterization is that indirect (global) lighting is automatically taken into account thanks to the algorithmic design thus providing soft shadows, reflections, ambient occlusion, and more out-of-the-box. Since the Equation 4.20 has to be evaluated for many rays for each pixel in the image the final computational cost is greater than the raster-based algorithms, thus making them non-suitable for real-time applications. However, thanks to the hardware acceleration support introduced in recent graphics cards newer engines are able to combine rasterization and global illumination algorithms. Rasterization is used to build the base color of the scene and ray-tracing or path-tracing is used to superimpose additional effects such as shadows, translucency, and ambient occlusion.

It is to be noted that sometimes global-illumination renderers are referred to as Physically-Based Rendering (PBR) engines. While this is true in the sense that they simulate physics to obtain the final image, this does not prevent rasterization to be a PBR engine. The concept of PBR means that the rendering procedure is based on the physics of light transport. Thus the PBR concept is more related to the material/shading model than the technique used to obtain the 2D image from the 3D scene.

4.2 Related work

The existing literature covers the task of material appearance acquisition for both devices and methods. Nowadays, traditional approaches such as the gonireflectometer are being integrated with newer deep learning techniques. This section describes the most relevant approaches and devices for material appearance acquisition, organizing them by the ability to recover the BRDF or the SVBRDF.

4.2.1 BRDF acquisition

A traditional device for measuring the BRDF of a material is the gonireflectometer (see Figure 4.3). Originally four-axis gonireflectometers [287, 297] allowed to capture the BRDF of a uniform material by using a point-light source and a photosensor placeable at various locations over the upper hemisphere of a planar material sample. By covering all of the possible positions for the light source and the photo-detector the device is able to sample the BRDF of the surface with accurate and dense measures at cost of extremely long acquisition times. Acquisition times can be reduced in the

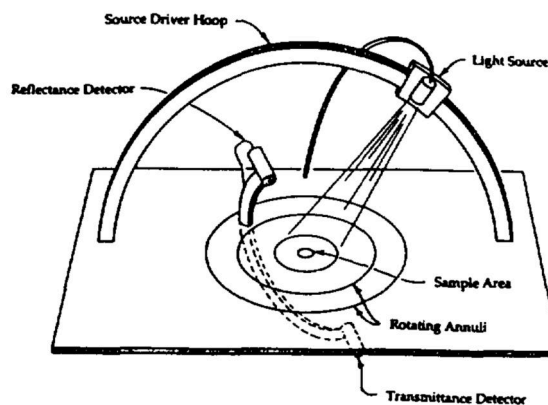


Figure 4.3: Example of a conventional gonireflectometer. The BRDF of a material is measured using a movable photosensor and light-source. Figure reproduced from [311].

case of isotropic materials which only require 3 degrees of freedom for the acquisition since the reflectance is independent of rotation about the normal.

The acquisition time was shortened by Ward [311] by using a hemispherical one-way mirror. In this way is possible to fix the camera position to simultaneously capture the reflectance in multiple outgoing positions. This makes it necessary to only move the light position. In 2014 Filip et al. [78] used a gonioreflectometer and defined data processing to acquire dense anisotropic BRDF. In 2018 Dupuy and Jakob [69] designed a goniophotometer that makes use of various light sources and photo-sensors to acquire the BRDF of isotropic and anisotropic materials. Gonioreflectometers can be constructed and calibrated to achieve high-quality measurements thus they are considered “gold standard” measurement devices for BRDF.

In addition to gonioreflectometers, other methods have been designed, such as image-based and catadioptric measurement systems.

Image-based BRDF measurement systems make use of pictures of an object taken using general-purpose equipment. The photographs capture light reflectance at various surface orientations. Marschner et al. [184] present an image-based solution for the acquisition of isotropic BRDF of a broad range of homogeneous materials. Later, Ngan et al. [203] presented a setup capable of acquiring anisotropic BRDF using two precise motors to rotate the sample and the light source. Naik et al. [201] exploited space-time images acquired by a time-of-flight camera to acquire reflectance of the sample making use of 3-bounce scattering and two known Lambertian materials.

Catadioptric optical systems make use of reflected and refracted light. The acquisition setups usually do not present moving parts making the device more robust. Relevant examples of such acquisition setups are Mukaigawa et al. [199], and Ghosh et al. [104]. Both measurement setups do not involve moving parts and use a projector as the light source, a mirror, a camera, and a beam splitter to acquire anisotropic BRDFs.

4.2.2 SVBRDF/BTF acquisition

Setups for SVBRDF or BTF acquisition can be categorized in (*hemispherical gantry, Photometric Stereo, LCD light source, flashlight and other*) based on the hardware used for acquisition.

Spherical and hemispherical gantries include the work of Rump et al. [248]. They used a hemispherical gantry with 151 cameras and used the cameras' flashes as light sources. Each camera acquires pictures for each flash (total of $151 \times 151 = 22,801$ pictures). The images are then processed to build the BTF of the subject. Ghosh et al. [103] proposed three different setups to estimate SVBRDFs of isotropic and anisotropic materials, using up to 9 polarised second-order spherical gradient illumination patterns.

Photometric Stereo is a technique that uses a similar (but simplified) hardware setup of the hemispherical gantries. This method is in-depth reviewed later in a separate Section 4.3 since it is the one used for the development of the device proposed in this chapter.

In the category of LCD light source systems, Francken et al. [82] and Aittala et al. [6] proposed similar systems based on the use of LCD display and an SLR camera. While Francken et al. recovery process is limited to the normal map, Aittala et al. are able to provide SVBRDFs of isotropic surfaces through Bayesian inference. Similarly, Wang et al. [307] use a camera and an LCD screen as an area light source to measure specular and diffuse albedos, two surface roughness parameters, and a 1D power spectrum over frequencies for visible surface bumps. Riviere et al. [244] propose a mobile reflectometry solution using a mobile device's LCD panel as the illumination source in a dimly lit room. Diffuse and specular components are separated by taking two pictures of the same sample with a differently-orientated linear polariser in front of the camera. Albedo, normals, and specular roughness are estimated using the same lighting patterns described in [103].

The last category of flash illumination setups include the work of Aittala et al. [7]. A single mobile device is used with its onboard flashlight is used to acquire a flash+no-flash image pair of textured material. A multi-stage reconstruction pipeline allows capturing of the full anisotropic SVBRDF of

surfaces with repetitive patterns. This allows us to assume that multiple points on the surface share the same reflectance properties. The input images are registered through a homography and the image is divided into sub-tiles roughly matching the size of the repeating texture pattern. A master tile is used to compute initial geometric and photometric data of the repeating pattern. Then, this data is augmented by transferring high-frequency detail from similarly lit tiles.

In recent years deep-learning based methods emerged to solve the task of material appearance acquisition, these systems usually employ flashlight setup or exploit global lighting. In 2018 Deschaintre et al. [64] proposed the use of a deep encoder-decoder convolutional neural network to recover per-pixel normal, diffuse albedo, specular albedo, and specular roughness from a single flash-lit picture of a flat surface. They also introduce the rendering loss to support the training process and evaluate the quality of the recovered SVBRDF by comparing its appearance with the ground truth by rendering both under the same lighting configuration. This takes advantage of an in-network render engine that supports inverse-rendering and backpropagation. Similarly, Li et al. [161, 326] used encoder-decoder neural networks to estimate the SVBRDF from a single picture of a planar surface without needing specular highlights generated by the flashlight. Li et al. [163] use a cascade of encoder-decoder networks to estimate the shape and material appearance from a single image. Gao et al. [87] proposed a deep inverse rendering framework that uses [64] and an arbitrary number of images to bootstrap the SVBRDF estimation that is then refined by a subsequent network so as to reintroduce fine details lost in the estimation. Deshaintre et al. [65] proposed to use multiple copies of a single image SVBRDF estimation neural network to work with multiple images, the output of each copy of the network is then processed through max pooling and a few convolutional layers to produce the final texture maps. One of the main challenges and limitations of using deep learning for this task is the necessity of a large amount of training data. While various datasets of BRDF are available [69, 77, 186], SVBRDF ones are not available in a large number of samples which is crucial for supervised training of deep-learning models. Manually generating such data is a complex and time-consuming task.

To solve this problem [64] used data augmentation over a set of procedural SVBRDFs that were sampled and rendered under multiple lighting directions. A different approach proposed by Li et al. [161, 326] used a small training set with known ground truth and a process of self-augmentation to generate additional data for training. Deshaintre et al. [65] further extended their previous approach using an online renderer for data augmentation during training.

4.3 Photometric Stereo

One of the techniques that can be used to recover a surface's properties is Photometric Stereo (PS). It is a method for recovering local surface shape and albedo from a set of images captured with the same viewpoint but under different illumination directions.

Woodham [315] was the first to introduce PS in 1980 proposing an efficient method that exploits the intensity of the acquired pixels. In fact, the intensity of each pixel in an image depends on the orientation of the corresponding surface patch (its normal), the reflectance of the material from which the surface is made of, and the direction and spectrum of the lighting. Although the reflectance properties are intrinsic to a surface, its relief produces shades that depend on the direction of the incident illumination. Changes in the illumination direction directly translate to changes in the appearance of a 3D surface. Photometric Stereo exploits this knowledge by using three identical light sources with different placements. In the ideal case of Photometric Stereo, the light sources are point lights, placed at infinite distance from the surface to be acquired, that provide uniform illumination on the whole surface. Using this illumination setup allows to assume that the light source direction is well-defined and constant across the surface for each of them. In the case of uniform Lambertian surfaces, the normal directions N can be recovered by inverting the linear equation $I = L \cdot N$, where I is a vector of m observed intensities and L is a matrix ($3 \times m$) of known incident light directions. The

linear formula can be extended to consider spatially varying albedo by taking into account the albedo reflectivity value ρ in the equation $I = \rho(L \cdot N)$. In the case of three source lights ($m = 3$) the equation can be inverted as $L^{-1}I = \rho N$. Since N represents directional vectors they are known to have unit length, thus ρ is the length of the vector ρN and N can be obtained by normalizing the direction of ρN . Please note that this formulation is suitable for grayscale images and exactly three light sources, the more generic case of RGB images and more than three light sources are described in detail in Section 4.4.3.2.

After the initial paper of Woodham [315], several works have been presented in the literature to exploit, extend, and adapt the concept of Photometric Stereo for specific applications. Most of the works take advantage of the Photometric Stereo technique to recover the 3D surface variations by using the estimated normal map to create a depth map.

In 2003 Barsky et al. [19] and later Plata et al. [230] proposed the extension of the Photometric Stereo approach to use four light sources instead of three and introduced the use of color images (Color Photometric Stereo). Previous works mainly targeted the problem of surface orientation recovery using grayscale images and were not interested in the recovery of color albedo maps. Plata et al. instead used an RGB camera, a single light source, and a turntable to acquire RGB images and recover both the surface orientations and the RGB albedo. Xie et al. [321] propose to use a PS setup with near point-light sources to acquire normal maps of 3D objects with a strong difference in surface orientations. The use of near point-lighting generates a nonlinear problem as the local surface normals are coupled with their distance from the camera as well as from the light sources. They propose a local/global mesh deformation approach to determine both, the position and the orientation of a facet simultaneously, where each facet is corresponding to a pixel in the image. Logothetis et al. [174] use a semi-calibrated near-field Photometric Stereo approach to perform the 3D reconstruction. They relax the point light source assumption by requiring only known positions and not intensities, light attenuation maps are explicitly computed. The method can jointly estimate depth, light source brightness, (scaled) albedo, light attenuation maps, and

reflectance coefficients. Liu et al. [167] present a near-light photometric stereo algorithm with circularly-placed point light sources and a perspective camera. In their work they model the acquired scene as a 3D triangulated mesh whose vertices correspond to the observed pixels. They use a two-stage process to first solve photometric stereo using the differential images captured by changing the light source position in a small amount along a circular path, and later refine the vertex positions using the original image formation model applied to the raw captured images. Their algorithm is sensitive to errors in calibration thus they propose an accurate light source positions estimation approach that uses a flat panel display. Li et al. [159] present a method to capture both 3D shape and spatially varying reflectance of isotropic materials using a multi-view photometric stereo. They combine the Structure from Motion technique to obtain precise 3D reconstruction and capture the SVBRDF by simultaneously inferring a set of basis BRDFs and their mixing weights at each surface point.

Multispectral imaging has also been used to solve the Photometric Stereo problem. Most notably, Guo et al. [109] and Zhou et al. [337] propose multispectral approaches that can recover 3D shape and albedo. Guo et al. use a monochromatic camera and assume uniform chromaticity but spatially varying albedo. Instead, Zhou et al. do not impose any prior about the surface characteristics but use a more complex hardware setup with an array of cameras arranged on concentric circles where each ring captures a specific spectrum.

All the works mentioned above use complex hardware setups that require expensive specialized hardware, such as industrial devices or laboratory prototypes. However, a few portable PS-based devices have been proposed in the literature. Gorpas et al. [106] proposed a miniature photometric stereo system, targeting the three-dimensional structural reconstruction of various fabric types. Their goal is the development of a robotic system that can navigate in unstructured environments, identify and remove textiles from piles or containers, and finally untangle and spread the textiles for some industrial process or folding. It is composed of a low-cost off-the-shelf camera, operating in macro mode, and eight light-emitting diodes. The device is a cylinder of

about 30mm in diameter and height. It is able to acquire a textile patch of size 10x10mm with a resolution of 400x400 pixels. Only grayscale images are used to acquire 3D geometry and albedo maps. Kampouris et al. [141] built a small portable device to acquire a patch of size 10x10mm of textiles to capture the microstructure of the fabrics. Their system uses an RGB camera and four LED light sources to acquire a normal map and grayscale albedo of the textiles. They used this recovered information to tackle the problem of textile classification using both handcrafted and deep-learning features using normal maps and albedo instead of plain images. Finally, Schmitt et al. [258] propose a handled device for the joint estimation of the pose, geometry, and SVBRDF. While the device is portable, it uses complex hardware design and software pipeline. The device uses a Kinect-like active depth sensor, a global shutter RGB camera, and 12 point-light sources (high-power LEDs) surrounding the camera in two circles (with radii 10 cm and 25 cm). They estimate the pose using Structure from Motion, and create a volumetric representation of the object by fusing acquired depth maps. Normals and albedo are initialized assuming Lambertian reflectance. Specular BRDF parameters are initialized as a uniform mix of base materials and later refined by optimizing (gradient-based optimization) their combination minimizing the photometric error.

Another topic of interest for the Photometric Stereo technique is the definition of the light configuration to be used for the acquisition setup. Spence and Chantler [273] derived and experimentally verified the optimal illumination setup for three-image PS of smooth and rough surface textures. They defined an overall figure of merit considering image-based rendering (i.e. relighting) of Lambertian surfaces. Then, the metric was optimized with respect to the illumination angles to find the optimal setup. As also experimentally verified the optimal separation between the tilt angles of successive illumination vectors was found to be 120° , and the optimal slant angle was found to be 90° for smooth surface textures and 55° for rough surface textures. Later, Drbohlav and Chantler [67] extended the previous work to more than three light sources. They showed that the optimal configuration for the slant angle is still with the light sources spaced equally in tilt by

$360/n$ degrees (where n is the number of lights). Instead, for the slant angle multiple optimal configurations can be used, including configurations with $n - 1$ lights with a constant slant plus one vertical light.

4.4 The proposed SVBRDF acquisition device

This section describes the proposed SVBRDF acquisition device to solve the problem of material appearance acquisition. Due to the high variety of real-world materials, the scope of the device is restricted, and some assumptions are made before describing the hardware and the software pipeline for the acquisition.

4.4.1 Scope and design choices

The goal of the proposed device is to be able to acquire the SVBRDF of a patch of a surface. Some constraints of the surface and its material are applied.

The device must be portable and usable by people who are not experts in the field. Aiming for widespread use of the device its cost should also be limited.

The materials acquired must be of a planar surface with micro surface height variations in the order of a few millimeters. While building a device for capturing the SVBRDF of generic materials and shapes is possible, it complicates the design of the hardware and makes it difficult to build a portable device. In addition to this, huge variations in the micro surface also need 3D geometry to be considered.

Some restriction on the kind of materials also applies. Only some kinds of opaque materials are taken into consideration. Transparent, translucent, as well as highly specular materials (e.g., mirrors, metals) are excluded.

The patch to be taken into consideration is of size 5x5cm; this allows

to include meaningful variations of the characteristics of the material while keeping the device size limited thus making it a portable device. This size allows also the reproduction of the materials on displays (e.g. smartphones) matching the screen size to the real size of the acquired material patch.

Among all of the techniques used for material appearance acquisitions, Photometric Stereo has been selected for different reasons. It allows for building a device that is compact and made of cheap consumer components. It is also a static device with no parts in movement that may be damaged during the transport of the device. For example, a gonioreflectometer-like or gantry setting have high hardware costs and are not suitable for a mobile device. LCD-based devices are often cheaper but they are still not suitable for building ready-to-use portable devices. Statistical-based methods heavily rely on dense pattern repetition to provide accurate acquisitions and this may not be true in the acquisition of a small patch of the surface. Deep learning-based methods are interesting since they can provide material's BRDF using RGB pictures and do not require specialized hardware, however huge computational capabilities are required for both training and inference. They also need a huge amount of data for the training process and such data is not easily collectible nor publicly available in large quantities. Furthermore, like many of the other techniques, Photometric Stereo requires some calibrations but since the device is self-contained those may be performed at production and are not required to be performed by the user.

The developed device uses the SVBRDF extension of the following BRDF. The Cook-Torrance BRDF is used as the base reflectance representation but a few changes are applied. First, Shlick's approximation of the Fresnel term is adopted. Both the microfacet distribution and the shadowing and masking terms of the Cook-Torrance model are replaced with the GGX formulation proposed by Walter et al. [306]. Refer to Section 4.1.1.2 for a description of the BRDF components used. This BRDF has been selected among the others since it fits the classes of materials that the device aims to acquire; it is also a common formulation of the BRDF adopted by many rendering engines. By providing the material appearance in the same format it is possible to render it in the engines without the need for conversion between different

representations or the definition of new shaders in the engines.

4.4.2 Hardware

The design of a new device is a challenge that comprises not only methods and software but also the necessary hardware. For the Photometric Stereo approach, some essential components are needed: a digital camera, a variable number of light sources, and a setup that allows acquisition without the interference of external light sources. Possible approaches to build the hardware required are: use consumer-ready devices, build a new device completely from scratch, or integrate consumer electronic components into a usable device.

While using consumer-grade devices is a possibility, to the best of our knowledge no cheap commercial photometric stereo setup exists. As shown for example by Google for the task of depth estimation¹, it is possible to 3D print support for multiple smartphones and uses some software to synchronize acquisitions. This solution requires the use of multiple smartphones making the cost of the device too high and their placement not easy limiting also the portability of the whole assembly.

On the opposite, designing a fully new hardware allows an extreme level of personalization making it possible to fulfill all of the system requirements except the cheapness. This solution would allow us to choose the best suitable electronic components and design a case to hold the hardware in place. Sadly, it requires also designing and producing the Printed Circuit Board (PCB) which is a complex task that requires expertise and it is not economic.

The solution adopted for the design and development of the hardware device is in-between the aforementioned possibilities. The device builds on the most suitable cheap electronics available on the consumer market and uses a custom case to house all of the components. Following this reasoning, the device uses the Raspberry Pi Camera Module V2, SK6812 LEDs, and the Raspberry Pi Zero W. The complete bill of material and the price of each

¹<https://ai.googleblog.com/2018/11/learning-to-predict-depth-on-pixel-3.html>

component is reported in Table 4.1.

Table 4.1: Bill of material.

Component	Price [€]
Raspberry Pi Zero W	10.61
Raspberry Pi Camera Module V2	28.37
LED ring 8x SK6812 CW	1.52
LED ring 16x SK6812 CW	3.04
Camera cable	4.98
LEDs connector	0.32
LEDs flat cable	0.96
SD card 8GB	5.03
3D printed case	26.00
<i>Total:</i>	<i>80.83</i>

Digital camera module — Among all the available camera modules the *Raspberry Pi Camera Module V2*² has been selected. It is a camera sensor module that comes in the format of a small (25x24mm) breakout board mounted with a Sony IMX219 [272] CMOS image sensor. This mobile format sensor (3.68x2.76mm, pixel size 1.12 μ m) provides 8MP resolution color images (3280x2464px) and is equipped with a small adjustable lens with a focal length of 3.04mm (62.2° horizontal Field of View). The choice of this specific module has been based on three different factors. First, the camera satisfies the image acquisition requirements; it can acquire an area of size 82x61mm with a spatial resolution of 40px/mm when placed 68mm over the target surface. Second, the camera module is provided with libraries that allow the acquisition of images using a pre-built camera pipeline as well as RAW-10 images. The possibility of acquiring RAW images is essential to build a custom camera pipeline for the device. Finally, the module is cheap (market price of \$25) and is easily interfaceable with the Raspberry Pi Single Board Computers (SBCs) as well as third-party SBCs thanks to its Camera Serial Interface (CSI).

²<https://www.raspberrypi.com/documentation/accessories/camera.html>

Light sources — For the device to work correctly it is essential to be able to illuminate the surface with different light source placements. While several light source types are available on the market, the Light Emitting Diode (LED) is known for its efficiency and the large number of variations (shape, power, wavelength) that are produced. The most important factors for the choice of LEDs are the wavelength of the emitted light, ease of wiring, mechanical assembly, power consumption, and emission.

The selected light sources are thus pre-built rings mounting SK6812 RGBW LEDs [266]. The SK6812 LED is a 5050 LED chip (5x5mm) that comes in a few different variations, the most suitable for this application is the RGB + Cold White. The cold white variation has been chosen since it provides a white color temperature of approximately 6500K which is equal to the CIE standard illuminant D65 which resembles the daylight illuminant. The white channel emits 6 ± 1 lumens and the light beam angle is 120° . This LED also offers some advantages compared to other ones in terms of light control and wiring. The wiring diagram consists of a 5V power supply to be provided to each LED and a single wire control bus that daisy chains the LEDs. Since the LEDs are already provided and assembled on PCB rings, it is only necessary to wire the power supply and the control bus to the first LED of each ring. Furthermore, cheap pre-assembled LED rings are available in diameters from 32mm (8 LEDs) up to 112mm (32 LEDs). Two rings have been used in the device, with 8 and 16 LEDs respectively for a total of 24 light sources. The choice has been based on mechanical constraints, the rings should not interfere with the camera field of view, and each LED must provide enough energy to each point in the area portrayed by the camera. Moreover, the use of a high number of light sources helps handling noise and shadows in the acquired images. Exact LEDs placement is described later in this section.

Controller board — The last electronic component needed is a controller board that is in charge of controlling the image acquisition process. Two different approaches can be employed. The first one is to use this controller board only for the acquisition and delegate the processing to software running on a computer. The second approach involves using the controller not only

for acquisition but also for processing the images. The second case requires more computational capabilities. Since a computer is in any case needed to recover the texture maps, the device employs the first solution to also keep the hardware of the device limited in size, power consumption, and cost. Based on this reasoning and the hardware already selected for the camera module and the light sources, the *Raspberry Pi Zero W*³ is the designated SBC to control the images acquisition process. It features a 1GHz single-core CPU, 512MB of RAM, a CSI camera interface, a wireless network adapter, and a General-Purpose Input/Output (GPIO) interface for controlling the LEDs. Power for the device is provided through the Raspberry micro-USB power port. Since the power consumption of the LEDs does not exceed the recommended maximum of 1A for the 5V rail an external voltage regulator is not needed. A single GPIO pin is used to drive all of the LEDs, refer to the SK6812 datasheet [266] for details about the control protocol. Communication and data transfer between the device and the computer employs a Wi-Fi connection.

Device case — Finally, a 3D-printed plastic case was designed to keep the hardware in place and block the light coming from external sources providing thus a dark room for the material appearance acquisition. Figure 4.4 shows Computer-Aided Design (CAD) sketches and the final device. Since the case was 3D printed using PLA which is highly specular, the inner surface of the device has been tinted using a black matte paint to limit the effects of reflections on the walls of the case in the acquired images.

The exact placement of the camera and LED rings has been determined to guarantee the acquisition of a usable material patch of size 5x5cm where enough light is received by each point of the surface for each LED. The camera is thus placed 68mm above the surface. The two LED rings are instead placed as follows: the smaller one, which has a radius of 13mm (center of the ring to center of the LED), is located 45mm above the surface generating a 75° angle with the surface. The larger one (33mm radius) is instead positioned 40mm above the surface producing an incident light angle of 51°. Both rings are

³<https://www.raspberrypi.com/products/raspberry-pi-zero-w/>

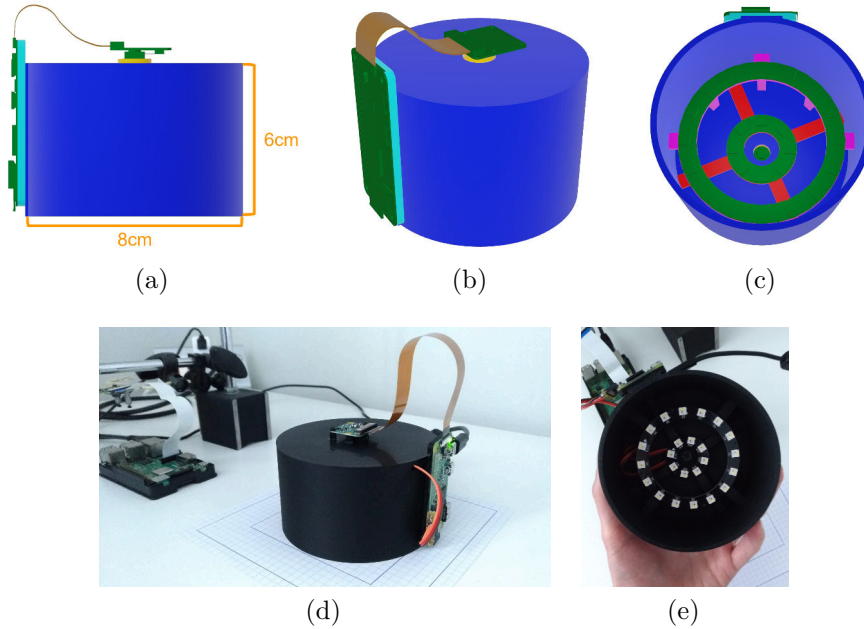


Figure 4.4: CAD drawings of the device (a-c) and the assembled device (d-e).

aligned to have their center corresponding to the main view axis of the camera.

4.4.3 Software

The software for the material appearance acquisition is composed of two main modules: a firmware that runs on the Raspberry Pi Zero for the LED control and image acquisition and an elaboration pipeline that runs on a personal computer.

4.4.3.1 Onboard firmware

Due to the limited computational power and RAM, the goal of the onboard firmware is to coordinate the image acquisition process and transfer the data to the personal computer where the processing takes place. The onboard firmware accepts commands from the main elaboration pipeline and mainly

controls the device's LEDs and camera. This software runs on the Raspberry Pi Zero, and the commands and data are exchanged through an HTTP API. Main functionalities are individual LED control, preview image acquisition, RAW image acquisition, and device power cycle management. Regarding the LED control it is possible to set the color and brightness of each LED independently. The preview image endpoint uses the built-in camera pipeline to rapidly acquire images whose quality and chromaticity fidelity is not important since those are used only for checking the alignment of the device over the patch of material to be acquired. Using the integrated pipeline it is possible to take advantage of the GPU-accelerated camera pipeline. The RAW image acquisition endpoint allows the acquisition of a single image RAW data stream specifying camera parameters such as shutter speed and analog gain. Given that the specified parameters may be tuned to the camera firmware, the actual parameters are returned in addition to the RAW data stream. The software uses Python and a set of libraries. The HTTP API is implemented using the Flask framework. The Raspberry Camera module is controlled through a customized version of the Picamera library⁴. The LEDs are driven using the Adafruit CircuitPython NeoPixel library⁵. The image transmission uses binary streams, while supplementary data is transferred using JavaScript Object Notation (JSON) documents.

4.4.3.2 Main software and processing pipeline

The software pipeline is composed of 3 main blocks: the camera pipeline, the preprocessing, and the photometric stereo processing. The camera pipeline takes as input the RAW Bayer pattern data from the camera sensor and converts this information into an RGB image. A total of 24 images are acquired for each sample, one image for each LED. The preprocessing applies a series of operations (see Figure 4.5) to ensure consistency between the acquired images: led shading correction and global brightness adjustment

⁴<https://github.com/waveform80/picamera>

⁵https://github.com/adafruit/Adafruit_CircuitPython_NeoPixel

are the most important. Finally, Photometric Stereo estimates the material appearance from the images and produces the normal map, albedo map, and roughness map.

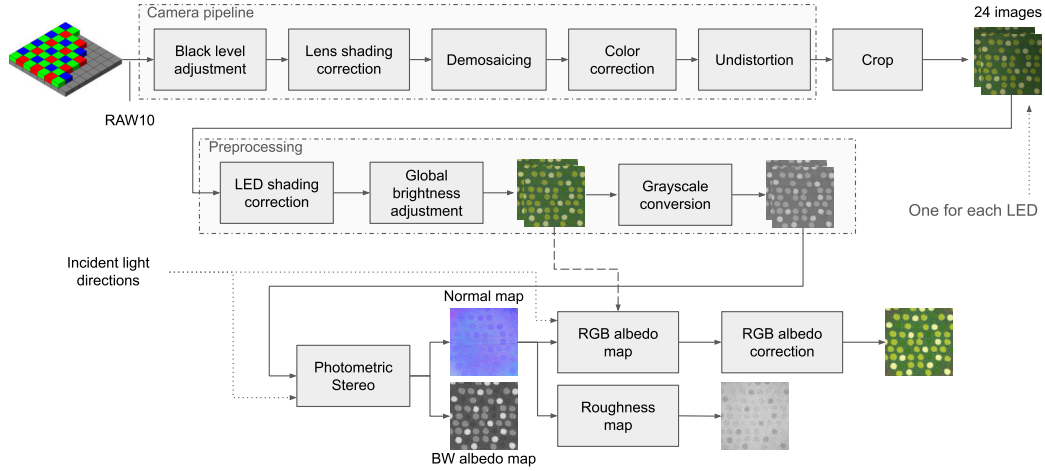


Figure 4.5: Main software pipeline.

In more detail, the camera pipeline applies the steps of black level adjustment, lens shading correction, demosaicing, color correction, and lens undistortion. Some of these steps require calibration, details about it can be found in Section 4.4.4. Black level adjustment is performed according to the camera sensor datasheet by subtracting value 64 from the RAW pixel values. Lens shading correction is applied using a pixel-wise multiplicative calibrated correction matrix. The demosaicing step uses the weighted average demosaicing approach to convert the RAW image into an RGB image. Color correction is then applied to account for the sensor’s non-linearity ($a \cdot RGB^2 + b \cdot RGB + c$), RGB correction ($M \cdot RGB$, where M is a 3×3 linear color correction matrix), and gamma correction (RGB^γ). Next, the lens radial distortion is corrected. Finally, the image is cropped to a square matching the 5×5 centimeter patch of the acquired tile (approximately 2048×2048 pixels). The camera pipeline is used to acquire 24 images, using the same camera acquisition setup but a different LED for each of them.

Some image pre-processing is then applied to ensure consistency between

the 24 images. Led shading correction takes into account the non-uniform light energy across the surface. Because the distance from the LED differs for each point on the surface and the emission lobe of the LED is not a perfect hemisphere, the portion of the surface more distant is usually less illuminated than the part directly under the light source. This step corrects the image to present the same brightness across the whole surface. The correction is done using a multiplicative matrix, similar to the lens shading correction. Then the global brightness of the remaining 23 images is adjusted to match the average global brightness of the first one. This takes into account tolerances in the power emitted by the different LEDs. The last pre-processing step creates the grayscale copies of the RGB images that will be used in the PS pipeline.

After pre-processing PS is applied to recover the albedo and normal maps. The equation of PS (see Section 4.3) assumes that the incoming light has the same incident direction for each point on the surface. This is usually true if the light source is a point light placed far from the surface. In this device, light sources are LEDs placed close to the acquired surface and can be approximated to quasi-point light sources. Because of this, the incident light direction is different for each point. To account for this, the shading function of the PS equation is rewritten as in Equation 4.21.

$$I(x, y) = \rho(x, y)L_iL(x, y) \cdot N(x, y) \quad (4.21)$$

The term $L(x, y)$ now uses a different incident light direction vector for each pixel in the image. Thanks to the LED shading correction and global brightness adjustment steps, it can be assumed that the incident light intensity is equal for each pixel and for each LED, thus considering $L_i = 1$. Equation 4.21 can be rewritten for each pixel as Equation 4.22.

$$\begin{aligned} I &= \rho L \cdot N = \rho N^T L \\ I &= G^T L = L^T G, \quad \text{where } G = \rho N \end{aligned} \quad (4.22)$$

Considering all of the 24 equations (one per image) it is possible to solve

the system of Equations 4.23 for G and obtain a vector for each pixel that encodes the normal direction N and the grayscale albedo ρ (see Equation 4.24).

$$\begin{cases} I_1 &= L_1^T G \\ \vdots & \\ I_{24} &= L_{24}^T G \end{cases} \quad (4.23)$$

$$N = \frac{G}{\|G\|}, \quad \rho = \|G\| \quad (4.24)$$

To obtain the color albedo the Equation 4.25 is minimized for each RGB channel separately. This is done by computing the value of Equation 4.26.

$$Q(\rho) = \sum_{i=1}^{24} (I_i - \rho L_i^T N)^2 \quad (4.25)$$

$$\begin{aligned} \frac{dQ(\rho)}{d\rho} &= -2 \sum_{i=1}^{24} (I_i - \rho L_i^T N) L_i^T N \\ \frac{dQ(\rho)}{d\rho} = 0 &\Rightarrow \rho = \frac{\sum_{i=1}^{24} (I_i L_i^T N)}{\sum_{i=1}^{24} (L_i^T N)} \end{aligned} \quad (4.26)$$

After obtaining the normal and albedo maps it is possible to recover the roughness map. According to Malzbender et al. [180] the variance of the normal map can be used as an indicator of the surface roughness. Following their intuition, here is defined a pipeline that can provide an estimation of the roughness map from the normal map recovered through the PS.

The steps of the pipelines are visible in Figure 4.6. The first operation is the slope exaggeration (Equation 4.27), followed by normal amplification (Equation 4.28) that considers a window around each pixel of the image. Those two steps are meant to boost the variability of the normal directions in preparation for the variance computation. The variance is later computed through Equation 4.29 which considers a window around each pixel.

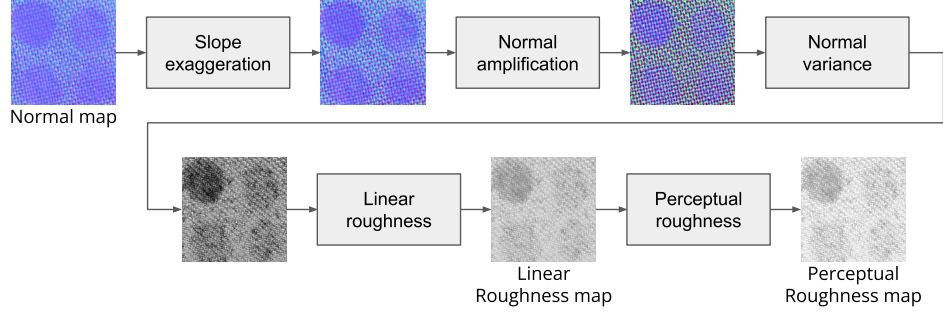


Figure 4.6: Roughness estimation pipeline.

$$n^* = \frac{n_s = (n_x, n_y, n_z/s)^T}{\|n_s\|}, \quad s = 1.5 \quad (4.27)$$

$$n^* = n + k \left(n - \frac{n_a = \sum_{j=1}^m n_j}{\|n_a\|} \right), \quad m = 23 \times 23 \text{px}, k = 2.5 \quad (4.28)$$

$$\sigma^2 = \frac{\sum_{j=1}^m ((n_{jx} - \bar{n}_x)^2 + (n_{jy} - \bar{n}_y)^2 + (n_{jz} - \bar{n}_z)^2)}{m}, \quad m = 23 \times 23 \text{px} \quad (4.29)$$

The linear roughness value is then computed using Equation 4.30 to better fill the 0-1 range.

$$r = \sigma^2 k_2, \quad k_2 = 150 \quad (4.30)$$

Finally, rendering engines usually accept as input the perceptual roughness instead of the linear one. The scope of perceptual roughness is purely to present the roughness values equally split between gloss and rough surfaces. The linear version of the roughness uses the majority of its range for surfaces that are perceptually perceived as rough. The perceptual roughness gives instead a more linear perception of the roughness variation and is computed as the square root of the linear roughness ($r_p = \sqrt{r}$).

The software pipeline is embedded into a graphical user interface through which the user can control the device, the calibration steps, and the material

appearance acquisition. This cross-platform desktop application is implemented in Python using Tkinter, and a set of libraries for processing such as NumPy, OpenCV, and PyTorch. Figure 4.7 shows the main screen of the application. The execution time for material acquisition is 6m00s on average on an Intel Core i7 6700k processor. In detail, image acquisition and data transfer from the device to the PC takes 2m35s, and pipeline execution the remaining 3m25s.

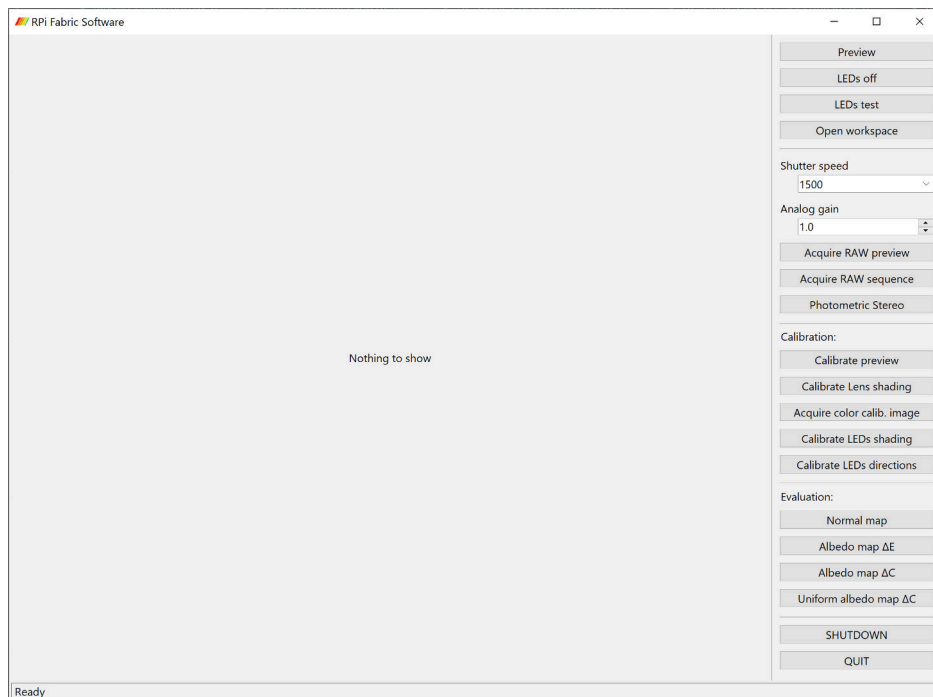


Figure 4.7: User interface of the main processing pipeline.

4.4.4 Device calibration

A few calibration procedures are required to build the calibration data used for some of the main pipeline steps.

Lens shading correction calibration — is the process that estimates a set of parameters to correct the vignette shading introduced in the image by the lens. The correction factors have been computed by using a RAW

Bayer picture of a completely white tablet’s screen. To acquire such an image a translucent white paper sheet has been placed right above the screen since the GSD of the camera is low enough to distinguish the color filters of the LCD screen. Figure 4.8 shows an example of a RAW image before and after lens shading correction. The channels of the acquired RAW image are separated based on the *rgGb* Bayer pattern. For each channel the center 64×64 px window is used to define the desired average brightness level avg_{ch} , average blur is then used to clear noise. The correction factors are computed as $avg_{ch}/image_{ch}$.

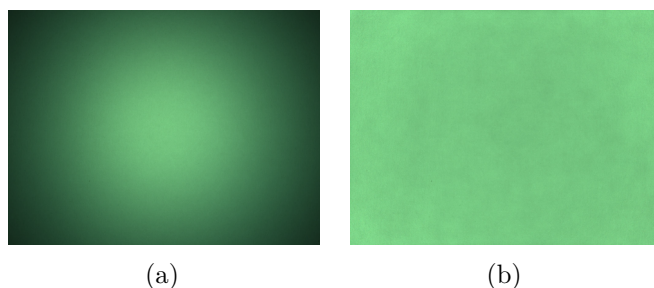


Figure 4.8: RAW image before (a) and after lens shading correction (b).

Color correction calibration — this calibration estimates the parameters a, b, c, M, γ used to correct the camera’s sensor non-linearity, RGB color, and gamma encoding. The parameters are optimized using the Nelder-Mead Simplex [202] method over color data acquired from an X-Rite ColorChecker Classic Mini. Due to the size of the color checker, each square has been acquired separately but using the same camera parameters.

Lens undistortion calibration — due to the shallow depth of field and mechanical mount of the camera the lens radial distortion parameters must be manually calibrated. Parameters for the Brown distortion model [41] have been manually estimated using a black-and-white checker board pattern.

LED shading correction calibration — similar to lens shading correction, this is the process that estimates the correction of the shading introduced in the image by the LEDs. One of the core assumptions of PS is that the

energy that lights up the material is equal for each point on the surface. When using light sources close to the surface this is not true, and light fall-off can be observed across the images. To compensate for the light shading a correction matrix has been computed for each of the 24 LEDs by acquiring an image of a uniform white paper sheet. The acquired image is then cleared of noise using average blur. Correction factors are then computed for each RGB channel separately.

Incident light direction calibration — given that the light sources are not point light placed at infinity, but quasi-point light placed near the surface to be acquired, the incident light direction depends on the LED placement and it also varies along the surface. To provide the correct light direction vector for each pixel acquired by the camera a physical calibration target with 71 mirror-like spheres has been designed (see Figure 4.9). The spheres are placed following a hexagonal pattern to allow later interpolation. The target structure has been printed with a resin-based 3D printer with resolution XY of $47\mu\text{m}$ and Z (layer height) of $10\mu\text{m}$.

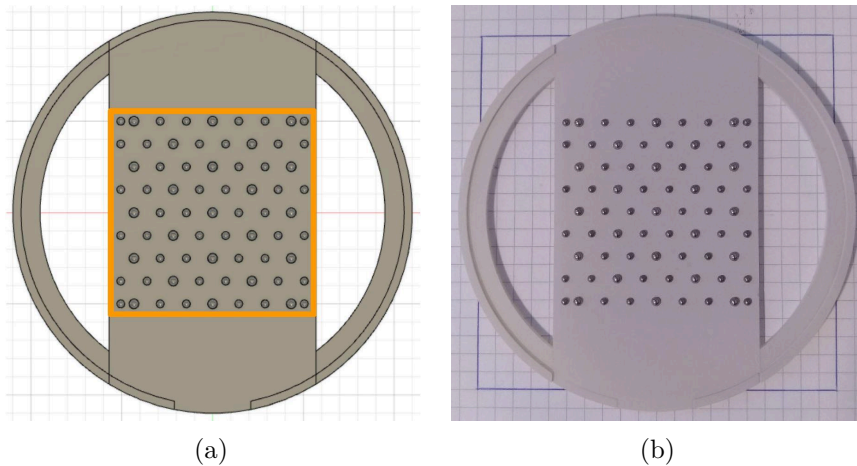


Figure 4.9: Incident light direction calibration target. (a) CAD drawing, in orange the camera field of view. (b) 3D printed target.

The calibration pipeline executes the steps visible in Figure 4.10. An image is acquired using all the LEDs to detect the sphere positions and contours.

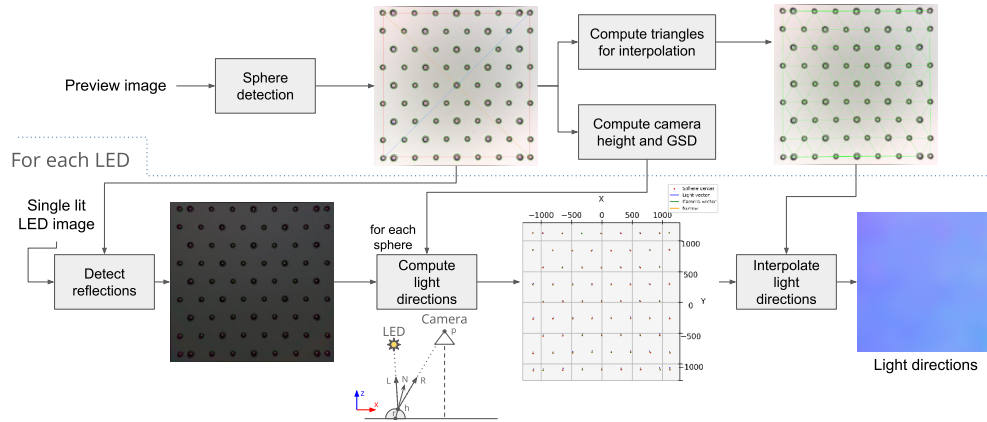


Figure 4.10: Incident light direction calibration pipeline.

The spheres’ centers are also used to determine the Delaunay tessellation later used for interpolation of the light directions, the boundary of the usable region-of-interest, and the length in pixels of the diagonals of the calibration target. Knowing the real length of the target’s diagonals and the camera’s intrinsic parameters is now possible to compute the exact height of the camera from the surface.

An image for each LED is acquired and then processed through various steps. Firstly, the reflections are detected over the spheres by finding the brightest spots. Then the light direction is computed using the following equation $L = 2(N \cdot R)N - R$, where N is the normal direction at the brightest spot on the sphere, and R is the vector of the direction reaching the camera through the focal plane. The 71 known light direction vectors are then used to build the full incident light vector map by using barycentric interpolation.

Albedo correction calibration — this calibration uses the Finlayson 2015 color correction method [79] to estimate the color correction to be applied to the estimated color albedo map. Estimation of the correction parameters is done using a GretagMacbeth ColorChecker DC color chart. This chart consists of 237 squares of paint, the 8 glossy patches are excluded and repeated patches are considered only once. Since the chart exceeds the acquisition size of the device, the tiles have been acquired and their albedo has been estimated in groups of 4 (2 by 2 patches) to speed up the process.

The same camera parameters were used for all of the acquisitions.

4.5 Experimental results

This section evaluates the obtained SVBRDF texture maps, and includes some examples of acquired materials as well. Quantitative evaluation of the normals map accuracy, as well as the albedo map in terms of color accuracy and color uniformity on rough surfaces are performed. A qualitative evaluation of the roughness map has also been performed.

A common approach to evaluate Photometric Stereo (PS) based methods is to use synthetic images for which the ground truth of the SVBRDF maps is known. This approach works well for the evaluation of the modifications to the PS pipeline but, in our case, this is not enough. In fact, in addition to PS it is necessary to evaluate the hardware assembly and its calibration, as well as the camera pipeline and the pre/post-processing steps. A procedure is therefore defined for the evaluation of the device as a whole, enabling evaluation of the various aspects involved in the material appearance acquisition.

4.5.1 Albedo map evaluation

The estimated albedo map of the material is evaluated in terms of both color accuracy and color uniformity. The first evaluates the ability of the device to provide perceptually correct albedo maps, the latter the capacity to produce correct albedo maps for uniform materials but with varied normal directions.

Color Accuracy This paragraph reports the evaluation of the color accuracy of the estimated albedo map using the GretagMacbeth ColorChecker DC color chart. This chart consists of 237 squares of paint, of which the 8 glossy patches were excluded and repeated patches are considered only once. Since the chart exceeds the acquisition size of the device, the tiles have been

acquired and their albedo has been estimated in groups of 4 (2 by 2 patches) to speed up the process. The same camera parameters were used for all of the acquisitions.

As pointed out by Barron and Malik [17] the estimation and evaluation of the albedo map presents the challenge of collimating the absolute brightness of the estimation with the one of the ground truth. The difference in global brightness is due to the acquisition setup and can be easily dealt with when performing later re-renderings; for this reason the evaluation should not consider mere differences in global brightness. In that work, they defined a scale-invariant MSE metric to evaluate the albedo. The scale invariance is enforced using a scalar α which is optimized to minimize the error, thus taking into account the ambiguity in the absolute brightness of the scene or absolute intensity of the albedo. While their proposal works for grayscale albedo, extending this concept to color albedo is not a trivial process. Following their reasoning, a scale-invariant version of the ΔE_{00}^* metric is proposed. The ΔE_{00}^* [265] has been defined by the International Commission on Illumination (CIE) as a way to measure the perceived visual difference between two colors in the CIELAB color space.

Delta E is a metric for understanding how the human eye perceives color differences. The presented modification uses a scalar α multiplied by the lightness to account for different global brightness between the estimated albedo values and the reference ones. The complete metric is reported in Equation 4.31. Since the albedo texture encodes only the base color of the surface with no information about light temperature or intensity, evaluation is performed against the ground truth colors of the ColorChecker as acquired under an ideal E illuminant which is an equal energy generator that provides a constant SPD in the visible spectrum.

$$\begin{aligned}
 si-\Delta E_{00}^* &= \frac{1}{n} \min_{\alpha} \sum \left(\sqrt{\left(\frac{\Delta L'}{k_L S_L}\right)^2 + \left(\frac{\Delta C'}{k_C S_C}\right)^2 + \left(\frac{\Delta H'}{k_H S_H}\right)^2} + R_T \frac{\Delta C'}{k_C S_C} \frac{\Delta H'}{k_H S_H} \right) \\
 \Delta L' &= L_2^* - L_1^* \alpha \quad k_L, k_C, k_H \text{ default 1} \\
 \bar{L}' &= \frac{L_1^* \alpha + L_2^*}{2} \quad \bar{C} = \frac{C_1^* + C_2^*}{2} \quad C_1^* = \sqrt{(a_1^*)^2 + (b_1^*)^2} \quad C_2^* = \sqrt{(a_2^*)^2 + (b_2^*)^2} \\
 a_1' &= a_1^* + \frac{a_1^*}{2} \left(1 - \sqrt{\frac{\bar{C}'^7}{\bar{C}'^7 + 25^7}} \right) \quad a_2' = a_2^* + \frac{a_2^*}{2} \left(1 - \sqrt{\frac{\bar{C}'^7}{\bar{C}'^7 + 25^7}} \right) \\
 C_1' &= \sqrt{a_1'^2 + b_1'^2} \quad C_2' = \sqrt{a_2'^2 + b_2'^2} \\
 \bar{C}' &= \frac{C_1' + C_2'}{2} \\
 \Delta C' &= C_2' - C_1' \\
 h_1' &= \text{atan2}(b_1', a_1') \bmod 360^\circ \quad h_2' = \text{atan2}(b_2', a_2') \bmod 360^\circ \\
 \Delta h' &= \begin{cases} h_2' - h_1' & \text{if } |h_1' - h_2'| \leq 180^\circ \\ h_2' - h_1' + 360^\circ & \text{if } |h_1' - h_2'| > 180^\circ, h_2' \leq h_1' \\ h_2' - h_1' - 360^\circ & \text{if } |h_1' - h_2'| > 180^\circ, h_2' > h_1' \end{cases} \\
 \Delta H' &= 2\sqrt{C_1' C_2'} \sin\left(\frac{\Delta h'}{2}\right) \\
 \bar{H}' &= \begin{cases} (h_1' + h_2' + 360^\circ) / 2 & \text{if } |h_1' - h_2'| > 180^\circ \\ (h_1' + h_2') / 2 & \text{if } |h_1' - h_2'| \leq 180^\circ \end{cases} \\
 T &= 1 - 0.17 \cos(\bar{H}' - 30^\circ) + 0.24 \cos(2\bar{H}') + 0.32 \cos(3\bar{H}' + 6^\circ) - 0.2 \cos(4\bar{H}' - 63^\circ) \\
 S_L &= 1 + \frac{0.015(\bar{L}' - 50)^2}{\sqrt{20 + (\bar{L}' - 50)^2}} \quad S_C = 1 + 0.045\bar{C}' \quad S_H = 1 + 0.015\bar{C}' T \\
 R_T &= -2\sqrt{\frac{\bar{C}'^7}{\bar{C}'^7 + 25^7}} \sin\left\{ 60^\circ \cdot \exp\left[-\left(\frac{\bar{H}' - 275^\circ}{25^\circ}\right)^2\right] \right\}
 \end{aligned} \tag{4.31}$$

Albedo color evaluation is performed using the $si-\Delta E_{00}^*$ following the leave-one-out cross-validation procedure on the acquired color chart data. The obtained average $si-\Delta E_{00}^*$ is equal to 2.89, which indicates a perception error that is limited but still observable by the human eye. When estimating the correction parameters using all the available patches simultaneously the error is about 2.85. Previous to the albedo color correction the average $si-\Delta E_{00}^*$ is equal to 3.62. Figure 4.11 shows the comparison between the desired ground truth albedo colors and the obtained albedo after correction. It is noticeable that the larger errors are equally distributed among the different shades.

A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K1	L1	M1	N1	O1	P1	Q1	R1	S1	T1
A2	B2	C2	D2	E2	F2	G2	H2	I2	J2	K2	L2	M2	N2	O2	P2	Q2	R2	S2	T2
A3	B3	C3	D3	E3	F3	G3	H3	I3	J3	K3	L3	M3	N3	O3	P3	Q3	R3	S3	T3
A4	B4	C4	D4	E4	F4	G4	H4	I4	J4	K4	L4	M4	N4	O4	P4	Q4	R4		T4
A5	B5	C5	D5	E5	F5	G5	H5	I5	J5	K5	L5	M5	N5	O5	P5	Q5	R5		T5
A6	B6	C6	D6	E6	F6	G6	H6	I6	J6	K6	L6	M6	N6	O6	P6	Q6	R6		T6
A7	B7	C7	D7	E7	F7	G7	H7	I7	J7	K7	L7	M7	N7	O7	P7	Q7	R7		T7
A8	B8	C8	D8	E8	F8	G8	H8	I8	J8	K8	L8	M8	N8	O8	P8	Q8	R8		T8
A9	B9	C9	D9	E9	F9	G9	H9	I9	J9	K9	L9	M9	N9	O9	P9	Q9	R9		T9
A10	B10	C10	D10	E10	F10	G10	H10	I10	J10	K10	L10	M10	N10	O10	P10	Q10	R10		T10
A11	B11	C11	D11	E11	F11	G11	H11	I11	J11	K11	L11	M11	N11	O11	P11	Q11	R11		T11
A12	B12	C12	D12	E12	F12	G12	H12	I12	J12	K12	L12	M12	N12	O12	P12	Q12	R12	S12	T12

Figure 4.11: DC ColorChecker comparison between ground truth and estimated albedo.

Color uniformity The color uniformity evaluation aims to test the ability of the device to correctly estimate the albedo as the normal direction varies. This evaluation uses the same 3D-printed target used for normal map evaluation. To perform this evaluation in the ideal conditions the tool used for evaluation should present a Lambertian surface. Since Lambertian surfaces cannot be easily created in the real world, the target uses a uniform matte gray painting which approximates the characteristics of a Lambertian surface (see Section 4.5.2 for more details about the evaluation tool). The evaluation of color uniformity of the albedo uses the ΔC_{00}^* metric. This metric is based on the ΔE_{00}^* but does not consider the lightness channel in the evaluation. Furthermore, since it was not possible to characterize the reflectance properties of the painting used and this evaluation aims to only measure the uniformity of the albedo, not its colorimetric accuracy (which is instead evaluated in the previous paragraph); the evaluation is carried out against the average color of the acquired albedo map. The evaluation presents a $\Delta C_{00}^* = 1.89$ which indicates that the device can correctly acquire the albedo of a material with a limited error even with changes in the normal directions.

4.5.2 Normal map evaluation

To evaluate the quality of the computed normal map of the surface a physical target with known normal direction variations was designed. This target was acquired using the device and its estimated normal map has been compared to the ground truth. The target for the evaluation is a square of size 30x30x2mm and presents four different shapes with a height variation of 0.25mm (see Figure 4.12). In detail, following the clockwise rotation from the top left, we have: a torus to stress the normal map at high slant angles, a flattened hemisphere to test the normal estimation at lower slant angles, a truncated cone to test constant normal angles at different height, and a grid pattern to simulate the variations of a textile. The first three shapes are also useful to test the variations of the normals for all of the possible rotations around the vertical axis. Finally, the four shapes are placed on an area lowered by 0.25mm to have their highest portion at the same height as the edges. The evaluation tool has been printed with the resin-based 3D printer already used for incident light direction calibration target. The 3D printed tool was painted with a light grey matt coat using an airbrush. The final result is visible in Figure 4.12b.

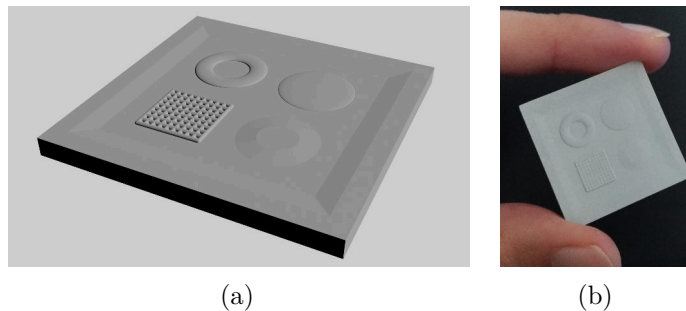


Figure 4.12: Normal map evaluation target. (a) CAD drawing. (b) 3D printed target.

The SVBRDF of the target is acquired by executing the software pipeline for a generic surface's material. Then, the obtained normal map is evaluated following the pipeline defined in Figure 4.13. First, the location of the target

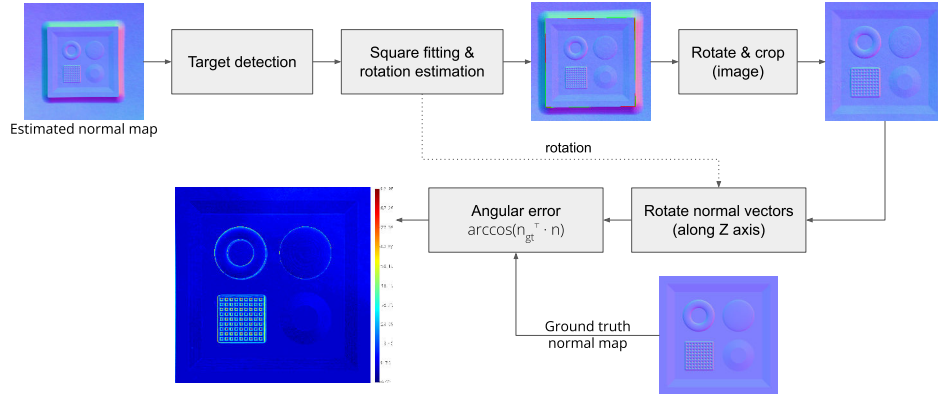


Figure 4.13: Normal map evaluation pipeline and heatmap.

is detected by creating a mask using thresholding, flood fill, erode, and dilate operations. Edges of the target are then detected using the Canny edge detector and a square is fitted on the edges using the algorithm defined in [281]. Once target detection is completed, an affine 2D transformation is estimated and applied to account for the rotation around the vertical axis of the target with respect to the camera. The corrected image is then cropped to contain only pixels belonging to the target. Since the content of the image are 3D vectors, the same rotation correction applied in the image space is also applied to the vectors w.r.t. the vertical axis Z. Finally, the angular error in degrees, defined in Equation 4.32, is computed against the ground truth normal map (\mathbf{n}_{gt}) of the target.

$$\text{Angular Error} = \arccos(\mathbf{n}_{\text{gt}}^T \cdot \mathbf{n}) \quad (4.32)$$

The result of the normal map evaluation shows a mean angular error of 9.5° , the median of 7.42° (min= 0.0° , max= 96.98° , 1st quartile= 7.27° , 3rd quartile= 10.40°).

By taking a closer look at the error heatmap in Figure 4.13 it is clear that the error is higher around the edges of the torus and the grid pattern where the slant is greater.

4.5.3 Roughness map evaluation

To the best of our knowledge, no method for quantitative evaluation of the quality of roughness texture maps of the real surface has been defined in the literature. Existing evaluation methods are thought to quantize the roughness of real surfaces for mechanical analysis purposes. While this is useful for some tasks, it is not sufficient to evaluate a roughness map since it encodes different information. Metrics such as R_a [130] measure physical properties such as the arithmetical mean roughness value of the profile deviations from the mean line of the roughness as a single real number for the whole surface. Instead, the roughness texture map describes the micro-facet distribution of each pixel separately.

The main limitation of the current process of roughness map estimation is that it uses various experimentally found values (i.e. window size and scaling factors). While fixed values are providing acceptable results adjustments to those values may be needed for some kinds of materials. In general, the pipeline tends to provide too glossy roughness but the problem is partly mitigated by the presence of the normal map. This is partly due to the map being derived for each pixel using information from an area around it. Since the roughness map encodes information that regards sub-pixel micro-surface geometry, additional spatial resolution could be used to improve the roughness estimation. Examples of roughness textures generated by the method are visible in Figure 4.14.

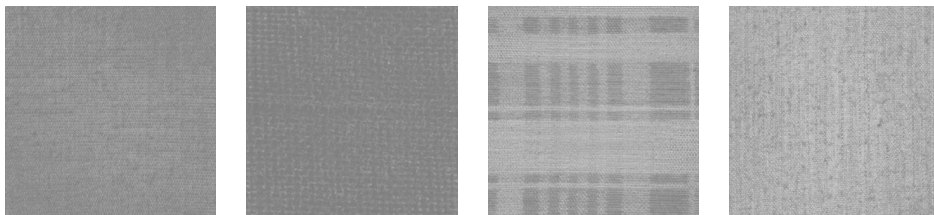


Figure 4.14: Samples of acquired roughness maps.

4.5.4 Visual results

In Figure 4.15 are provided some examples of the textures acquired by the proposed device as well as re-renderings where samples are illuminated by a single strong point light. As visible in the figure, the device can capture the fine details of the three texture maps (albedo, normals, roughness). Best results are achieved on the acquisition of surfaces that present limited specular reflections such as the samples of textile, cardboard, stone, and cork. Enough details are still captured in the plastic and wood samples which present a reflective surface. However, the printed metal sample presents some artifacts in the texture maps. In particular, it is visible how the specular reflections generated by each LED generate a normal map that represents the flat surface as a hemisphere (normals closer to the edges are slightly pointing in the direction of the edge). Being the roughness map generated from the normals it highlights the presence of noise in the acquisition and it makes visible the artifacts generated by the LEDs of the device. It is also visible in the re-rendering that the not correct roughness does not induce specular reflections on the metal sample.

It is also worth noting that the device can correctly acquire and generate texture maps for surfaces that present strong height and normal direction variations. This is mostly visible in the cork and stone samples, with the first one having height variations of about 1mm and the latter of about 4mm.

4.6 Known limitations

The proposed SVBRDF capture device presents some known limitations and open problems. The device is not suitable for the acquisition of dark-shaded surfaces. Materials with predominant dark blue or black albedo are known to be the most critical. This is due to the hardware lighting and camera setup, on such surfaces the signal-to-noise ratio (SNR) is not sufficient to clear the noise and estimate correct normal directions using the photometric

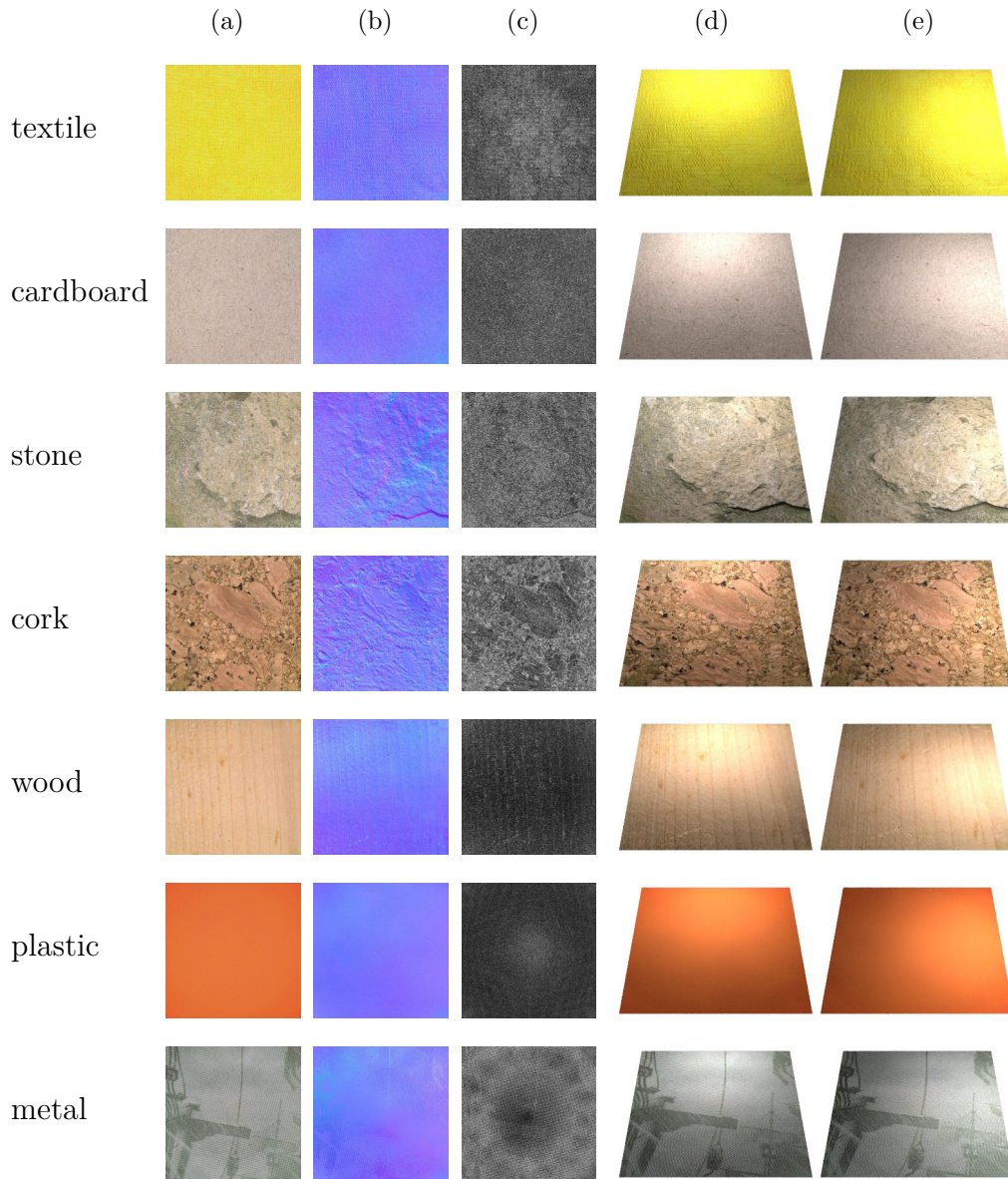


Figure 4.15: Samples of acquired maps and re-renderings. (a) albedo texture map. (b) normal texture map. (c) roughness texture map. (d,e) material re-renderings illuminated by the same point-light in two different positions.

stereo. This results in noisy normal maps and thus noisy albedo and roughness maps. A similar problem with the SNR is observable on some acquisitions in the corners of the estimated maps. This is partly due to the nonconstant

illumination provided by the LEDs across the surface patch and mainly due to the strong shading effect of the small lens equipped by the Raspberry Pi Camera module. This causes the processing steps of lens shading correction and light shading correction to heavily boost the noise together with the useful signal in the corners.

An interesting future development of the device could be the support of the acquisition of translucent, transparent, metallic, and glossy materials. This will allow us to cover a broader range of materials and also take into account combined surfaces, such as mixed opaque/glossy surfaces, printed paper, and coarse woven fabrics. Additional research could be done to improve the process of estimation of the roughness map. The main known limitation is that the map is derived for each pixel using information from a window around it. Since the roughness map encodes information that regards sub-pixel micro-surface geometry, the additional spatial resolution could be used to have more insights into the micro-surface geometry and improve the estimation. Since no method for quantitative evaluation of the quality of roughness maps of real surfaces exists, defining one could be useful for the community. Providing a mapping between physical roughness metrics and the roughness texture map could be useful to enable the evaluation.

4.7 Impact

This chapter has demonstrated the feasibility of building a low cost portable device for material appearance acquisition of various surfaces. Thanks to the Photometric Stereo technique it has been possible to design and develop a compact device able to acquire the material properties of a limited surface patch. New hardware design, software pipeline, calibration, and evaluation procedures have been defined. The device is based on consumer-grade electronics making its overall cost close to 80€. Quantitative and qualitative evaluations on acquired texture maps validated the ability of the system to build a correct spatially varying representation of materials with

Lambertian-like reflectivity. In addition, visual results showed the capacity of the device to acquire material representations of surfaces that present non-Lambertian reflectivities such as wood and some kind of plastics.

5

Rendering and Interaction

In the creation of a Mixed Reality (MR) catalog, the last important component is represented by both the visualization (rendering) of the catalog and the interaction with it by the user. This is the component in charge of the final experience that is perceived by the user. In an MR catalog, the rendering may happen in different ways due to the different levels of blending between the real and the virtual world. Similarly, the interaction may be performed using hand, visual, or voice controls.

This chapter covers these aspects by presenting a generic Mixed Reality catalog framework suitable for the creation of various MR catalogs. The framework is validated through three different case studies (Section 5.1). The first case study is about the creation of an MR catalog of textiles (Section 5.2); in such a case, the elements to be shown are single objects (i.e. the textiles), in particular, their appearance. The proposed catalog prototype uses real-time light simulation to enable experiencing the textile's appearance under different lighting conditions through a smartphone. The second use case is the virtual try-on, specifically the eyeglasses virtual try-on (Section 5.3). Despite the catalog still being about single objects (i.e. the eyeglasses), the proposed system involves 3D reconstruction and other techniques to provide a full-3D virtual try-on experience solving problems of existing applications. Finally, the feasibility of integrating the 3D virtual try-on technology into smart mirrors, which are new means of interaction that are gaining attention, is investigated through a prototype (Section 5.4).

5.1 The Mixed Reality catalog framework

With the development of advanced technologies in the last few years, Augmented Reality (AR) and Virtual Reality (VR) have emerged as key technologies in different interactive application domains. For example, they are utilized in the gaming industry [49], for human-computer interaction [10, 116], and in the manufacturing industry [38, 252].

The development of such applications has been made possible by advances in AR and VR technologies. The first time that AR technology reached the broader public was in 2016 when Niantic launched Pokémon GO, an AR-enhanced game available both to iOS and Android users. Later, the use of AR was democratized even more, after popular smartphone apps such as Snapchat, Instagram, and Facebook developed their new AR filters and, more recently, after Google launched AR objects on Google search and introduced a set of functions for face detection, 3D reconstruction, and augmented reality in the ARCore SDK [3].

Recently, AR and VR have been increasingly employed in applications designed for physical and online retail stores [37, 134]. In the retail scenario, AR and VR are technologies that a consumer interacts with and directly experiences while in the physical store, or while navigating the virtual online store. These technologies enhance the users' experience during the purchase by allowing them to interactively browse, study, or try the products before buying them [142]. This is especially true for online shopping, where the users cannot directly interact with the products they are interested in. According to Statista, "online shopping is one of the most popular online activities worldwide, and e-commerce sales in 2019 amounted to USD 3.53 trillion and are projected to grow to USD 6.54 trillion in 2022" [276]. Similar trends can be found in other countries as well. According to research by Gartner, "46 percent of retailers plan to deploy either AR or VR solutions to meet customer service experience requirements" [93].

Appealing to users with novel and engaging advanced AR and VR applications is thus a key factor in a market that is growing to such an extent. Examples of such applications are MR catalogs, virtual try-on systems, and

smart mirrors [147, 169, 221, 293]. Smart mirrors are devices designed to interact with users in a different ways. They are often based on the Internet of Things (IoT) concept and allow users to interact with applications using touch or voice controls and to display feedback and various information [33].

Although smart mirrors are general-purpose devices with a wide range of applications, here, we are more interested in MR catalog systems that are instead specifically designed to effectively improve product visualization and pre-purchase information for the retailers and to enhance the entertainment value of the shopping experience for the end-users. These include the MR catalog of products and the virtual try-on that are presented in detail in the following subsections.

A MR catalog allows a user to check the appearance and characteristics of a product. This kind of catalog can be used to provide an immersive experience along with additional details about the product the user is interested in. The design of a MR catalog system has several challenges; for example, how to present to the user the product and how he can interact with such presentation.

Regardless of the modality chosen, advanced processing software modules to support the user experience are required. Depending on the type of catalog, it may be necessary to recognize the presence of the user (e.g., using face detection or people detection algorithms), the real or virtual elements in a 3D scene may be detected, the pose of objects has to be taken into account (i.e., pose recognition), and occlusions may occur and should be dealt with (i.e., occlusion detection). Finally, the rendering of the virtual items in the real-world scene should be as seamless and plausible as possible.

There are many methods to computationally address the above-mentioned challenges. In recent years, computer vision has taken great leaps thanks to advances in artificial intelligence and machine learning. In particular, deep learning and artificial neural networks have served to boost the development of robust and fast computer vision-based algorithms, which are effectively and efficiently exploited in many application domains [231, 240].

This section describes a framework for the design of a generic MR catalog that is later adapted for two different use cases: a virtual catalog of textiles,

and eyeglasses Virtual Try-On. Different MR catalogs may require different processing modules to achieve various levels of blending between the real and virtual world as well as different types of interaction. However, the main structure of a MR catalog system is easily customizable to achieve the desired result. The framework described here is a generic solution that can be used as a blueprint to build other, extended and specialized, MR catalog applications.

5.1.1 Related works

The focus of this work are generic catalog of products and the virtual try-on technology. Therefore, the first part of this section, reviews the existing MR catalogs presenting single products. The second part reviews virtual try-on solutions, discussing their strengths and weaknesses.

MR catalogs of single products are usually available as mobile applications, and have the objective to allow a potential customer to visualize a virtual model of the product realistically. This can be achieved by placing the object in a real scene and/or by simulating the light interaction of a real environment. The development of such applications has been made possible by advances in AR and VR technologies. In addition to MR product configurators presented in Section 2.2, here some popular MR catalog solutions for single products are briefly reviewed:

- Safilo VirtualEyes [251]—this is an application that can photo-realistically render, in Augmented Reality, a vast selection of glasses on any surface. The application exploits Safilo’s 3D eyeglasses models, which have been optimized through the analysis of the ambient light in order to achieve a realistic effect.
- Ikea Place [129]—is an application developed for iOS using the ARKit that allows to virtually place furnishings in a real space. It includes true to scale 3D models of the products sold by Ikea, allows to place and rotate the models, and also allows to share pictures of the result.

Virtual try-on solutions are usually available as mobile or web applications, and have the objective to allow a potential customer to virtually try on himself some products sold by a store or manufacturer, giving him an experience similar to the one he would have in a physical store. It is mainly used to provide a virtual-wear experience of clothes and accessories. In the last few years, there has been a large increase in demand for the development of virtual try-on applications by commercial companies. The need for virtual try-on applications has further increased in the last couple of years due to the pandemic, which made it impossible for customers to participate in a physical try-on in stores. The following briefly reviews the arguably most popular virtual try-on solutions:

- Ditto’s virtual try-on [66]—this is a 3D eyeglasses and sunglasses try-on application that pays particular attention to the actual sizes. It uses a library of glasses and fits them to the estimated user’s face size. The application can also recommend the best-looking glasses for the user. For the try-on process, the user is asked to follow specific instructions. He has to record a short video while rotating the face horizontally. A credit card-sized object placed on his forehead is exploited to estimate the face size. The try-on result is shown by rendering the glasses on multiple video frames with a different face orientation.
- XL Tech Apps’s Glassify [322]—this is a virtual try-on application that works with a single frontal face image. The application requires the user’s intervention in several steps: first, the user chooses the shape that best fits his face; then, the software fits the eyeglasses on his face; finally, the user manually adjusts the position and scale of the glasses over the picture. This application works correctly only with frontal face images, and only the forepart of different glasses models can be rendered over the input image.
- Perfect Corp’s YouCam Makeup [226]—it is a virtual try-on application mainly conceived for makeup but can also be used to virtually change hair color, hairstyle, and accessories (e.g., jewelry and glasses). The

framework also includes a virtual beauty advisor, as well as tools for face detection, face tracking, and augmented reality. Some of the try-on features work in real time on the live camera video stream. The application presents some limitations in the glasses try-on: it renders only the front frame without the temples and frequently fails to properly fit the glasses when the face is not in a perfect frontal position.

- Perfect Corp’s Makeup AR [227]—it is a set of virtual try-on applications. Similarly to YouCam Makeup it includes VTO for makeup, hairstyle, hair color, lipsticks, eyeglasses, jewelry, nail polish, and more. The company makes these virtual try-on applications available as website plugins that are currently used by several sellers including MAC Cosmetics and Deborah Milano.
- MemoMi’s Memory Mirror [190]—this is an application that works differently from the previous ones. It is mainly conceived to be used in physical stores, and it requires the user to wear real eyeglasses in front of a magic mirror used to record a video of the user while trying different accessories or makeup. Each tested element can be reviewed by replaying the video, and comparison with other products is possible by displaying two videos side by side. The main limitation of this solution is the need for specific hardware and real glasses; this makes it not suitable for try-on outside of stores.
- Jeeliz [135]—this is an application for real-time web-based glasses virtual try-on. The application is available as a JavaScript widget, thus permitting integration with glasses virtual try-on in a website or a mobile web application. The application renders the 3D model of the glasses in real time on the live camera video stream. The user sees his face as in a mirror, but with glasses. There are some limitations: the tracking of the face is slow, and the glasses positioning has some delays; it uses only the front frame and the very first part of the temples, which very often penetrate the user’s face.
- Luxottica’s Virtual Mirror [177]—this solution provides eyewear try-

on in real time on a camera video feed. The user has to stand still, in a frontal position, looking downwards, for face detection and the 3D glasses model positioning on the face. The rendering follows the movements of the head and gives a digital reflection on the lens and frame for increased realism. The main limitation is that the fit of the glasses to the user's face is not automatic and can only be manually adjusted using a dedicated button.

- Voir [304]—provides a mobile application for makeup virtual try-on. This application is available for Apple iOS only and was initially conceived to work as interactive VTO stations in retail stores. Now the application provides makeup filters and also includes an AI driven makeup suggestion engine that guides the user in the process of choosing makeup products.
- Hapticmedia's Virtual Try-on [119]—they developed a watch VTO solution for Baume & Mercier. It allows the user to check the appearance of the watch on his wrist and integrates with their product configurator allowing the user to customize the watch.
- FXMirror [86]—is a magic mirror for clothes virtual try-on. It uses dedicated hardware to provide the try-on experience which can be performed on a virtual avatar or on a live video feed of the user. Interaction is performed through gestures or a tablet. Integration with a mobile application is provided to transfer the try-on session and complete customization and purchase. While the virtual avatar can be customized, it keeps a digital-appearance and does not resemble the real world. When the users uses himself as the avatar for the try-on the main limitation is the delay between the movements and the software fitting the clothes on the body.

The vast majority of the above solutions are available as standalone applications and frameworks integrable with existing services and platforms. Most of these applications offer integration with social services to allow the user to share their virtual try-on sessions and with store platforms to allow the user

to buy the products. The key features of the virtual try-on systems reviewed above are summarized in Table 5.1 for ease of comparison.

Table 5.1: Comparison of the main features of virtual try-on applications.

Applications	Input	Output	3D models	Size fitting	Markerless
Ditto [66]	video	images	✓	✓	—
Glassify [322]	image	image	—	—	✓
YouCam [226]	image	image	—	—	✓
MakeUp AR [227]	video/image	video/image	—	—	✓
Jeeliz [135]	video/image	video/image	✓	—	✓
Memory Mirror [190]	video	video	—	—	—
Virtual Mirror [177]	video	video	✓	—	✓
Voir [304]	video/image	video/image	—	—	✓
Hapticmedia [119]	video	video	✓	—	✓
FXMirror [86]	video	video/3D	✓	✓	✓

5.1.2 Workflow of a generic Mixed Reality catalog system

Figure 5.1 illustrates the workflow of a generic MR catalog system. It shows the essential components and modules of the system and their interactions. The described components and modules are intended to be generic. Different applications can require specific modules or sub-modules to operate. Some of the modules may not be required in specific use cases.

A generic virtual MR catalog system can be composed of a set of front-end modules responsible for the management of the user interaction with the system, and a set of back-end modules that implement the system logic and operational activities. Moreover, the modules and activities can be further categorized with respect to their usage. We have offline activities usually performed either at the system’s initialization or periodically to update the system. We also have real-time activities performed while the system is running and the user interacts with it. The following describes the role of the components and modules depicted in Figure 5.1.

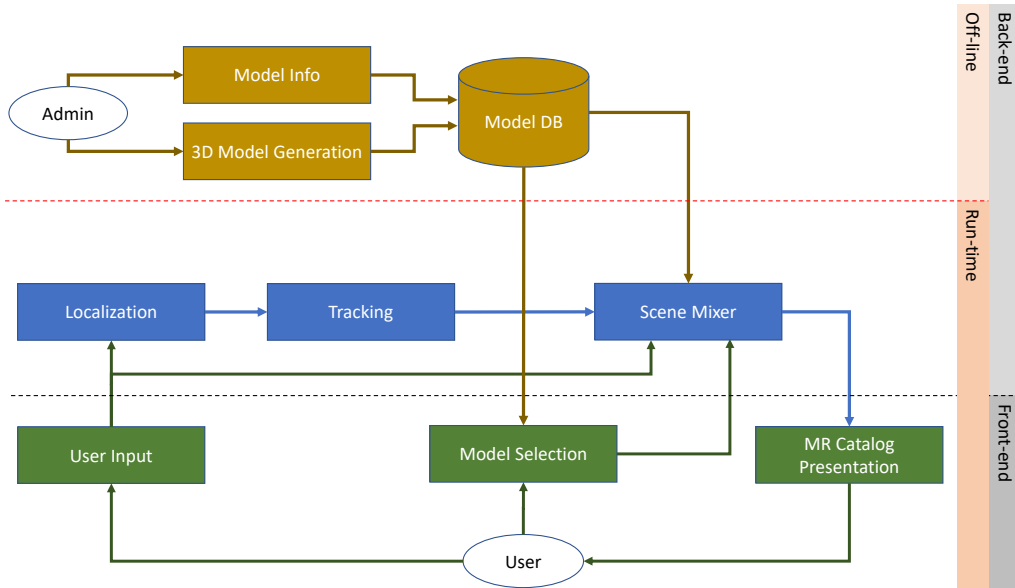


Figure 5.1: Workflow of a generic MR catalog system.

In the back-end, offline group of modules, we can find all the administrative activities usually involved in the creation of all the data required for the virtualization of the items to be displayed to the user. These activities comprise the collection of the metadata of the items (e.g., attributes, prices, descriptions) and the generation of the corresponding virtual models. These models can be of different types, but a 3D model is usually required if the items are shown in different poses. The 3D models can be directly generated from CAD files if they are available, can be created from scratch by a graphical designer, or can be acquired using 3D reconstruction techniques such as structure from motion-like approaches. If 3D reconstruction is used, the methods for synthetic data creation and pipeline evaluation defined in Chapter 3 can be used. Furthermore, if the object needs to provide an accurate material representation along with the geometry the solution proposed in Chapter 4 may be also used. These activities are periodically performed to add and/or remove items from the collection. All the generated data and information is stored in the system (e.g., in a database) and made available to the other modules that operate at run-time.

Concerning the front-end modules that operate at run-time, we have a user

input module that usually continuously acquires data from the environment and streams them to the back-end modules to be processed. Depending on the back-end processing, the module can capture RGB images (i.e., visible light) or RGBD images (visible light + depth information). The latter requires additional specialized hardware to extract depth information from the scene. In a complete virtual scenario (e.g. the virtual catalog of textiles presented in Section 5.2) 3D position and/or orientation of the user or device used may be sufficient. Usually, RGB images are sufficient for the majority of applications, as modern computer vision techniques coupled with advanced machine learning methods can extrapolate additional information from them. In some contexts (e.g., AR tourists guides, virtual tours, etc.) 3D reconstruction or depth estimation may be useful to provide insights into the scene geometry and deal with virtual object placement and occlusions. In these scenarios, the methods defined in Chapter 3 can be used to select the most suitable approach.

The front-end modules are also responsible for the management of the user experience with the system. To this end, the system has modules that encapsulate the user interface logic. For example, the user can interact with the system with a touch or touchless device to choose the items he is interested in, browse the catalog, and display the information associated with the shown items. Once the user has made his choice, the system can display it worn by the user in the virtual scene. The user can interact with it differently depending on the technology used for rendering. For example, the rendered scene can be static (e.g., a single photo), dynamic (e.g., a video stream), or interactive (e.g., a 3D scene). Each of these presentation modalities requires specific modules to operate. These modules also depend on the hardware used for presentation: this includes desktop computers, mobile devices, and head-mounted displays.

Between the offline, back-end modules and the run-time, front-end modules, we have the operational core of the system composed of the run-time back-end modules. These modules are usually organized in a pipeline. The framework indicates three main modules, but in actual applications, some modules can be merged or separated into further sub-modules. Regardless of this, a generic MR catalog application needs to identify the presence of some

elements to start the process of creating the virtual scene (e.g., plane or object localization for object presentation, and user localization for virtual try-on). This can be accomplished using advanced computer vision techniques. For example, if we are interested in placing a virtual object in a real scene, robust plane detection [92, 124, 168] and object detection [102, 295] algorithms have been developed. In contrast, in the case of virtual try-on, we may be interested in the face of the user (e.g., for virtual makeup or eyeglasses try-on), robust face detection and recognition approaches are present in the literature [137, 302, 332]. Otherwise, if we are interested in the full body (e.g., for clothing try-on), human body detection can be accomplished using techniques borrowed from the human action and pose recognition research fields [22, 170, 204]. Some of those algorithms are readily available in open-source software libraries, either exploiting standard feature-based methods or deep Learning-based ones. Examples of these libraries are OpenCV [215] and DLib [149]. Once localization is complete, the tracking module is activated. The role of the tracking module is to follow the movements of the detected elements in the scene over time. This is necessary to ensure temporal coherence in the display of the virtual scene. In fact, while the user moves in the real world, the virtual elements should be positioned and superimposed coherently in the scene. Tracking of objects and planar regions may be achieved by detection and matching to a reference or by frame-by-frame tracking aiming to minimize camera displacement between consecutive frames [223, 295]. In the case of virtual try-on, the tracking module can provide information about the user's pose in the form of facial keypoints [279] in the case of the face, or skeleton points [189] in the case of the body. This information is then passed to the scene mixer module for the generation of the virtual scene.

The scene mixer module collects information from several other modules and uses it to create the final virtual scene rendered and displayed to the user. In order to generate the virtual scene, it is necessary to blend the real one acquired by the camera. The transformed item's 3D model is superimposed onto the real scene with the user in order to create the virtual scene. The composite output of the scene mixer module can either be a static image, a recorded video sequence, or a dynamic virtual scene. In the latter case,

the system can provide a live video stream augmented with the virtual item superimposed onto the user in real-time. To this end, all the back-end processing needs to be executed as quickly as possible to cope with the user's movements. Alternatively, the system can generate a completely virtual scene allowing the user to inspect it freely and see it from different points of view without restrictions. This approach lessens the requirement for the fast-paced and real-time processing of the back-end modules.

Other challenges in the design of a robust scene mixer module are related to the realness of the virtual scene; for example, the virtual items should be rendered at the correct size. Occlusions may occur and must be dealt with. Moreover, the real and virtual parts of the scene must blend seamlessly as much as possible. Finally, the user must not be constrained in the interaction with the system.

5.2 The textiles virtual catalog use case

This section discusses the use case of a virtual catalog of textiles. This virtual catalog should be able to show the user different textiles on the screen of a smartphone and allow the user to interact with it by changing the light's intensity, color, and direction.

5.2.1 Workflow

This section describes the virtual catalog designed for textiles. The solution is designed to provide the user with an intuitive way to check the appearance of textiles under different lighting conditions while also being user-friendly, and easy to use. To achieve this, a mobile application that uses a render engine to simulate light conditions is provided. These solutions build upon the SVBRDF maps estimated by the device described in Section 4.4. By using the acquired material and a 3D rendering, the user has a plausible idea

of how the textiles will look under a given light condition. Figure 5.2 shows the workflow of the virtual catalog solution. The front-end of the system is a mobile application, while its back-end is implemented as a cloud-based web-service. The following subsections provide further details about the various components.

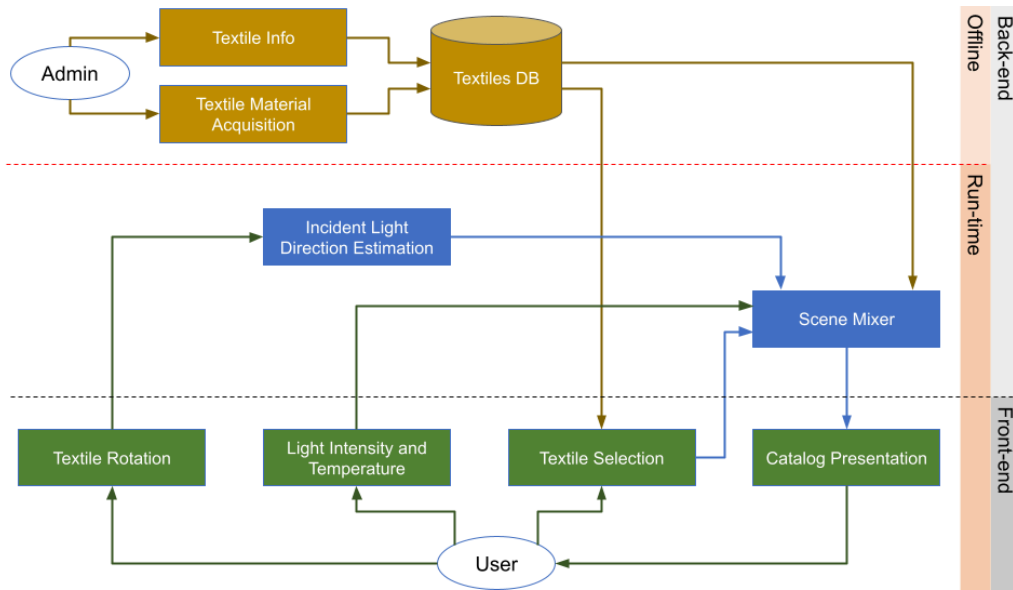


Figure 5.2: Workflow of the virtual catalog of textiles system.

5.2.1.1 Simulating the light interaction

The core functionality on which the virtual catalog builds is the ability to show plausible images of textiles under different light conditions. Simulation of different light power, color, and incident direction on a surface can be achieved using a render engine. In a render engine it is possible to set up a virtual 3D scene with the textile, a camera, and a light source as shown in Figure 5.3a. In the real world, we usually check the appearance of a textile changing its orientation w.r.t. a light source being it artificial or natural sunlight. Moreover, to replicate the behavior the user is used to, we should think of the screen of the device used to show the catalog as the fabric itself. Thus, by rotating the fabric (i.e. the device) the user should be able to change

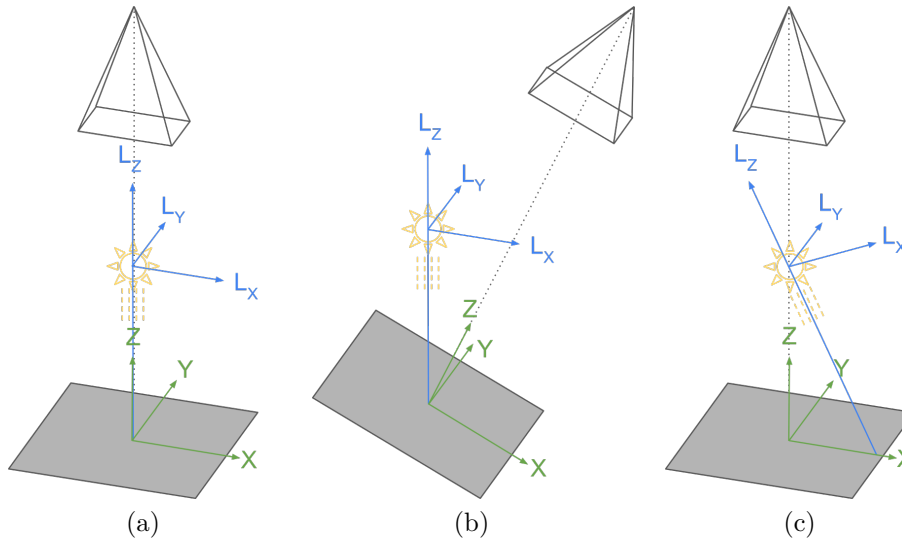


Figure 5.3: Scene setup for light interaction simulation. (a) ideal condition with view direction and incident light direction perpendicular to the sample; (b) real-world condition viewpoint and the sample are rotate while the light source remain fixed; (c) virtual setup that replicates the real-world, sample and viewpoint remain fixed while the incident light direction is rotate.

the angle of incident light rays (Figure 5.3b). This behavior can be achieved thanks to the gyroscope sensor available onboard modern smartphones and tablets. In the 3D virtual scene it makes more sense to change the position and orientation of the light source instead of moving the mesh used for rendering and the camera. Using a sunlight type of light source that provides uniform illumination across the whole scene it is not necessary to change its position but only the orientation of the light rays. This results in the setup shown in Figure 5.3c where the light is rotated at the same angle as the surface in Figure 5.3b but in the opposite direction.

Finally, the rendering of the textiles uses the Cook-Torrance-based BRDF shader model defined in Section 4.4.1. It thus uses the albedo, normal, and roughness maps provided by the acquisition device. Since the device does not generate the specular map, the value of 0.5 which provides a balance between different kinds of textiles is used for rendering.

5.2.1.2 User interface

The user interface for the virtual catalog of textiles has been built as a mobile application for smartphones and tablets. The application is in charge of textile rendering and user interaction. It also interfaces with the back-end to obtain the list of textiles along with their metadata and material data needed for rendering.

The key feature of the virtual catalog is the ability to show textiles in real-time under different light conditions; this has been achieved using the Unity [298] game engine. While this application is not a videogame, Unity presents features that make possible to handle real-time rendering, data loading, and user interaction on a mobile device in a single framework. It also makes easy to deploy the application on different mobile platforms (i.e. Android, iOS). As visible in Figure 5.4 the application presents a tile of the textile as the main content. A bottom bar with sub-panels allows the user to (from left to right): reset the textile position, customize the light's intensity and temperature, and change the textile by choosing between previews of available textiles. Using a color picker for the light color was also investigated in the preliminary designs of the application. While it allows more freedom in the choice of the illuminant, it is also difficult for the user to make slight changes to it. On the opposite, the color temperature is a familiar concept to users and allows them to adjust the illuminant color covering the majority of real-world scenarios. By default, the temperature is set to a D65 to mimic the daylight illuminant.

The application supports touch navigation, in particular, it is possible to resize the textile by using a pinch and translating it by using a single-finger swipe. Although navigation, textile selection, light intensity, and color make use of touch-screen interaction, the incident light direction is instead computed in real-time and adjusted based on the procedure described in Section 5.2.1.1. By changing the rotation of the device the user can adjust the position of the sunlight providing light to the virtual scene and thus the final rendering of the textile. To achieve this behavior the application uses gyroscope data to rotate the sunlight in the 3D scene. By default the light

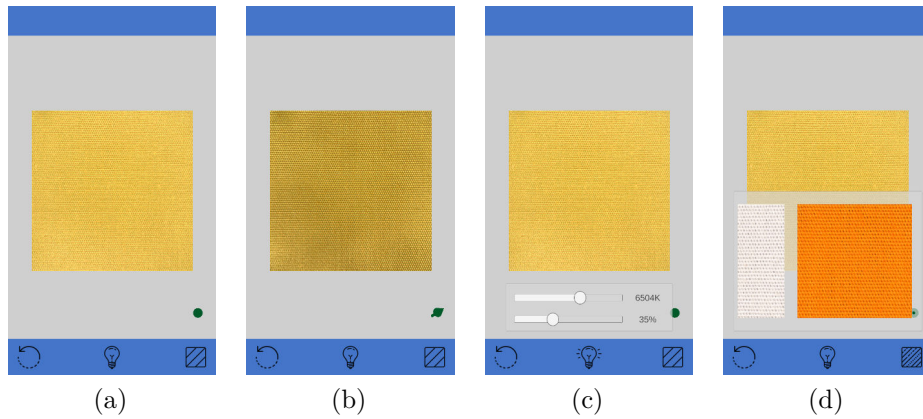


Figure 5.4: Screen captures of the virtual catalog of textiles. (a) Textile with perpendicular lighting; (b) textile with almost tangent lighting; (c) light setup panel; (d) textiles list panel.

direction is vertical above the textile disregarding the device rotation. This initial positioning acts as a 0° rotation position to account for later changes in the device rotation. A 3D arrow pointer in the lower-right corner of the screen shows the current incident light direction. The user can force a new 0 position by touching such an arrow.

The 3D scene in unity is composed of 3 main assets (see Figure 5.5): the camera, the plane used for textile rendering, and the sunlight.

By default the size of the tile on the screen is set to match the real size of the acquisition. Since the tile acquired is 5x5cm this dimension is used for

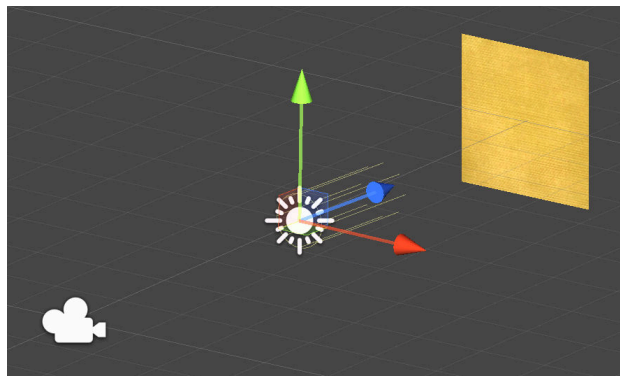


Figure 5.5: Virtual scene setup for light interaction simulation and rendering.

rendering on each device taking this into account for the screen size difference between devices. Finally, the rendering of the textiles uses the BRDF shader with the albedo, normal, and roughness maps. Textiles metadata and texture maps are loaded on the first startup from the remote web-service and a local cache copy is kept to speed up subsequent runs.

5.2.2 User evaluation

This section reports the results of the evaluation of the proposed prototype by users in terms of their experience and observations about the developed application. A standard usability test [207] with a panel of users has been used to evaluate the user experience. A total of 15 subjects of different age (25–85), expertise, and educational background were selected.

The experiment was conducted using a 5.5 inches Android smartphone where the user interacts with a touch interface and by rotating the device. The same room with controlled light has been used for all the participants. Before starting the usability test, the application and its scope were briefly described to the users. They were then observed using the application to perform a session by consulting different textiles and their appearance changes under different light orientations, intensities, and temperatures. The actual textile for each of the samples available in the app was also provided to them to allow direct comparison between the real and the virtual. No time limit was imposed. At the end of the test session, each user evaluated his experience by filling in a questionnaire. The questionnaire is based on the standard System Usability Scale (SUS) questionnaire developed by John Brooke [39]. In order to gain more insights into the application, users were asked to rate (from 1 to 5): the quality of the rendered textiles, the usefulness of the light interaction simulation, the usefulness of the light intensity control, the usefulness of the light temperature control, and their interest in using this application if made publicly available. In addition users ranked various textiles according to the fidelity of the rendering w.r.t. the real textile, the same rank for multiple textiles was not allowed. The previews of samples used are visible in Figure

5.6. Furthermore, users were asked to explain their rationale behind the ordering chosen for the textiles. Finally, some free comments from the users were also collected.

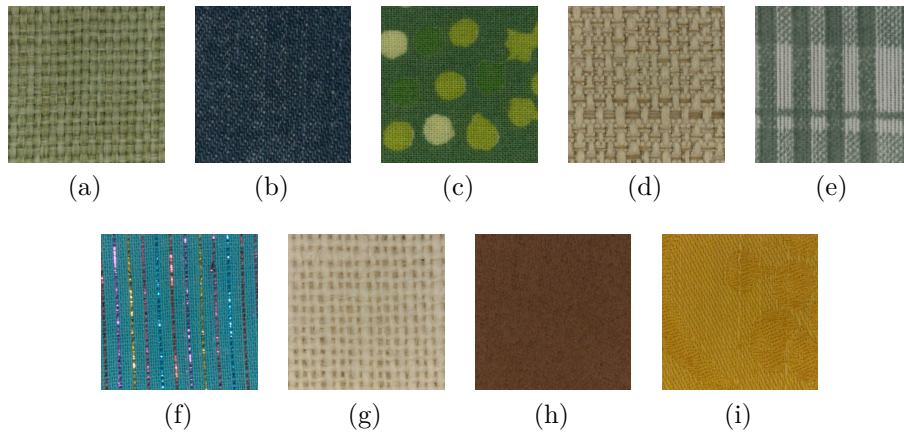


Figure 5.6: Previews of textile samples used for evaluation of the virtual catalog of textiles. (a) Green nylon; (b) Blue jeans; (c) Green cotton; (d) Beige nylon; (e) Printed nylon; (f) Sparkling cotton; (g) White jute; (h) Brown alcantara; (i) Yellow damask cotton.

Results of the SUS questionnaire are summarized in Table 5.2. The users evaluated the application very positively. It was considered easy and simple to use, without having previous knowledge or experience. By applying the standard procedure [39], the application obtained an overall SUS score of 78 out of 100, which is considered above average. The average score is set to 68 from a study on 500 systems, as described in [255]. Table 5.3 shows the scores of the five additional questions specific to the textiles application. The quality of rendered textiles was highly appreciated, with a score of 4.0 out of 5. The users found useful the main feature of the application, rating its ability to simulate light interactions with a score of 4.6. Interestingly, the controls for light temperature and intensity are perceived of slightly different usefulness, making the temperature control more important. Finally, the application was found to be engaging and the majority of users would use this kind of application if made publicly available. Moreover, some users linked their interest in the application to the possibility of using it to display

a catalog of clothes.

Table 5.2: System Usability Scale (SUS) results.

	Statement	Strongly Disagree				Strongly Agree	Avg.	SUS Score
		1	2	3	4	5		
1	I think that I would like to use this application frequently.	0	1	3	6	5	4.0	2.6
2	I found this application unnecessarily complex.	8	4	2	1	0	1.7	3.7
3	I thought this application was easy to use.	0	1	1	4	9	4.4	3.9
4	I think that I would need assistance to be able to use this application.	8	4	2	1	0	1.7	3.9
5	I found the various functions in this application were well integrated.	0	0	1	8	6	4.3	3.3
6	I thought there was too much inconsistency in this application.	12	3	0	0	0	1.2	3.9
7	I would imagine that most people would learn to use this application very quickly.	0	0	1	6	8	4.5	3.8
8	I found this application very cumbersome or awkward to use.	10	4	1	0	0	1.4	3.6
9	I felt very confident using this application.	0	1	1	5	8	4.3	3.5
10	I needed to learn a lot of things before I could get going with this application.	11	4	0	0	0	1.3	4.0

In general, the opinions of the users are positive. The majority of them appreciated the features of the application. The users also commented on the usefulness of light interaction simulation on the textiles and how this

Table 5.3: Application-specific question results.

Question	Average Rating (Range 1–5)
Quality of rendered textiles	4.0
Usefulness of light interaction simulation	4.6
Usefulness of light temperature control	4.5
Usefulness of light intensity control	4.4
Would use the application	4.2

helps in providing a more complete look and feel on a display in contrast to static pictures. Finally, the light controls (intensity and temperature) were appreciated. However, some users found the light intensity control less necessary than the temperature one. The ranking of the different textiles (Figure 5.7) shows how there is a clear opinion about which are perceived as the worse textiles and the best ones. However, there is no strong preference for textiles that place in the mid-field. In detail, the least-ranked textile samples are brown Alcantara (7.3) and sparkling cotton (8.2). Their bad placement is due to missing or poor specular reflections. The best samples are the green nylon and the yellow damask cotton (2.6) closely followed by the beige nylon (3.0). The two nylon samples (green and beige) present the same texture and material, nevertheless, the green one was consistently evaluated slightly better than the beige sample. The other samples rank close together with ranks

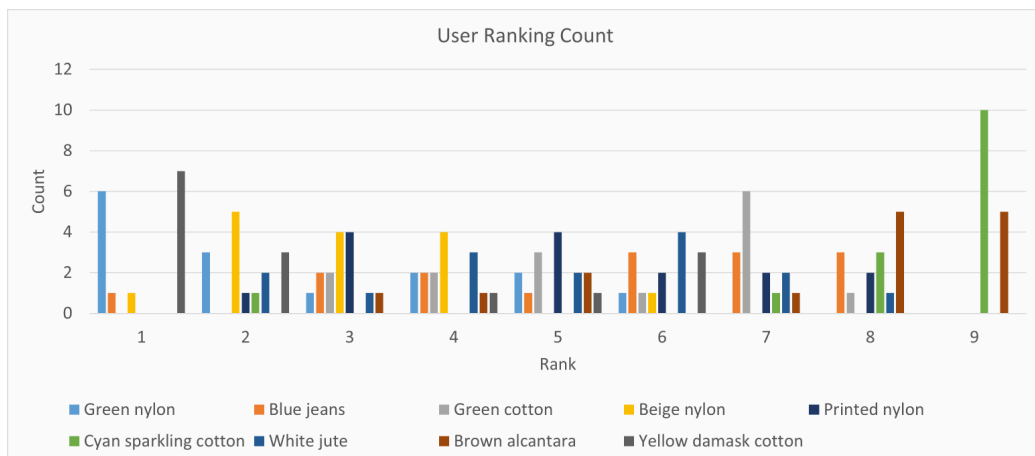


Figure 5.7: User ranking of the in-app textiles with respect to the real samples.

of 5.0 for the white jute, printed nylon 5.1, blue jeans 5.5, and green cotton 5.7. The jute sample was judged by some users to provide a plastic-looking feel. The majority of the users noticed and lamented the absence of specular reflections on the sparkling threads in the cyan textile. Interestingly, the majority of the users did not notice the drift in the color that exists in the corners of some samples, especially the darker ones. Furthermore, it is a common opinion that was easier to rank samples that provide coarse weaving and strong changes with respect to the changes in light direction. These include the damask cotton, the first two nylon samples, the jeans, and the jute.

Users also pointed out some problems with the application. The main concern was the presence of reflections on the screen of the smartphone when placed under a strong light source. In this scenario, the reflexes have a huge impact on the usefulness of the application since they prevent the correct perception of the rendered textile. To solve this problem one user suggested changing the way he interacts with the application to change the incident light direction. Instead of rotating the device, he would have preferred to keep the position and rotation of the device fixed and interact by moving his face in front of it. This may be explored as another way of interaction by tracking the pose of the face in the video feed of the frontal camera of the device. Some users instead lamented the missing possibility of manually choosing the direction of the light and suggested implementing such a feature by adding a videogame-like control on-screen. While the application wasn't perceived as unnecessarily complex, it is a common opinion that a tutorial on the features of the application at the first startup would speed up the process of learning how to use them. Furthermore, less-experienced users in photography and rendering suggested adding an explanation about the color temperature adjustment feature. Finally, a user suggested providing a larger sample of textiles on the screen and scaling it down to fit the whole space available. This is in direct contrast with other users who commented about the real-life size matching considering it crucial for the application in order to provide a precise visualization of its texture and weaving.

5.2.3 Known limitations and future work

The mobile application for the virtual catalog of textiles has been proven to be useful to allow people to remotely check the appearance of textiles without having the real fabric in their hands. However, some limitations exist and are discussed here below.

The size of the rendered tile is limited to the size of the acquired material patch. Future work could improve this by using a device that is able to acquire a larger patch of textiles. Since some textiles present a repetitive texture or pattern it will also be possible to define some strategy to take advantage of such characteristics to provide a seamless render of the textile, similar to the approach of tileable textures.

The light interaction simulation through rendering does not take into consideration external lighting. In real-world conditions, the device's screen sometimes presents strong reflections, and is thus difficult to show real-world resembling textiles in such conditions. This becomes obvious when the device is placed under direct sunlight. Another situation that makes the rendering not useful is when the user tries to check the textile at very low slant angles, in this case, the line of sight is almost parallel to the screen, and brightness and reflections have major influences on the usability. Following the suggestion of some users, it may also be useful to implement a manual control for the light source direction to solve these problems.

Brightness is also a crucial factor in the final result in other situations. Two different values of brightness play a key role: rendering brightness and screen brightness. While for the rendering one, the user can adjust the power of the illuminant in the virtual scene, the brightness of the real world is not taken into account. Many modern mobile devices are equipped with light sensors that may be used to automatically adjust the brightness of the rendering to match the real-world. Similarly, a picture taken from the device's camera may be used to estimate the light temperature in the environment and automatically match it in the application.

Furthermore, the application does not currently take into consideration loose weave textiles or fabrics that present transparencies in general. Thanks

to the general availability of a camera module that faces the opposite side of the screen on mobile devices, future development of the application may include support for transparency by superimposing the rendering to a live video stream from the camera.

5.3 The eyeglasses Virtual Try-On use case

This section discusses the use case of an eyeglasses virtual try-on catalog. This catalog should be able to virtually show the user how he or another person will appear while wearing a specific model of eyeglasses or sunglasses. The proposed virtual try-on process builds on a 3D face model reconstructed from a single input image. For this reason the section presents first the 3D face reconstruction methods available in the literature. Then, a comparison of the approaches is presented. After, the proposed virtual try-on solution is presented. Finally, the user experience and known limitations of the application are discussed.

The methods and techniques defined in Chapter 3 find here a possible use in defining the best reconstruction pipeline to generate 3D models of eyeglasses and sunglasses. Furthermore, the device for material appearance acquisition proposed in Chapter 4 can be used to provide textures for rendering such glasses. Nevertheless, the method for evaluation of 3D reconstructions defined in Section 3.3 can be used for evaluation and comparison of deep learning-based face reconstruction methods.

5.3.1 3D face reconstruction

Three-dimensional face reconstruction, as well as 3D reconstruction in general, is a long-standing problem, usually requiring specific hardware [1] or multiple images [32, 229, 245]. While the SfM-based 3D reconstruction techniques (see Chapter 3) and Photometric Stereo (see Chapter 4) may be

employed for face reconstruction of both geometry and materials, in the case of a virtual try-on application the hardware required and processing times have a great impact on the usability of the system and experience provided to the user. Since the intent is to provide the user with an easy-to-use application for virtual try-on, the system tries to simplify the face acquisition and 3D face reconstruction processes as much as possible. Therefore, the focus is on 3D face reconstruction methods that only need a single image as input, with particular emphasis on AI-based approaches, which are able to learn from data both the best reconstruction parameters and more robust prior information. The most common existing methods for 3D face reconstruction from a single image can be grouped into two main categories: *3D Morphable Model (3DMM) fitting-based* and *shape regression-based*.

The use of 3D Morphable Models (3DMM), introduced by Blanz et al. [34], represents a common solution to the 3D face reconstruction problem from a single view. Within this category, Huber et al. [127] searched for the correspondence of local features on the images and landmarks on the 3DMM. These correspondences are then used to regress the 3DMM deformation coefficients that generate a 3D face mesh similar to the one in the image. More recent methods belonging to this category, such as that of Tuan et al. [294], use Convolutional Neural Networks (CNNs) to learn to regress the 3DMM coefficients. The main advantage of the solutions exploiting 3DMM is the availability of the complete face 3D model under any circumstance since even the parts that are occluded in the input image are reconstructed using the 3DMM's geometry. On the other hand, the main limitation of these 3DMM-based solutions is that the reconstruction is often too similar to the original 3DMM model. This means that the characteristic facial traits normally defined by small details of the facial geometry are not usually well reconstructed. A recent approach based on 3DMM models, proposed by Ranjan et al. [238], claims to be able to obtain better results than previous works, especially for the reconstruction of facial expressions. To this end, they exploit the FLAME [160] 3D face base model representation. To cope with the lack of details in the reconstructed face, Extreme3D [290] introduced a bump map to model wrinkles on the Basel Face Model (BFM) [224]. The

bump map applies geometric deformations on the reconstructed 3D face, and a symmetry constraint is used to cope with occlusions. The Ganfit method proposed by Gecer et al. [94], and its fast extension [95], use a different approach and generate a realistic face shape and texture by means of Generative Adversarial Networks (GANs). The method can reconstruct very accurate models but is not available for general use. In order to better fit the generic 3DMM model to the actual face, researchers have tried to integrate different properties and constraints into their training pipelines. For example, using deep networks, Deep3DFace [63] regresses several coefficients that are used to model and fit the face: the identity of the user, expression, texture, pose, and lighting. Coupled with photometric and perceptual losses, they are able to better fit the actual user’s face. RingNet [253] is also a deep network that regresses model parameters. Specifically, it regresses the shape, pose, and expression of the FLAME model. Differently from the previous methods, RingNet exploits multiple images of the same person and an image of a different person during training to enforce shape consistency. Zhu et al. [339] proposed a Fine-Grained Reconstruction Network (FGNet) that can concentrate on shape modification by warping the network input and output to the UV space, achieving a final reconstruction that includes fine-grained geometry. Lin et al. [165] proposed to refine the initial texture generated by a 3DMM-based method with facial details from the input image. To this end, they use graph convolutional networks to reconstruct the detailed colors for the mesh vertices. The space of the parameters that need to be regressed for a 3DMM model is high-dimensional, i.e. 230 values. Guo et al. [110] propose instead to use a small subset to speed up the fitting process. Moreover, a meta-joint optimization is introduced during the training phase to improve the overall fitting accuracy. The final method is able to perform face reconstruction faster than in real-time using a MobileNet as a light backbone. The reconstructed 3D face can be used in VR applications—for example, to animate virtual avatars. However, most of the approaches are not able to reconstruct faces that can be animated in an accurate way. DECA [76] has been specifically designed to regress accurate 3D face shape and animatable details that are specific to an individual and that can change

according to the subject’s expressions. The method is based on the FLAME face model. Another approach that has been designed with animation in mind is INORig [15]. The approach is built on an end-to-end trainable network that first parameterizes the face rig as a compact latent code with a neural decoder, and then estimates the latent code as well as per-image parameters via a learnable optimization.

The methods in the second category, i.e., shape regression-based methods, were developed to obtain a more accurate reconstruction than the 3DMM-based ones. Among the methods in this category, Jackson et al. [133] propose to straightforwardly map the input image pixels to a full 3D face structure in a voxel space via volumetric CNN regression. The main advantage of this approach is that the output face shape is not restricted to a face model space, therefore permitting applicability to the reconstruction of other objects (e.g., [319]). A similar idea is exploited by Feng et al. [75], who directly regress the face mesh from the input image without making use of a voxel space: the advantage of this method, called PRNet, is that it is lightweight and fast to execute, and it also usually leads to a precise and detailed reconstruction. Guo et al. [111] proposed the use of different CNNs to reconstruct the coarse-scale geometry and the fine detail. Recently, Wang et al. [309] proposed a framework that solves the problem of face reconstruction in three steps: face region segmentation, coarse-scale reconstruction, and detail recovery. Finally, Wang et al. [310] propose a novel unsupervised 3D face reconstruction architecture by leveraging the multi-view geometry constraints to train accurate face pose and depth maps. Once trained, the approach is able to perform reconstruction from a single image as well.

5.3.2 Comparing the 3D face reconstruction approaches

The core of a virtual try-on application for glasses is undoubtedly the 3D face reconstruction module. In the literature, several approaches differ in their underlying architecture, complexity, and performance. This section compares different 3D face reconstruction approaches that can be potentially

used in the virtual try-on application. This comparison can be used by developers and practitioners to select the most suitable approach for their specific task, depending on the operational and environmental constraints of the final application.

Among the possible approaches, only those that have made their code publicly available are considered: DECA [76], 3DDFAV2 [110], 3DSFMFace [310], Extreme3D [290], RingNet [253], Deep3DFace [63], INORig [15], and PRNet [75].

3D face reconstruction approaches are here compared with respect to different criteria: (i) underlying architecture and characteristics; (ii) computational costs (in terms of time and memory consumption); (iii) quantitative evaluation of the reconstruction error against a ground truth; (iv) qualitative evaluation of the goodness of the reconstructed face texture. These criteria capture different aspects of the reconstruction phase. To the best of our knowledge, no standard quantitative evaluation exists to assess the texture, so a subjective assessment with a panel of users has been carried out. Both the geometry and the texture are important for an assessment of the fidelity of the reconstruction and thus the user's acceptance of the try-on application.

Comparison of the architectures Table 5.4 summarizes the characteristics of the 3D face reconstruction methods. All the methods rely upon Neural Networks to perform feature extraction or parameter regression. The most common backbone used is the Residual Network, which has been successfully exploited in different application domains and is one of the most efficient and robust networks. All the methods need a face detector to locate the facial region and extract facial landmarks to be used for pose estimation and normalization. Any face detector could be used, but most methods already include one. Three of them do not include a face detector but assume that the input images have either a set of facial landmarks associated (Deep3DFace), or that the face has been already located and cropped (RingNet) or segmented (3DSFMFace). For these methods, images are provided in the intended format. All the methods, except Extreme3D, provide both geometry and texture as output. INORig is the only one that requires multiple images of the subject

as input. All the methods are implemented in Python and are based either on TensorFlow or PyTorch frameworks.

Table 5.4: Main characteristics of 3D face reconstruction methods in the state-of-the-art. SR: Shape Regression, MM: Morphable Model, S: single image, M: multiple images, G: geometry, T: texture, BM: BumpMap, BFM: Basel Face Model, DRN: Dilated Residual Network. TF: TensorFlow, PT: PyTorch.

Method	Year	Category	Input	Face Detection	Network Backbone	Output	F.work
PRNet [75]	2018	SR	S	DLib	U-Net	G + T	TF
Extreme3D [290]	2018	MM/BFM	S	DLib	ResNet	G + BM	PT
Deep3DFace [63]	2019	MM/BFM	S/M	External *	ResNet	G + T	TF
RingNet [253]	2019	MM/FLAME	S	External †	ResNet	G + T	TF
DECA [76]	2021	MM/FLAME	S	FaceNet °	ResNet	G + T	PT
3DDFAV2 [110]	2021	MM/BFM	S	FaceBoxes ×	MobileNetv3	G + T	PT
3DSFMFace [310]	2021	SR	S/M	External ‡	ResNet	G + T §	PT
INORig [15]	2021	MM/BFM	M	S3FD †	DRN	G + T	PT

* The method requires 5 facial landmarks along with the source image;

† The method uses loosely cropped face image as input;

° The method uses a fast version of MTCNN [320] for face detection;

× The method uses FaceBoxes face detector [333];

‡ The method requires that the face is segmented from the background;

§ The method outputs a colored point cloud;

† The method uses the Single Shot Scale-Invariant Face Detector [334].

Computational costs For each method, have been measured the time and memory required to perform the facial reconstruction, as well as the geometry error. Table 5.5 summarizes the results. The execution times reported in the papers describing and comparing the different methods are usually relative only to the 3D face reconstruction step and do not include the preprocessing and postprocessing steps. When building an application such as the virtual try-on, it is important to consider all the steps involved in the reconstruction process to evaluate the overall run-time. The source codes of the 3D face reconstruction methods were modified to perform a consistent evaluation across them. Each pipeline was initialized once and then used to process iteratively 101 loosely cropped face images from the FFHQ dataset [143]. The execution time for each sample includes all the steps from image loading to

Table 5.5: Evaluation results of 3D face reconstruction methods in the state-of-the-art. All the timings are computed on 101 images, discarding the execution times of the first one in order to simulate a hot start condition for the system. Hardware used: Intel Core i7 7700 CPU and NVIDIA Quadro RTX 6000 GPU.

Method	Time Min (Seconds)	Time Max (Seconds)	Time Median (Seconds)	Time Mean (Seconds)	Time Std (Seconds)	Memory (MB)	GPU Memory (MB)
PRNet [75]	0.736	0.808	0.751	0.749	0.010	2361	1161
Extreme3D * [290]	15.564	15.840	15.604	15.598	0.031	1968	1925
Deep3DFace † [63]	0.582	0.627	0.592	0.591	0.009	2867	1235
RingNet [253]	0.741	0.826	0.789	0.789	0.016	3108	23,035 °
DECA [76]	0.850	1.361	0.893	0.883	0.055	3022	18,227 †
3DDFAV2 × [110]	0.740	0.798	0.770	0.769	0.009	3290	4683
3DSFMFace ‡ [310]	0.349	0.636	0.438	0.435	0.053	3260	1379
INORig § [15]	2.901	3.228	3.049	3.037	0.081	4253	21,691 †

* CuDNN disabled due to incompatibilities with the GPU;

† Added face detector and 5-point descriptor from DLIB as a preprocessing step;

° The method seems to allocate all the available memory on the GPU even if the behavior is explicitly disabled;

× Method modified to reconstruct only one face even if more are detected in the input image, “-onnx” flag not used;

‡ Face segmentation done manually and not included in the run-times;

§ For each image, the reconstruction is performed on the pair input image and its horizontal flip;

† The peak memory usage is a short spike.

the creation of the 3D model representation (i.e., obj or ply files). Since the frameworks require some time for caching and loading of data on the first run, the evaluation discards the execution time of the first image to simulate a hot start of the system. The times of Table 5.5 are therefore relative to the execution of 100 face reconstructions on a machine with an Intel Core i7 7700 CPU and an NVIDIA Quadro RTX 6000 GPU. Some changes to the source codes were needed due to the differences in the input data and setup required by the methods. For the Extreme3D, the cuDNN back-end was disabled due to incompatibilities with the GPU. Deep3DFace needs five face landmarks as input in addition to the image; those were automatically computed in an additional preprocessing using the DLib library as suggested by the authors. Moreover, 3DDFAV2 reconstructs all the faces detected in the input picture by design; it was thus forced to work only on the first detection. The authors of 3DDFAV2 stated that using the ONNX Runtime [213] library it is possible to obtain a noticeable increase in the inference speed. Since this optimization library is potentially usable by the other implementations, the test uses the plain (non-onnxRuntime) version of their method for a fair comparison. INORig requires at least two images of the same person; the evaluation uses the input image and its horizontal flip, as done by the authors.

The majority of the methods can provide face reconstruction in less than one second on average. The only exceptions are INORig and Extreme3D; the former works on two images, while the latter performs part of the computation on the CPU, slowing down the process. The fastest method is 3DSFMFace, although its output is a point cloud and not a 3D mesh, as with the other methods. It also requires a segmented face over the input image, and the reported time does not include the time necessary to perform such segmentation since it was manually conducted.

System RAM usage and GPU dedicated RAM usage is also evaluated. Table 5.5 reports the peak memory allocated for both CPU and GPU by each method during the reconstruction of a single 3D face. System memory allocation varies between 2GB and 4GB, depending on the pipeline. On the GPU memory side, the amount of memory used varies between 1.1GB for PRNet and 4.6GB for 3DDFAV2. While most of the implementations have a constant GPU memory usage, DECA and INORig present some short spikes of allocation that bring the peak memory usage to 18GB and 21GB, respectively. The implementation of RingNet seems to allocate all the available GPU memory, even if this behavior is explicitly disabled in the TensorFlow library.

Reconstruction errors Reconstructed 3D geometries were evaluated on the UMB-DB dataset [59] to assess the geometrical error of different face reconstruction methods on the same input data. This dataset contains RGB images and the corresponding 3D ground truth geometry acquired using a Minolta Vivid VI-900 laser depth scanner. Reconstructions of 15 subjects were performed for each method starting from a single neutral expression input image without occlusions. Since the methods use different coordinate reference systems, a first coarse alignment matches the reconstruction to the ground truth geometry using the seven face landmarks annotated in the UMD-DB. This rigid alignment only allows the rotation, translation, and scale of the reconstructed geometry. Considering that the completeness of the reconstructed geometry varies between the methods, all the reconstructions were cropped to the same area of the face. Given that INORig is the method whose reconstruction includes the smallest portion of the face, it was decided

to crop out the parts that were not reconstructed by INORig—for instance, the ears (reconstructed by Extreme3D, PRNet, DECA, and RingNet) and the cranium (provided by DECA and RingNet). Another rigid alignment step was performed through the Iterative Closest Point (ICP) algorithm to register the cropped 3D reconstruction to the ground truth. Finally, the geometry was evaluated using the absolute distance between each vertex of the 3D reconstruction and its closest point on the ground truth mesh. These steps of alignment and distance evaluation follow the same procedure defined in Section 3.3.2 where the mesh vertices are used instead of a point cloud.

Table 5.6 reports the results of the evaluation. Since DECA does not provide a texture for the detailed 3D model, it was decided to evaluate the coarse one, which includes the texture image. 3DSFMFace was not evaluated due to the limited usefulness of the point cloud recovered for a virtual try-on application. As can be seen, all the values are similar and well within a reasonable tolerance for a try-on application. For completeness, the same table also reports the reconstruction performance of the methods on the NoW dataset as per the NoW challenge benchmark [187]. See [253] for further details. Extreme3D and 3DSFMFace have not been evaluated on the benchmark. Figure 5.8 shows some examples of 3D face reconstructions.

Table 5.6: Geometry evaluation results of 3D face reconstruction methods in the state-of-the-art.

Method	Median (mm)	Mean (mm)	Std (mm)	NoW Median † (mm)	NoW Mean † (mm)	NoW Std † (mm)
PRNet [75]	1.50	1.58	0.45	1.50	1.98	1.88
Extreme3D [290]	1.83	1.93	0.24	-	-	-
Deep3DFace [63]	1.35	1.50	0.45	1.11	1.41	1.21
RingNet [253]	1.46	1.43	0.16	1.21	1.53	1.31
DECA * [76]	1.30	1.46	0.34	1.09	1.38	1.18
3DDFAV2 [110]	1.66	1.65	0.41	1.23	1.57	1.39
INORig [15]	1.51	1.51	0.24	-	1.33 °	0.28 °

† Values from NoW Challenge [187];

* Evaluation of the coarse 3D model;

° Values from the INORig paper [15], not reported on the challenge website.

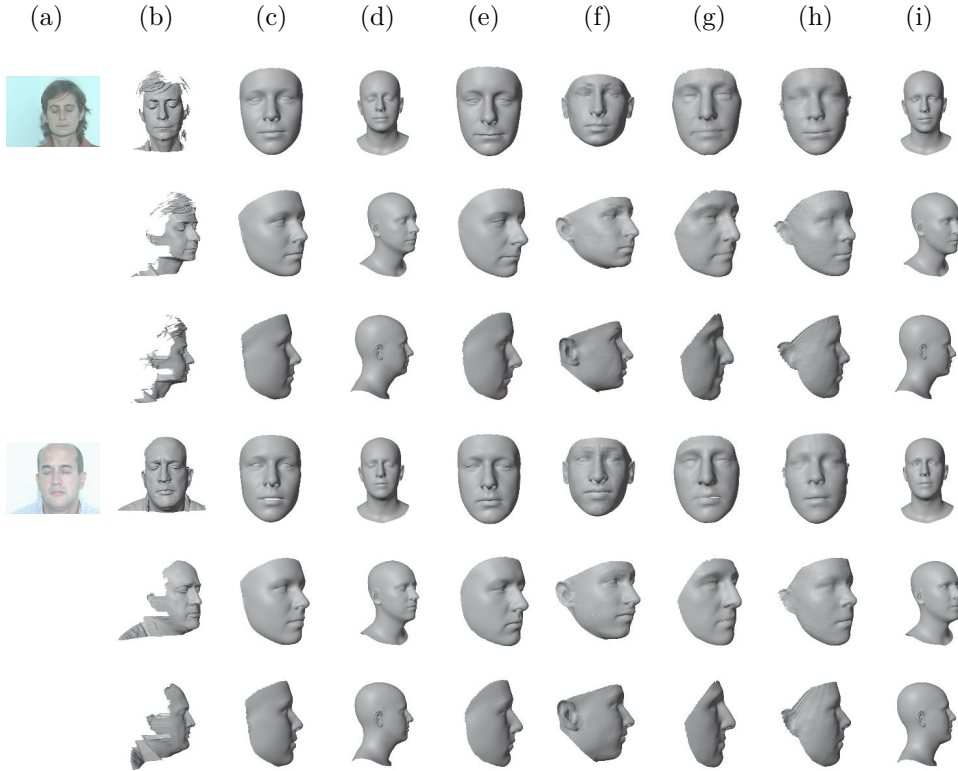


Figure 5.8: Examples of 3D face reconstruction results on images from the UMB-DB dataset [59]. For DECA, the figure shows the coarse mesh that, opposed to the detailed one, provides a texture. (a) Input image; (b) ground truth; (c) 3DDFAv2 [110]; (d) DECA [76]; (e) Deep3DFace [63]; (f) Extreme3D [290]; (g) INORig [15]; (h) PRNet [75]; (i) RingNet [253].

Texture quality As stated before, the application needs both the geometry and the appearance (i.e. the face texture) to be reasonably accurate. To this end, a subset of 10 face images was selected from the FFHQ dataset [143] as a test set. The images were processed with each method using the code, model, and parameters provided by their respective authors. The output of each method was then evaluated in a comparative way. Given an input 2D image, the different 3D outputs were assessed by a panel of subjects that compared the results against the original image and selected the best 3D output with respect to the fidelity of the reconstructed texture and potential appeal within a try-on application.

Table 5.4 shows that Extreme3D does not output a texture, so it was excluded from the subjective evaluation. Moreover, 3DSFMFace was excluded because it outputs a 3D point cloud and not a mesh. For the assessment of the six methods, a simple web application was developed to show the 3D models reconstructed by the methods with their texture applied. The users were chosen among researchers and postgraduate students of the University of Milano-Bicocca. All the users had normal or corrected-to-normal vision and were knowledgeable about virtual try-on applications and 3D modeling. The users were asked to rank the results from one (the best) to six (the worst). Ties were allowed. The responses were collected and average rankings are provided for each method. In total, 11 users participated in the experiment. Figure 5.9 shows some of the texture results judged in the subjective experiment. In the web application, the users were able to scale and rotate the models to inspect them more thoroughly.

The average rank of each method is as follows: PRNet: 1.93, 3DDFAv2: 2.95, RingNet: 3.07, DECA: 3.59, Deep3DFace: 4.63, and INORig: 4.85. Figure 5.10 shows how many times a method was ranked at a given position. Overall, PRNet was judged to provide the best texture on the samples. RingNet and 3DDFAv2 have similar ranks. Next is DECA, which has been voted mostly in the fourth position. Finally, Deep3DFace and INORig gave similarly poor results. It was surprising that PRNet was judged the best against more recent methods such as DECA or INORig. This can be explained in that the reconstruction methods are mostly designed with geometry reconstruction as the main goal. The visual texture is usually not considered as a main focus and is used only for visualization. From the experiment, it emerged that one of the problems of existing methods based on 3DMMs is that they tend to create gaps in the mouth when the person is smiling. Since there is no texture for the inner mouth, this creates an uncomfortable hole in the texture. Postprocessing is required to cope with this issue. PRNet, being based on a shape regression technique, has no such problem: the mesh is closed, and the texture is complete. This can be seen in the third row of Figure 5.9.

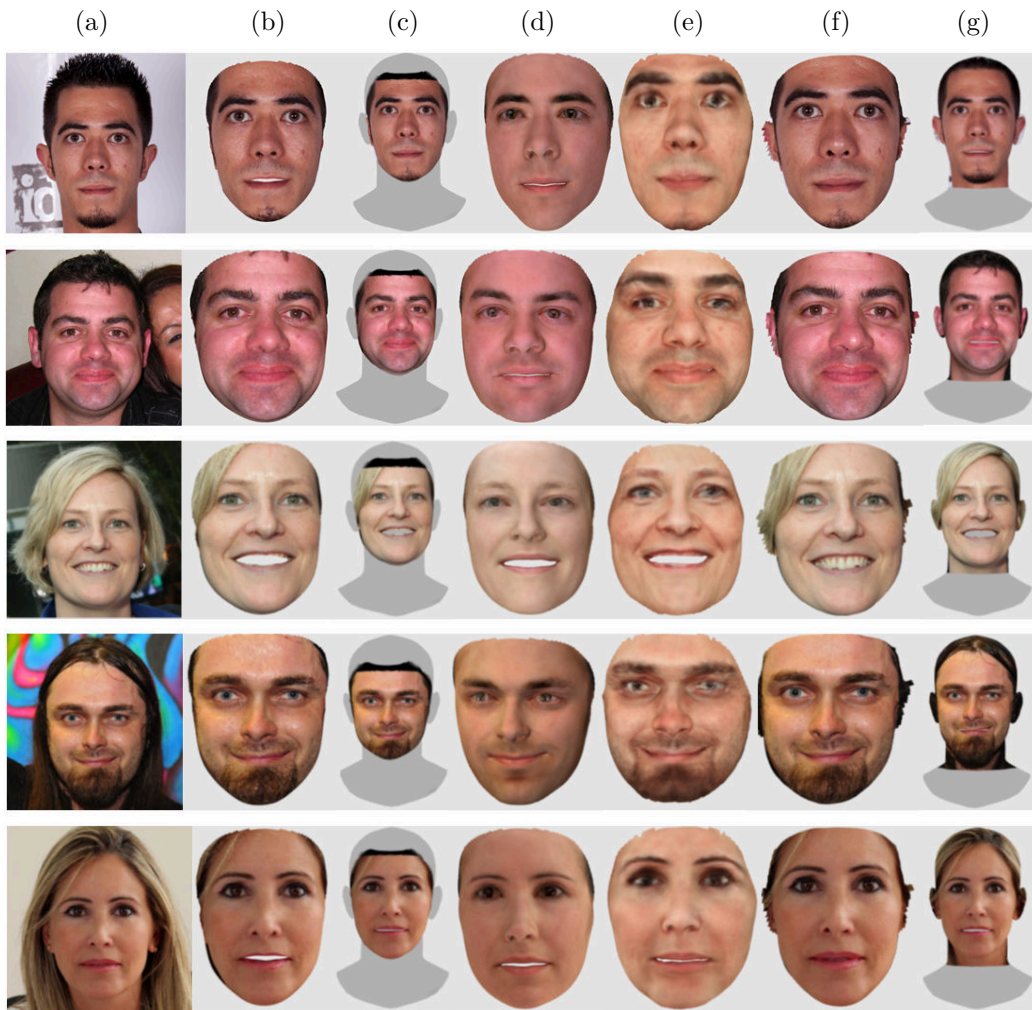


Figure 5.9: Some samples used in the texture quality experiment. (a) Input image; (b) 3DDFAv2 [110]; (c) DECA [76]; (d) Deep3DFace [63]; (e) INORig [15]; (f) PRNet [75]; (g) RingNet [253].

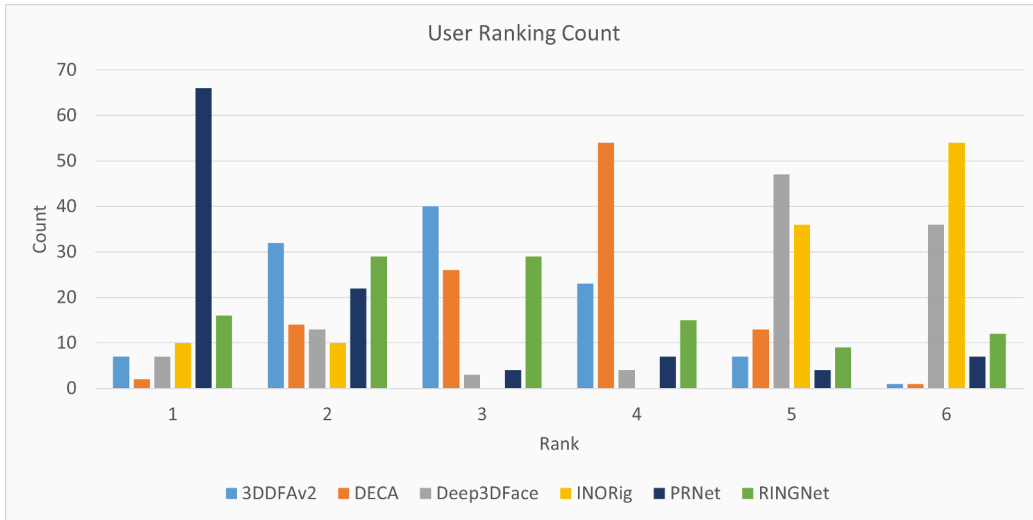


Figure 5.10: User ranking of the 3D face reconstruction methods in the texture quality experiment.

5.3.3 Workflow

The currently available virtual try-on applications for face accessories present some limitations. For example, many applications show the results of the virtual scene in a static 2D image; this greatly limits interactivity. Other applications require some kind of physical marker to be acquired along with the user’s face for size estimation. This can be cumbersome in the case of mobile applications. Moreover, the use of a video stream as output limits the degrees of freedom with which the user can see himself with the accessories.

This section describes the proposed virtual try-on solution for eyewear. To overcome limitations in existing try-on applications, the proposed solution leverages artificial intelligence methods, specifically CNNs, for the robust detection and tracking of the face of the user. The solution is designed to be more user-friendly, interactive, and easy to use than existing applications. To achieve this, the application creates a complete 3D virtual scene, with the 3D-reconstructed head and face of the user wearing the selected eyeglasses. The reconstruction is performed using a single 2D image of the user’s face. After the 3D reconstruction, the face size and the fitting parameters for the eyeglasses model are automatically computed, leveraging information

5.3 The eyeglasses Virtual Try-On use case

from the user's face and without any external physical marker. The result is displayed using a 3D rendering framework that also allows the user to rotate the reconstruction and test different glasses models. By using the full 3D results, the user has a realistic idea of how the eyeglasses will look on himself and can freely pose the virtual head, viewing the glasses model from any point of view. Figure 5.11 shows an overview of the workflow of the try-on solution. The back-end of the system is implemented in the cloud, while its front-end is implemented as a web-based application. In the proposed workflow, the localization module is split into the face detection and 3D reconstruction sub-modules. There is no tracking module since a 3D representation of the user's face is reconstructed. Face size estimation, face keypoint detection, and fitting parameter estimation are supporting modules used to blend the chosen eyeglasses and the user's reconstructed face with the correct sizes and proportions for a realistic and faithful rendering.

The following subsections provide further details on each component.

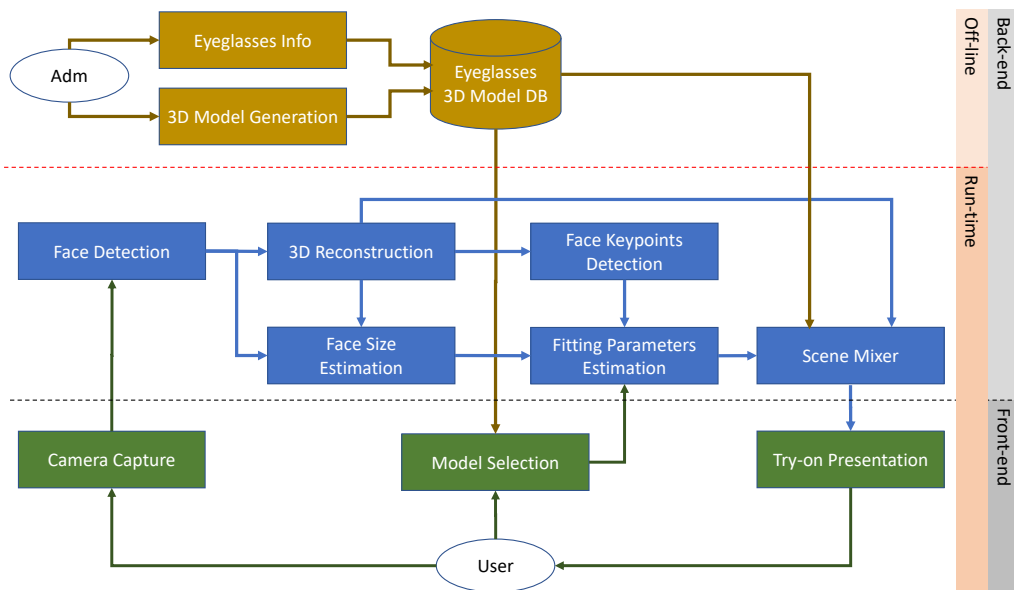


Figure 5.11: Workflow of the 3D eyeglasses virtual try-on system.

5.3.3.1 Face detection and 3D reconstruction from a single image

Section 5.3.1 surveyed several approaches in the state-of-the-art for 3D face reconstruction from a single image. In theory, any of the mentioned approaches can be exploited to build a virtual try-on application. However, we must take into account that in a real application scenario the approach used must be computationally efficient, and it should reconstruct not only a reasonable 3D geometry of the face, but also an accurate replica of the facial details via a texture to be applied on the 3D model. Without all these constraints, the try-on application would not be appealing to the final users, both in terms of usability and fidelity.

Implementation of the 3D face reconstruction The creation of the virtual scene starts with the user's face detection in a captured photo. To reduce the computational burden, the detection must be executed with a fast and lightweight algorithm. Among the possible algorithms in the state-of-the-art, it was selected the CNN-based face detector in the DLib software library [149]. The detector is very robust and accurate; it takes as input an RGB image and produces as output a bounding box of the area of the face. In the case of multiple detections, only the largest located face is used to run the reconstruction and the subsequent try-on process, thus avoiding unintentional runs of the try-on over faces that may be present in the background of the input picture. Although face detection and reconstruction can work with low resolution images, the face is acquired at least with a resolution of 1000x1000 pixels in order to have sufficient information in the area of the eyes.

Based on the results in Section 5.3.2, the PRNet [75] method was selected for the 3D face reconstruction phase. This method allows us to not enforce any particular restriction on the face pose or occlusions because the reconstruction process can work on almost every image depicting a human face. The input of the 3D face reconstruction module is the area of the face detected by the face detection module. The area is first resized to a 256x256 pixel RGB image and used as the input of the CNN. The output of the network is a vertex map, a 256x256 matrix where each element is a vertex of the reconstructed

face. The face mesh is generated from the 65k vertices in the regressed vertex map. Its texture image is generated by remapping the input image to the reconstructed mesh using the regressed vertices map. The resolution of the texture map is 512x512 pixels. This resolution is deemed of sufficient quality for the application.

If occlusions are present in the input image, the corresponding portions will be missing in the generated texture image. This is a limitation of the single image 3D face reconstruction approach. The face mesh is always complete, regardless of the initial face pose, but the face parts not visible to the camera are not textured correctly. This problem could be partially solved by using multiple face acquisitions from different points of view, and then using Structure From Motion approaches [32] so that the 3D face can then be reconstructed. However, this solution could annoy the user since it requires many images in order to have a faithful reconstruction. For this reason, a very fast and realistic 3D face reconstruction using a single image with acceptable coverage of the face texture is preferable. To ensure this, if the detected face has a very skewed pose, the user is asked to take another shot in a more frontal position. Furthermore, to keep dimensions consistent between reconstructions, the vertex map is rotated according to the detected face pose to obtain the front view and is scaled in a cube of size 1x1x1 units in 3D world space, centered in the origin of a canonical right-handed global coordinate reference system. The 3D face model will be scaled to its final true size according to the parameters estimated by the fitting parameter estimation module, as described in Section 5.3.3.2.

In addition to the reconstructed 3D face, the 3D reconstruction module outputs other information needed to estimate the true size of the face: the 68 landmarks defined by the Multi-PIE landmark scheme [108] used for locating the eye regions, and the face keypoints recovered from the vertex map, used for determining the fitting parameters for the eyeglass frame. This information is sent to the face size estimation module.

The average time required by the 3D reconstruction module is in the order of 0.62 s, of which 0.06 s is for face detection, 0.01 s is for mesh reconstruction, and the remaining time is for texture generation.

5.3.3.2 Face size estimation

Building a virtual try-on application requires that the wearable item(s) must be of the correct size when placed on the virtual model. With respect to the eyeglass application, this means that the glasses frame size must match the face size. Existing try-on applications often lack a proper face size estimation, and this cannot provide the user with a realistic try-on experience in terms of size fitting. To cope with this problem, some commercial applications estimate the face size using markers whose sizes are known, such as a credit card, placed on the forehead [66]. The use of markers is a common approach but it requires the user to inconveniently perform additional acquisitions and to have the proper marker at hand. Another problem with this approach is that the marker must be clearly visible, forcing the user to perform actions to deal with acquisition errors. This can negatively influence the overall experience, annoying the user.

To deal with the above-mentioned issues, is here proposed a markerless approach for estimating the face size. The proposed approach exploits facial features without requiring additional items to be acquired. Specifically, the diameter of the iris is used as a reference to determine the actual face size. By measuring the iris diameter in pixels, and knowing its average diameter in millimeters, we can estimate the actual face size. The iris diameter is related to the cornea diameter and, according to Rüfer et al. [247], the average white-to-white horizontal diameter is 11.71 ± 0.42 mm.

The complete flow of the face size estimation process is summarized in Figure 5.12, and Appendix C shows the algorithm. First, the eye location is identified using the landmarks provided by the 3D face reconstruction module. Then, it crops the eye regions, extracts the red channel, and applies a gamma correction to enhance the visibility of the iris and pupil. Inspired by Kerrigan et al. [146], the processed eye crop were fed into a DRN-D-22 network [329] to generate a segmentation map of the iris. Ideally, near-infrared images should be used, but would require the adoption of additional hardware. Through experimentation, it was found that gray-level images can be successfully used. Finally, the Hough Circle Transform fits a circle on the iris in the segmentation

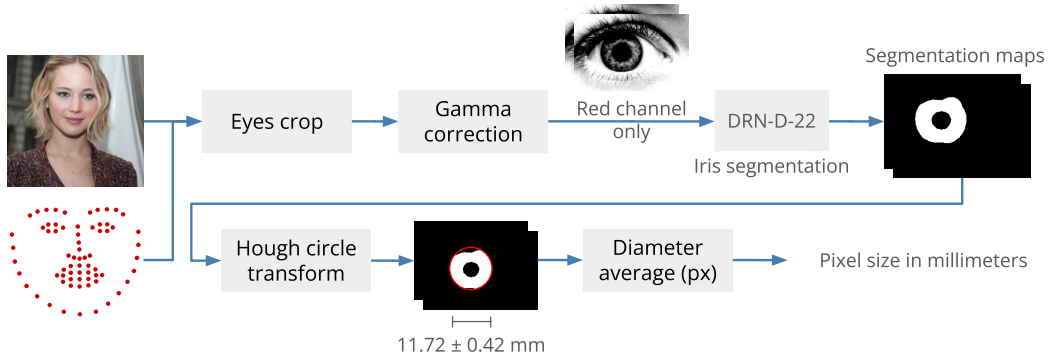


Figure 5.12: Face size estimation workflow.

map and finds its radius in pixels. The size in millimeters of each pixel is finally computed as $11.71/2r$, where r is the estimated iris radius in pixels. This procedure is applied to both eyes, and the final iris size estimation is the average of the two values. If the calculation fails for one eye, the single value obtained is used. In the unfortunate event that the estimation fails for both eyes, the subsequent step of glasses fitting estimates the parameters to best fit the glasses on the user's face without using the size information; in this case, the user is notified of such failure.

It is then possible to align the 3D reconstruction on the input image and compute the face size using the distance between the ear keypoints as the number of pixels between them multiplied by the size of each pixel in millimeters determined from the iris size. The whole face size estimation procedure requires 0.13 s on average.

5.3.3.3 Fitting parameter estimation

Once the face size is estimated, we need to define the geometric transformation required to correctly place the glasses on the face. This is done by a fitting procedure that generates the transformation parameters required for rendering the face with the selected glasses frame. Since the glasses frames have different geometries, we need to perform the fitting for each selected one with respect to the current face geometry.

The parameters are estimated by finding the transformation that aligns

the glasses frame on the reconstructed face when viewed from a frontal pose. The alignment is performed by using some of the facial landmarks extracted from the reconstructed 3D face (facial keypoints) and keypoints extracted from the glasses model (eyeglasses keypoints). For this reason, all the available glasses models in the database are assumed to be already annotated with these keypoints, along with all the other relevant information. Specifically, available information must include the brand name, model name, preview image, width, and keypoints. The required eyeglasses keypoints are shown in Figure 5.13a. They correspond to the bridge location (G_n) and both the temples' far end-points, where the glasses lean on the ears (G_l and G_r). The corresponding facial keypoints, extracted from the reconstructed 3D face, are shown in Figure 5.13b. At the end of the fitting procedure, the two sets of keypoints geometrically overlap and the glasses are fitted to the face, as shown in Figure 5.13c.

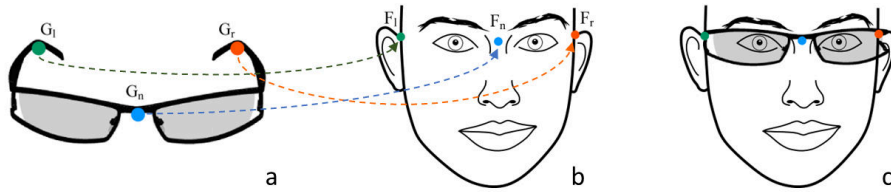


Figure 5.13: Example of keypoints for glasses fitting. Nose keypoints in blue, left ear keypoints in green, right ear keypoints in red. (a) Eyeglasses keypoints sample. (b) Example of facial keypoints. (c) Eyeglasses fitting result.

The fitting process requires several stages. First, it is necessary to determine the glasses' position. This is done by using Equation (5.1), which computes the translation transformation that aligns the glasses bridge keypoint G_n with the face nose keypoint F_n , which is used as a reference point:

$$\mathbf{t} = F_n - G_n \quad (5.1)$$

Then, the glasses' temples are laid on the ears. This is done by computing the projection of the keypoints on the 2D plane defined by the Z and Y axes,

as shown in Figure 5.14 and Equation (5.2):

$$\begin{aligned}
 \hat{\mathbf{g}}_{nl} &= \frac{G_l - G_n}{|G_l - G_n|} \\
 \hat{\mathbf{f}}_{nl} &= \frac{F_l - F_n}{|F_l - F_n|} \\
 \alpha &= \arctan2(|\hat{\mathbf{g}}_{nl} \times \hat{\mathbf{f}}_{nl}|, \hat{\mathbf{g}}_{nl} \cdot \hat{\mathbf{f}}_{nl}) = \arctan2(\sin \alpha, \cos \alpha)
 \end{aligned} \tag{5.2}$$

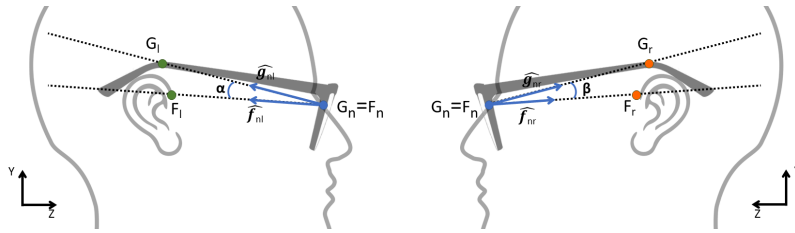


Figure 5.14: Eyeglasses pitch angle between ear and temple keypoints.

The direction from the nose keypoint (F_n) to the glasses left keypoint (G_l) is computed. It is then normalized to a unit vector, obtaining the unit directional vector that we indicate as $\hat{\mathbf{g}}_{nl}$. Similarly, the unit vector $\hat{\mathbf{f}}_{nl}$ is defined to represent the direction from the nose keypoint (F_n) to the left ear keypoint (F_l). It is then computed the angle α between the two unit vectors using the 2-argument arctangent, where the first argument is the norm of the cross product between $\hat{\mathbf{g}}_{nl}$ and $\hat{\mathbf{f}}_{nl}$, and the second argument is the dot product between them. With $\hat{\mathbf{g}}_{nl}$ and $\hat{\mathbf{f}}_{nl}$ being unitary vectors, the first argument corresponds to the sine of the angle α between them, and the second one to the cosine of α . In a similar way, a second angle β is computed using the direction from the nose keypoint to the glasses right keypoint ($\hat{\mathbf{g}}_{nr}$), and the direction from the nose keypoint to the right ear keypoint ($\hat{\mathbf{f}}_{nr}$). Finally, the rotation transformation around the X-axis needed to lean the glasses on the ears is built using the mean rotation angle between α and β .

The process of scaling the glasses frame to match the face size, takes into account the face size estimation described in Section 5.3.3.2, and the difference between the mesh and face coordinate systems. In case of problems in the face size estimation, the scale is determined as the mean X-axis scale factor that best fits the temples on the ears. Figure 5.15 shows the difference

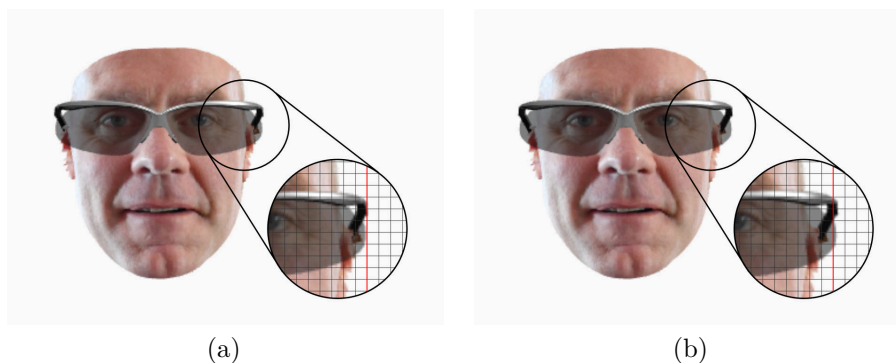


Figure 5.15: Examples of eyeglasses fitting using keypoints and face size. (a) The glasses are scaled to match the keypoints of the face. (b) The glasses are scaled according to the estimated face size and the real glasses sizes.

between the proposed fitting solution and a simple approach based on keypoint correspondence. The keypoint correspondence approach does not take into account face and eyeglasses dimensions. The glasses are simply scaled to make the keypoints match the face ones (Figure 5.15a). Instead, with the new approach, the glasses are slightly thicker on the face with respect to the keypoint fitting approach (Figure 5.15b).

The overall fitting parameter estimation requires around 0.4 milliseconds for each eyeglasses model in the library. The 3D face mesh, the eyeglass model, and the fitting parameters are passed to the scene mixer module in order to render the final virtual 3D scene. Some virtual try-on results with different face pictures and eyeglasses models are visible in Figure 5.16, where the input images are taken from the FFHQ dataset [143], which provides high-quality images of human faces.

5.3.3.4 User interface

The virtual try-on user interface is provided as a web application. The web application is responsible for image acquisition and user interaction. It also interacts with the back-end modules, specifically the face reconstruction and fitting parameter estimation modules, accessed as web services. The



Figure 5.16: Examples of virtual try-on results on images from the FFHQ dataset [143]. Glasses models from <https://www.cadnav.com>. (a) Input; (b) Front view; (c) Side view; (d) Input; (e) Front view; (f) Side view.

use of a web-based front-end allows us to deploy the try-on application on different devices with minimum system requirements, provided that modern web browsers are available. The user uploads a picture or snapshot of his face using the web browser, along with the choice of the desired glasses frame. The image is received by the server, and processed by the designed processing pipeline. The result is returned to the browser, allowing the user to see the glasses from different points of view.

The web application has been implemented using a set of open-source

5.3 The eyeglasses Virtual Try-On use case

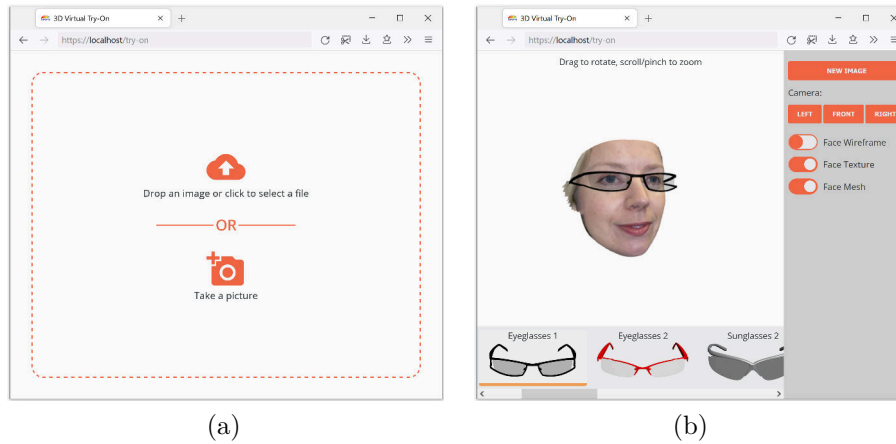


Figure 5.17: Screen captures of the virtual try-on web application user interface. (a) Image upload interface; (b) virtual try-on interface.

tools and frameworks. Specifically, it uses *Vue.js*¹ as the core JavaScript framework, *Dropzone.js*² as the image upload handler, and *babylon.js*³ as an advanced 3D rendering framework to manage Physically-Based Rendering (PBR) Materials and lens transparency. The 3D models of the face, as well as those of the glasses, are exchanged using the Wavefront OBJ file format accompanied by the material definition file MTL and the texture images. While the 3D face model is automatically loaded shortly after the try-on back-end completes the processing, the 3D glasses models are dynamically loaded based on the user's selection. Since the web application must still be usable on low-end devices and slow internet connections, the 3D models of the glasses have been optimized to reduce the size of the files to be transferred, in order to keep loading times below one second in most cases. For the glasses models, in addition to the geometry and the materials, the application also loads a JSON file containing the translation, rotation, and scale parameters to render them correctly on the 3D face.

Figure 5.17 shows the user interface, which is composed of two main pages; the first is for face image upload or acquisition, and the second one is

¹<https://vuejs.org/>

²<https://www.dropzonejs.com/>

³<https://www.babylonjs.com/>

to display the 3D virtual try-on. The displayed acquisition interface is for desktop/mobile devices. If the application is installed in physical stores, a continuous video acquisition can be implemented, as described in the previous sections. The majority of the screen area is used to render the 3D face and glasses. The list of glasses available for try-on is displayed at the bottom of the screen, and a few controls for the rendering and some predefined camera positions are available in the right sidebar. If the user's device has a screen with a limited size (e.g., a smartphone), the sidebar starts in a collapsed state to leave sufficient space for the try-on rendering. The face can be rotated by dragging, and the zoom level can be adjusted by using pinching or scrolling, depending on whether the device supports touch controls or not.

5.3.4 User evaluation

A standard usability test [207] with a panel of users has been used to evaluate the user experience. A total of 16 subjects of different age, expertise, and educational background were selected.

The experiment was conducted using a 10-inch tablet where the user interacts with a touch interface and a desktop PC where the user interacts with the mouse. The participants were randomly split between these two interaction modalities. Before starting the usability test, the application scope was briefly described to the users. They were then observed using the application to perform a virtual try-on session by taking a picture of themselves. No time limit was imposed. At the end of the test session, each user judged his experience by filling in a questionnaire. The questionnaire was the standard System Usability Scale (SUS) questionnaire developed by John Brooke [39]. In order to gain more insights into the application, users were asked to rate (from 1 to 5): the precision of glasses fitting, the realism of the 3D view, the usefulness of such a 3D view, and their interest in using this application if made publicly available. Finally, some free comments from the users were also collected.

Results of the SUS questionnaire are summarized in Table 5.7. As can be

seen, the users rated the system very positively. The try-on was considered easy and very simple to use. No previous knowledge or experience was necessary to approach it. By applying the standard procedure [39], the application obtained an overall SUS score of approximately 90 out of 100, which is considered above average. The average score is set to 68 from a study on 500 systems, as described in [255]. Table 5.8 shows the scores of the four additional questions specific to the try-on application. The virtual try-on application with the 3D visualization of the face and glasses received very high appreciation, with scores of 4.9 out of 5. The users found this type of application enjoyable and useful and were more than happy to try more. The technology was found to be engaging, although the overall visual quality of the rendering should be improved. This is demonstrated by the slightly lower score for fitting precision and realism, with a score of 3.8 and 3.7, respectively.

In general, the opinions of the users were quite positive. The majority of them appreciated the ease of use of the application. The users also commented on the usefulness of the 3D visualization and lamented the lack of this visualization in other online try-on systems. Finally, the automatic fitting of the glasses frame was appreciated. However, some users suggested to allow a manual resize and position of the frame. This could be interesting for a vendor that personalizes glasses frames or even produces custom eyeglasses to meet the user's preferences. Many users requested the possibility to integrate the application with social media in order to share their selection.

Users also pointed out some problems with the application. The main concern was the lack of hair and neck in the face model. These are limitations of the actual 3D reconstruction model used that are common to many other reconstruction approaches. One way to cope with the hair problem is to use more recent 3D reconstruction approaches, such as the one proposed by Xu et al. [323], which tries to also reconstruct the hair. However, these approaches are quite limited, and the hair is not fully reconstructed. Missing chunks of hair may annoy the user more than having no hair. Another way to cope with the hair is to employ a dedicated CNN [338]. This could potentially solve the problem at the cost of adding computational complexity and time. Another common opinion is that the 3D face mesh needs improvements in

Table 5.7: System Usability Scale (SUS) results.

	Statement	Strongly Disagree				Strongly Agree	Avg.	SUS Score
		1	2	3	4	5		
1	I think that I would like to use this application frequently.	0	0	8	6	2	3.6	2.6
2	I found this application unnecessarily complex.	11	5	0	0	0	1.3	3.7
3	I thought this application was easy to use.	0	0	0	1	15	4.9	3.9
4	I think that I would need assistance to be able to use this application.	15	1	0	0	0	1.1	3.9
5	I found the various functions in this application were well integrated.	0	0	3	6	7	4.3	3.3
6	I thought there was too much inconsistency in this application.	15	1	0	0	0	1.1	3.9
7	I would imagine that most people would learn to use this application very quickly.	0	0	1	2	13	4.8	3.8
8	I found this application very cumbersome or awkward to use.	11	4	1	0	0	1.4	3.6
9	I felt very confident using this application.	0	0	0	8	8	4.5	3.5
10	I needed to learn a lot of things before I could get going with this application.	16	0	0	0	0	1.0	4.0

Table 5.8: Application-specific question results.

Feature	Average Rating (Range 1–5)
Glasses fitting precision	3.8
Realism of the 3D view	3.7
Usefulness of the 3D view	4.9
Favorable to virtual try-on	4.8

both geometry and texture. As before, it is also possible to use more recent 3D reconstruction approaches and super-resolution techniques to cope with these issues. Finally, some users suggested that adding highlights and reflections on the lenses would make the 3D model more realistic and appealing. This can be obtained by adjusting the properties of the materials and incorporating shaders in the rendering engine used to display the models.

5.3.5 Known limitations

The proposed application uses a customized version of the MR catalog framework to overcome the identified limitations of existing virtual try-on applications for face accessories: first, the eyeglasses virtual try-on is performed in the 3D space using a 3D model of the user's face, reconstructed from a single input picture, allowing the user to interact with the application in a deferred mode and giving the possibility of observing the results from different points of view. Second, the glasses fitting takes into account the actual size of the glasses frames and the actual size of the face of the user, providing a realistic size fitting. The user's face size is estimated with a markerless approach that exploits the iris diameter, thus removing additional acquisition steps as required by other applications.

The results of a usability study of the eyewear virtual try-on system shows that the application is engaging and easy to use; the users appreciated the full 3D try-on and expressed willingness to use the proposed application if made publicly available. Future works can overcome some limitations observed in the usability study, such as the global quality of the rendering, which should be improved by providing better and complete 3D face reconstruction, and enhanced textures. Another possible extension is that of designing a processing pipeline capable of handling the possible presence of occlusions and self-occlusions in the input picture: while the face geometry reconstruction is possible even in the case of occlusions or self-occlusions, the texture will present missing parts. Restoration techniques such as [30, 58] can be used to recover the missing portions, or the application may ask the user to provide a non-

occluded face picture. Finally, the application can take advantage of better integration with online stores and social media services. Providing the try-on experience through a mobile application will also improve the integration with user devices: in this case, the new front-end mobile application can use the existing back-end web service.

5.4 Experimenting Mixed Reality catalogs on Smart Mirrors

This section discusses a proposal for the fruition of Mixed Reality catalogs on smart mirrors. A prototype smart mirror design is proposed which includes the ability to provide the eyeglasses virtual try-on experience presented in Section 5.3.

As previously explained the virtual try-on technology allows a user to virtually check the appearance of some product on himself. This process is usually carried out using mobile devices or desktop computers but, thanks to recent developments in the field, smart mirrors are emerging as a new way to provide the virtual try-on experience. The main concept is to add smart features to an object that everyone owns. The idea of a Smart Mirror is not something new and the very first concept of this novel idea can be dated back in the late 1990s and early 2000s science-fiction movies. Even though it started as a “Do It Yourself” (DIY) project around 2013-2014, the first known smart mirror was built by Michael Teeuw back in 2014 using a Raspberry Pi 2 (MagicMirror⁴). The original project has moved forward with version 2 of the Magic Mirror (MagicMirror² ⁵). On the same year as MagicMirror was released, Toshiba presented their own concept of “smart mirror” at CES [178]. Those mirrors, using a reflective screen, were able to display general information such as weather, news, and email but also personal data like calories consumption, heart rate, steps, etc. obtained from connected devices.

⁴<http://michaelteww.nl/tagged/magicmirror>

⁵<https://magicmirror.builders/>

With time the concept of Smart Mirror evolved and nowadays it's really easy to create a smart mirror with those basic features whereas more specialized types of smart mirrors started to emerge. Based on the field of use, smart mirrors can be classified into: general purpose, medical, fashion, and hotel.

General purpose smart mirrors are those built exclusively for everyday use. These mirrors are usually able to display general information such as news, weather, time, calendar, reminder, and alarms. The latest mirrors introduced more complex features (e.g., email reader, media player, browser, etc.) and security features (e.g., facial and voice recognition).

Medical smart mirrors are an advanced version of the general purpose mirrors. They are in fact capable of displaying general information, but also medical features such as facial expression detection, emotion detection, skin problem detection, body pose detection, posture detection, etc. Furthermore, they can include related tips and tricks to help the user to solve the discovered problem.

Fashion smart mirrors commonly include virtual try-on technology. The goal of these mirrors is to enable customers to try on products such as clothes, shoes, cosmetics, or jewelry using integrated cameras and AR.

Hotel smart mirrors comprise specific features for the hotel in which they are currently installed. These features usually allow the user to access room amenities, such as changing room temperature or humidity, control room devices (e.g., TV or lighting), pay for additional services, and book and receive notifications from the hotel staff. These mirrors are not sold to the general public, and not many hotels are currently adopting this technology.

5.4.1 Related work

The popularity of smart mirrors has rapidly increased and many articles on the subject are featured in the literature. Most of the existing work describes the design of a general type of smart mirror that includes only simple basic functions such as news, weather, alarms, time, etc. However, there are works involving smart mirrors developed for medical or fashion

purposes. This section briefly reviews the state-of-the-art smart mirrors both in the research/prototype stage and commercial ones. Considering the field of application of this thesis more relevance is given to fashion-related smart mirrors.

Smart Mirror prototypes Several general-purpose smart mirrors are presented in the literature. The majority of them are equipped with an RGB camera used for interaction. They usually activate when they recognize a user in front of them [31, 62, 194, 283]. Several smart mirrors are equipped not only with visual input but also with audio input. Most of these mirrors use voice control algorithms to access the mirror functions [14, 144, 327], while others exploit the combination of audio-video recognition to obtain more robust security protocols [89, 138].

In the medical field, smart mirrors can improve both clinical and at-home healthcare. Several proposals rely on the analysis of the face and facial expressions for the daily personal check-ups [11, 29, 122, 305]. Some mirrors complement the diagnosis with alternative medicine treatments such as music therapy [131, 330] and color therapy [324].

Some smart fashion mirrors include recommendation systems. For example, in [264] to suggest the ideal outfit based on the user's mood or in [205] the makeup that best suits the user's face. The use of Augmented Reality (AR) and Virtual Reality (VR) in smart mirrors to improve the customer experience is increasingly widespread. In fact, both AR and VR can help make retail stores more interactive and the online experience more real [169, 250, 308].

Commercial Smart Mirrors This paragraph is dedicated to commercial smart mirrors that are developed outside the research laboratory and the DIY community. The smart mirror is still a growing market and companies are still developing their own product. However, we can find some smart mirror products on the market. There are few general-purpose commercial smart mirrors. Ekko [118] offers basic features such as general information and a personalized profile and can be controlled with hand gestures. Griffin Technology [61] developed the Connected Mirror, a smart mirror showing

general information (time and weather, phone notifications), and updates from other Griffin devices. Toshiba [178] created a mirror that can help the user prepare recipes and act as a personal fitness monitor through a connection with the smartphone. Chakra Groups [48] released a smart mirror offering health-related features (e.g., tracking for weight, calories, sleep, and exercise) by connecting to Apple Health or Fitbit.

As part of the smart mirror for the fashion category, Memories [190] uses a Multi-Layer AR and AI engine. It allows a realistic and personalized augmented reality experience to try on clothes with colors and patterns of user's choice. Instead, Hi-mirror Plus [46] was an intelligent makeup mirror that could detect and analyze the condition of the user's skin and offer advice to hide imperfections.

Anna smart mirror [5] was developed for hotels and, through the recognition of hand gestures, allows booking transport, viewing general information, integration, and management of social web via a web app. Philips [117] produced a smart mirror/TV that can be installed in hotels and helps the customer pay their bills or pay-per-view movies. In 2017, Panasonic [220] unveiled Digital Concierge, a smart mirror powered with IBM Watson advanced functionalities.

More recently, CareOS [44] presented Themis, a small smart mirror that can track the user's condition by collecting data from different sensors such as a high-quality RGB camera, an IR temperature sensor, and a UV light for skin analysis.

5.4.2 The proposed Smart Mirror

The proposed smart mirror provides the user with an interactive interface that can comfortably be used in a home or store environment. The mirror can provide the eyeglasses virtual try-on experience. The following sections describe how the mirror interacts with external stimuli and what technologies (hardware and software) are used to develop the prototype.

5.4.2.1 Functional requirements

The smart mirror is normally in standby mode until a subject standing in front of it acquires the *active user* role and triggers the mirror. The presence of an active user activates the visual interface and allows him to access and interact with all the built-in functionalities. If there are multiple subjects in front of the mirror, the subject whose detected face has the highest resolution is selected (usually the user closest to the mirror). Through visual and audio stimuli, the user can interact with the mirror and have access to the eyeglasses virtual try-on. The audio is the main stimuli for the interaction with the Alexa-based services; voice commands are used to initialize and terminate the virtual try-on session.

Visual Interaction When the mirror is in standby mode, the face detection module is executed in the background. It consists of two steps: localization of the face in the current frame, and alignment of the cropped face region. For both steps the implementations are based on the algorithms provided by the Dlib library [149]. The face detector is built using a sliding window detection scheme based on the Histogram of Oriented Gradients (HOG) as a descriptor combined with a linear SVM as a classifier. An image pyramid is used to detect faces at different resolutions. For the success of the subsequent modules, faces should be detected in a frontal or near-frontal pose. For this reason a HOG-based face detector is preferred to a deep learning-based one. The first, in fact, although less robust, is sufficiently accurate and very efficient. Regions corresponding to detected faces are cropped to obtain facial images.

In addition to face detection, hand palm detection is used in the phase of virtual try-on to rotate the 3D virtual face. By moving his hand in front of the mirror the user should be able to see the 3D model from different angles. This detection is performed using Mediapipe's palm detection model.

Audio Interaction The audio interaction is mainly intended to complement and support the visual one. Audio interaction is incorporated into the mirror

through Amazon's Alexa Virtual Assistant. The Alexa virtual assistant module extends the interactive capabilities of the mirror thus allowing us to make the mirror smarter. Thanks to Amazon's Alexa it's possible to achieve a grade of artificial intelligence that can boost and facilitate the user's interaction with the mirror. In particular, the system is able to start a conversation if certain conditions are met, and answer specific questions or requests thanks to customized skills. A custom skill coordinates the eyeglasses virtual try-on process.

Display Information The display monitor is the principal tool for showing information and mirror status. During the standby mode, only time is displayed. Once a face is detected in front of the mirror and an active user is elected, guidance on the try-on process is provided to the user. During the try-on the 3D face model of the user is shown and he can interact with the system to change the eyeglasses and control the 3D model rotation.

5.4.2.2 Technology

This subsection discusses the hardware and software technologies used in the building of the smart mirror system.

Hardware As illustrated in Figure 5.18, the smart mirror consists of five main hardware components:

- Display monitor: 27 Inch LCD monitor is used as display set.
- Micro-controller: Raspberry Pi 3B, one of the most popular single-board computer, for the role of the client.
- Camera and Microphone: 720P WebCam and Micro Microphone are employed to fulfill the visual- and audio-based functionalities.
- Mirror: A two-way surface with one reflecting surface is chosen.

- Frame: A solid wood frame box is built to cover internal components, and place the display monitor.

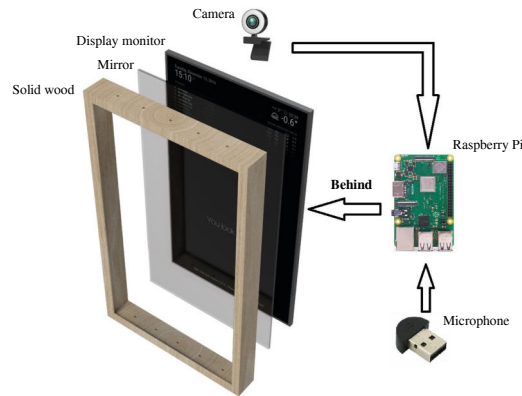


Figure 5.18: Hardware components of the proposed smart mirror.

The microcontroller is connected to the Internet for data fetching and browsing. It is also linked to a serverless scalable service, Amazon Web Service (AWS) Lambda, to host and run the code for the Alexa Skill Service and to an external server to offload the heavy computing and data storage. Micro-controller is Raspberry Pi 3 model B, a single-board based on Linux Operating System. The board has a 64bit CPU at 1.2GHz, 1GB of RAM at 900Mhz, WIFI, and Bluetooth. Raspbian Stretch is the installed operating system. The board offers many I/O ports including a 3.5mm Audio Jack, 4 USB ports, GPIO, LAN, and HDMI. The LCD monitor is connected to the board through the HDMI port. USB ports are instead used to link both the camera and microphone.

By using this hardware and offloading heavy computations, the resulting power consumption of the mirror depends only on the Raspberry and the display monitor. When the mirror is in the idle state, the power consumption is approximately 23W. This consumption is split between 15–20W for the display monitor (that is always on) and 2.6–2.8W for the Raspberry (including connected devices). In the working state, the power consumption increases to about 25W. The consumption of the Raspberry rises to 4.7–4.8W, while that of the display monitor remains unchanged.

Software The smart mirror software is written in Python 3.8, and it is deployed in a Docker container. OpenCV and Pyaudio are exploited to capture the video frames and the audio signal, respectively. The client-server architectural model is implemented in Flask and then deployed using Waitress as Web Server Gateway Interface. Amazon Voice Service (AVS) is a cloud-based service that allows the integration of Alexa’s features into the smart mirror. The interaction between Alexa and the Raspberry is handled by the `avs` library⁶. The library has been modified to send the audio signal to the internal audio interaction module and not just to Alexa. MagicMirror2, one of the most popular open-source DIY smart mirror projects, is used as a starting point for providing the smart mirror with basic utilities. The main software components in the proposed system are the following:

- The **Visual-Audio Manager** is responsible for the visual/audio inputs and outputs, including the interaction with Alexa, recording of audio and capturing of visual frames.
- The **Graphical User Interface Manager** controls the information displayed in the mirror, including the virtual try-on.
- The **Data Processing Manager** handles and computes results from the given visual input.

The relationships between the three software components are shown in Figure 5.19. The Data Processing Manager receives the *raw data* (i.e., the video frames) from the Visual-Audio Manager and returns it the *computed data* (i.e., the virtual try-on result). The Data Processing Manager also interacts with the Database for storing and retrieving data. Finally, the Visual-Audio Manager sends the updates to the Graphic User Interface Manager for displaying information.

The **Visual-Audio Manager** coordinates the interaction with the user and the other software components. It can be considered the core module of the whole system. Figure 5.20 graphically shows its parts and interactions. The main component of the Visual-Audio Manager is the Coordinator. It is

⁶<https://pypi.org/project/avs/>

5.4 Experimenting Mixed Reality catalogs on Smart Mirrors

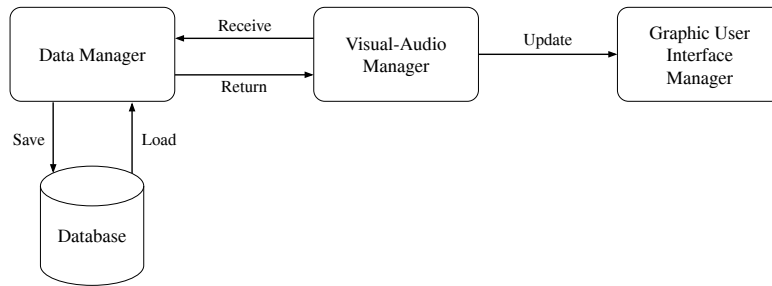


Figure 5.19: A simple diagram showing the relationship between each software component.

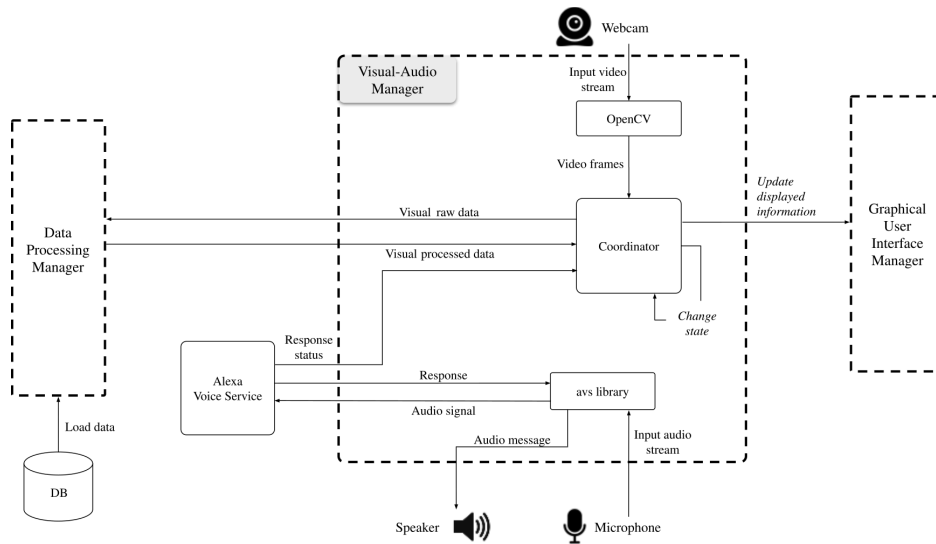


Figure 5.20: The software components and their interactions.

in charge of handling the input-output streams for both the visual and audio signals. In the input data flow, the Coordinator acts as an intermediary for the Data Processing Manager. It receives the video frames acquired with the Webcam, and the audio signal recorded through the Microphone. It then forwards the Visual raw data to the Data Processing Manager which returns the processed data. The Coordinator is also in charge of avoiding overloads and conflicts. This problem can arise due to (i) the availability of only one camera and one microphone, (ii) the presence of asynchronous services (i.e. Alexa Voice Service). Whenever a feature requires one of these devices, the Coordinator changes status to “locked” and blocks access to the resources

until the devices' proprietary function unlocks it. Finally, the Coordinator manages determines which screen (or page) has to be displayed from the Graphical User Interface Manager. The input audio stream is intercepted by the avs library. The avs library involves several components, namely the Voice-engine, the Alexa Voice Service (AVS), and the Alexa skill. Since each Alexa skill is related to the pronunciation of a specific keyphrase, the Voice-engine can exploit the voice acquired with the microphone for user commands or a pre-recorded audio file for automatic commands. The user's voice is used to initialize a conversation with Alexa through keyword detection. This method involves external libraries: Snowboy and Hotword Detection. Thanks to these two libraries it is possible to train a model with a specific keyword for activating Alexa every time the user says that keyword. The audio file, on the other hand, is used to automatically initialize the conversation when a certain system condition occurs. An example of a condition is the detection of the user by the system. Alexa Voice Service acts as an intermediary between the Voice-engine and Alexa Skill. In fact, Alexa Voice Service processes the input audio stream eliminating the silence portions at the beginning and end of the recorded speech, and subsequently receives and plays the response from Alexa Skill. The received audio message is saved as a temporary file and then eliminated by the operating system. Alexa Voice Service also sends a Response status to the Coordinator to manage the change in the status of resources. Finally, Alexa Skill receives the processed audio signal and returns the response. If the key phrase is included in the Alexa Skill list, it will be processed; an exception is thrown otherwise. In the proposed system, Alexa is used to begin, end, and coordinate the eyeglasses virtual try-on sessions.

The **Data Processing Manager** contains most of the smart functions of the mirror and specifically all those relating to the virtual try-on process. These functions are the most computationally expensive. For the visual data, given the video frames, the face detection pipeline is first executed and then the virtual try-on pipeline presented in Section 5.3.3 is executed upon user's request to start a try-on session. The virtual try-on pipeline is embedded in the data processing manager: it is the same used for the web application,

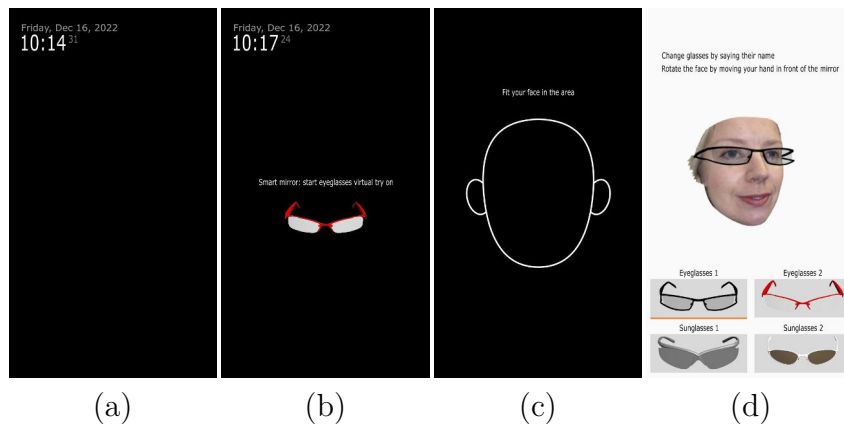


Figure 5.21: Pages that the user can view. (a) Standby. (b) User detected. (c) Try-on session, image acquisition. (d) Try-on session, 3D rendering.

but a few changes are here needed for the user interaction. While the main interaction is done by voice, the rotation of the virtual face model requires visual interaction. Hand detection is performed to rotate the 3D model based on the left-right position of the hand in the frames acquired by the camera.

The **Graphical User Interface Manager** handles the GUI of the smart mirror. Figure 5.21 shows the relevant screens (or pages) displayed to the user during interaction:

- **Standby.** It is displayed when no user is detected in front of the mirror. Whenever this page is active, a simple clock is shown, displaying the current date and time.
- **User detected.** It is shown when a user is standing in front of the mirror. In this page the user can see his face and a message with instructions about how to start the eyeglasses virtual try-on session.
- **Virtual try-on.** It is shown when the person in front of the mirror asks for an eyeglasses virtual try-on session. Two pages are used, in the first one the user is guided to acquire a face picture, in the second one the result of virtual try-on is presented. In this last screen he can change the glasses via voice control and rotate the face moving his hand horizontally in front of the mirror.

5.4.3 Known limitations

This subsection discusses the obtained results and the known limitations of the prototype. This experimental prototype highlighted some issues with both the hardware and the software. Hardware improvement is under investigation to increase the resolution and image quality even with poor lightning. The quality of the frames acquired has a great impact on the quality of the virtual try-on. While the current camera is sufficient to correctly identify the presence of a user and detect hand gestures, the quality and resolution of the face picture have a negative impact on the quality of the 3D model and texture used for the virtual try-on. Also, some problems with the audio interaction have been tracked back to the quality of the microphone which will be replaced with a more performing one. It also appears that the combination of voice and gesture interaction with the mirror is not always intuitive. Better audio and text messages along with an improved interaction workflow could be studied to mitigate such issue. Moreover, a usability study may guide further development and highlight other issues. In addition, it would also be relevant to study the feasibility of superimposing directly the 3D eyeglasses models on the user's face reflected in the mirror. This requires to correctly align the glasses and also to handle occluded portions that should not be rendered. Finally, one of the limitations of the proposed prototype is that it is a proof of concept experimented in a laboratory environment. A future development could be the deployment of the proposed prototype in a real space environment in order to increase the technology-readiness level of the system.

6

Conclusion

This thesis presented a computational framework for Mixed Reality catalogs, several aspects and fields involved in the creation of the aforementioned catalogs are taken into account.

3D geometry reconstruction represents a key factor in the creation of a MR catalog, as it allows to provide integration of 3D objects (being them 3D reconstructed or modeled) with environments (which can be themselves real or 3D reconstructed). SfM Flow, a software toolkit for tackling the problem of selecting the most suitable 3D reconstruction method is proposed. It allows for the creation of synthetic data with pixel-precise ground truth that can be used to simulate and stress 3D reconstruction for specific needs. SfM Flow includes tools for the setup of a 3D virtual scene, the generation of the images, the execution of reconstruction pipelines, and the evaluation of the obtained results in terms of camera poses and geometry accuracy. Moreover, the tools included in SfM Flow can render images with a combination of different lighting and camera effects to simulate critical conditions, stress the reconstruction pipelines, and assess their performances. Blender was chosen as the modeling and rendering software based on its popularity and its extension capabilities; this allowed us to provide an easy-to-use toolkit that encourages its usage and the integration of other functionalities. SfM Flow has been used to create three 3D benchmarking datasets with different characteristics aiming to validate the value of the data, the evaluation procedure, the ability to stress the pipelines, and the quality of the results obtained by

different 3D reconstruction algorithms. IVL-SYNTHSFM and his revision IVL-SYNTHSFM-v2 focus on single-object geometry reconstruction. The ENRICH dataset comprises three sets of data featuring images with different formats, cameras, and environmental and acquisition conditions. Besides the rendered images, the datasets also include GCP coordinates, depth maps, and 3D models as pixel-precise ground truth. The datasets provide challenging data to boost several research activities in photogrammetry and computer vision fields. The variety of data provided is suitable for testing methods and algorithms designed for different application domains, such as remote sensing, photogrammetry, and computer vision. The experiments show that it can be effectively used in several challenging tasks. The experimental results show that it is possible to generate synthetic datasets from which SfM reconstruction can successfully run obtaining satisfactory results. The use of the SfM Flow add-on greatly simplified the data generation and evaluation procedures, also reducing the amount of time and resources usually needed for the evaluation of 3D reconstruction methods under specific conditions. It also allowed taking the pipelines to their limits highlighting critical conditions that can negatively affect the reconstruction process. According to experiments about vanilla SfM pipelines, among the tested incremental SfM implementations, COLMAP showed the best average results. Further work can be done to evaluate other aspects of the pipeline such as reconstructed object coverage to identify missing parts. The evaluation method can also be extended to include the subsequent mesh reconstruction and texture extraction phases. A uniqueness of ENRICH is the availability of multi-scale data, from aerial to terrestrial, enabling the evaluation and comparison of methods and algorithms under various conditions and gathering insights on their strengths and limits. While the datasets proved themselves useful for the evaluation, they can be further extended to include monocular and stereo video sequences, semantic segmentation information, and camera distortions to support the development and evaluation of video-based real-time 3D reconstruction methods, as well as better scene understanding.

The material appearance acquisition of real-world surfaces allows giving virtual elements a realistic look when rendered in a mixed reality context. In

this thesis it is also proposed a device for the acquisition of material representations that can be used to provide a realistic feel of different materials in an MR catalog. This ability also complements the 3D geometry reconstruction. The proposed device has been successfully designed and built to use cheap consumer-grade components. The device built on the photometric stereo technique that was originally presented to recover surface directions for 3D geometry reconstruction has been successfully extended to allow material appearance acquisition of a wide range of materials presenting limited specular reflections. Quantitative and visual evaluations have been performed to assess the quality of the material texture maps recovered. It has been proved that the device can acquire surfaces such as textiles, wood, stones, cork, and plastic and still be able to acquire with the presence of some artifacts highly reflective materials such as metals. Enhancements can be applied to both the hardware and the software of the device. A more performing LED setup could be used to provide stronger lighting on the surface allowing better acquisition of dark materials. A camera with a higher resolution can be used to acquire more micro-detail of the surface in order to have better insights into the surface's roughness. This would also make possible to find a mapping between the actual physical roughness and the roughness values used by material models used for rendering. Finally, the software pipeline could be reviewed to require fewer calibration steps and to integrate concepts of the SfM pipeline to allow the acquisition of larger surfaces.

Rendering and interaction is the most important component of a MR catalog for what concerns how the catalog is perceived and experienced by the final user. In this document different solutions for providing such experience are analyzed, a generic framework was defined and it has been tested with different use case scenarios.

A prototype mobile application for a virtual catalog of textiles has been developed. It allows the final user to see textiles under different lighting conditions. The application is based on the previously defined blueprint design for MR catalogs and uses the proposed device for the acquisition of materials to build the material representation of different textiles. The light intensity and temperature can be changed and the interaction also takes into

account the device orientation to simulate real-time light interactions. The prototype has been evaluated by a pool of users. They appreciated its ability to simulate light interactions with the textiles and found it more useful than static pictures of textiles. In addition, they highlighted some limitations. The missing or wrong specular reflections on some samples were the main reason for their bad evaluation. Further work can be done to provide a better simulation of light interaction and a more accurate appearance of the samples. It would also be interesting to integrate the application to work in an augmented textile store. By adding markers on the different textiles a user would be able to see the real fabric and discover how it would appear under different lighting conditions prior to making the purchase. Another possibility is to use such technology to display a catalog of clothes as suggested by some users during the app evaluation.

In addition, the virtual try-on technology has also been investigated to provide a prototype eyewear virtual try-on based on the generic AR catalog framework. After a critical analysis of existing eyewear virtual try-on applications that aimed to identify their limitations, this thesis introduced the workflow of the virtual try-on system and described the main modules used to provide the try-on experience. Since 3D face reconstruction is a key component of the proposed try-on application, an extensive analysis of the state-of-the-art methods was conducted in order to evaluate their design, complexity, geometry reconstruction errors, and reconstructed texture quality. After such an evaluation, PRNet was selected as the method for 3D face reconstruction in the proposed virtual try-on. The application uses a customized version of the generic AR catalog blueprint to overcome the identified limitations of existing virtual try-on applications for face accessories: first, the eyeglasses virtual try-on is performed in the 3D space using a 3D model of the user's face, reconstructed from a single input picture, allowing the user to interact with the application in a deferred mode and giving the possibility of observing the results from different points of view. Second, the glasses fitting takes into account the actual size of the glasses frames and the actual size of the face of the user, providing a realistic size fitting. The user's face size is estimated with a markerless approach that exploits the iris diameter, thus removing

additional acquisition steps as required by other applications. It was also performed a usability study of the proposed eyewear virtual try-on application. The results show that the application is engaging and easy to use; the users appreciated the full 3D try-on and expressed willingness to use the proposed application if made publicly available. Future works can overcome some limitations observed in the usability study, such as the global quality of the rendering, which should be improved by providing better and complete 3D face reconstruction, and enhanced textures. Another possible extension is that of designing a processing pipeline capable of handling the possible presence of occlusions and self-occlusions in the input picture: while the face geometry reconstruction is possible even in the case of occlusions or self-occlusions, the texture will present missing parts. Restoration techniques such as [30, 58] can be used to recover the missing portions, or the application may ask the user to provide a non-occluded face picture. Finally, the application can take advantage of better integration with online stores and social media services. Providing the try-on experience through a mobile application will also improve the integration with user devices: in this case, the new front-end mobile application can use the existing back-end web service.

Finally, in this thesis it is also proposed the design of a prototype of a smart mirror that is capable of providing an eyeglasses virtual try-on experience. The mirror exploits deep learning techniques to implement the relevant tasks associated with user localization, interaction, and the virtual try-on process. The result has been achieved by including an external server in the design and separating the workload between the onboard device and the server. This led to the possibility of including many features which normally require a lot of computational power while decreasing the computation time at the same time. Also, the interaction with Amazon Alexa is another strength of this prototype. First having a virtual assistant that allows a dialogue really close to the natural language definitely improves the interaction with the mirror, especially for those users that are not used to technology such as elders. Second, exploiting this service further reduces the computational workload since every skill resides on the Amazon Web Service. Additional improvements can be done to solve problems such as image quality (especially in poor lighting

conditions). A more performing pair of microphone and speaker could also be used to improve the overall quality of the vocal interaction. It would also be of great usefulness to perform a usability study of the prototype. The current proof-of-concept could be further developed and strengthened with respect to interaction with users; this would allow us to validate it in an actual store environment instead of a laboratory environment.

Appendices

A

SfM Flow: software architecture

SfM Flow is an add-on for the 3D modeling and rendering software Blender, and thus its architecture is mainly constrained by the coding style of Blender. *SfM Flow* is structured as a python module in which the main components are split into sub-modules. An “operators” sub-module implements the main functionalities as Blender’s operators, and the “ui” sub-module groups the user interface elements, panels, and menus. The “utils” sub-module contains the shared code used across multiple add-on functionalities. The “prefs” sub-module handles user preferences and add-on properties. The “reconstruction” sub-module handles data structures of 3D reconstructions. Finally, an asset folder contains the textures used by the add-on during the initialization step to create the concrete-looking ground of the scene. This folder includes a template flags file used by the Theia [282] 3D reconstruction library and a camera sensor size database used by OpenMVG [197]. The modules organization of the add-on is schematized in Figure A.1.

Some parts of *SfM Flow* can be easily modified to introduce support for specific use cases. The scene setup and initialization logic are implemented in the “SFM_OT_init_scene” operator that can be modified to add support for additional setups. The “SFM_OT_animate_camera” and “SFM_OT_animate_sun” classes respectively define the animations for the camera and the sunlight lamp; these can be also extended to provide additional animation types. These classes are defined in the “operators” sub-module. *SfM Flow* currently supports the N-View Match (NVM) and Bundle (.out)

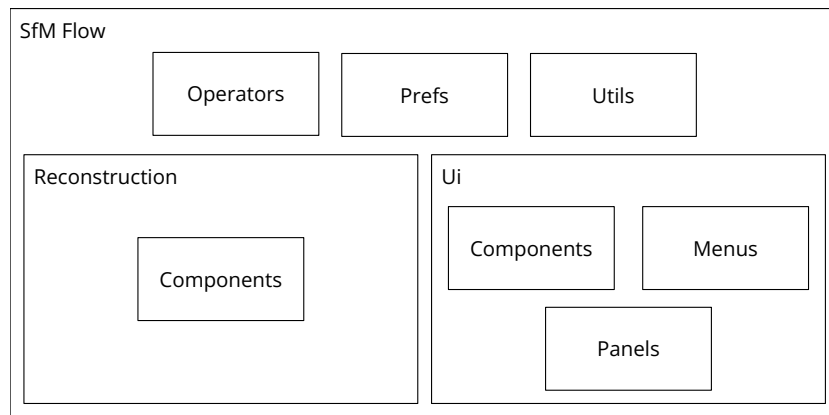


Figure A.1: Simplified modules diagram of SfM Flow.

3D reconstruction exchange formats; other formats can be supported by extending the “ReconstructionBase” class. *SfM Flow* automatically discovers and loads implementations located in the same directory of the base class which is located in the reconstruction sub-module. *SfM Flow* supports token substitution for the custom pipeline commands and the Theia flags template file; both can be extended to support additional tokens in the “replace_tokens” function defined beside the “SFMFLOW_OT_run_pipelines” operator.

B

SfM Flow: illustrative example

Here below are described the steps for a simple use case: a single object is the subject of the dataset, and COLMAP is used to perform the reconstruction.

1. Create a new project and place an object in the scene. Add a camera to acquire the images that will compose the dataset. An example of the initial setup is shown in Figure B.1a.
2. Initialize the scene by adding a ground surface and a lighting setup through the "Initialize current scene" operator (Figure B.1b).
3. Animate the camera to acquire images from different points of view. Figure B.1c it is shown the case of circular animation around the center of the scene. If necessary, also animate the Sun lamp movement.
4. Render the images using the provided operator. EXIF metadata are added to images if supported by the chosen export file format.
5. Run a 3D reconstruction, select COLMAP from the drop-down list, and start it using the "Run 3D reconstruction" operator.
6. Once the reconstruction is ready, import the N-View Match file (NVM) (Figure B.1d) and manually scale and align the reconstructed point cloud to the object(s) in the 3D scene.

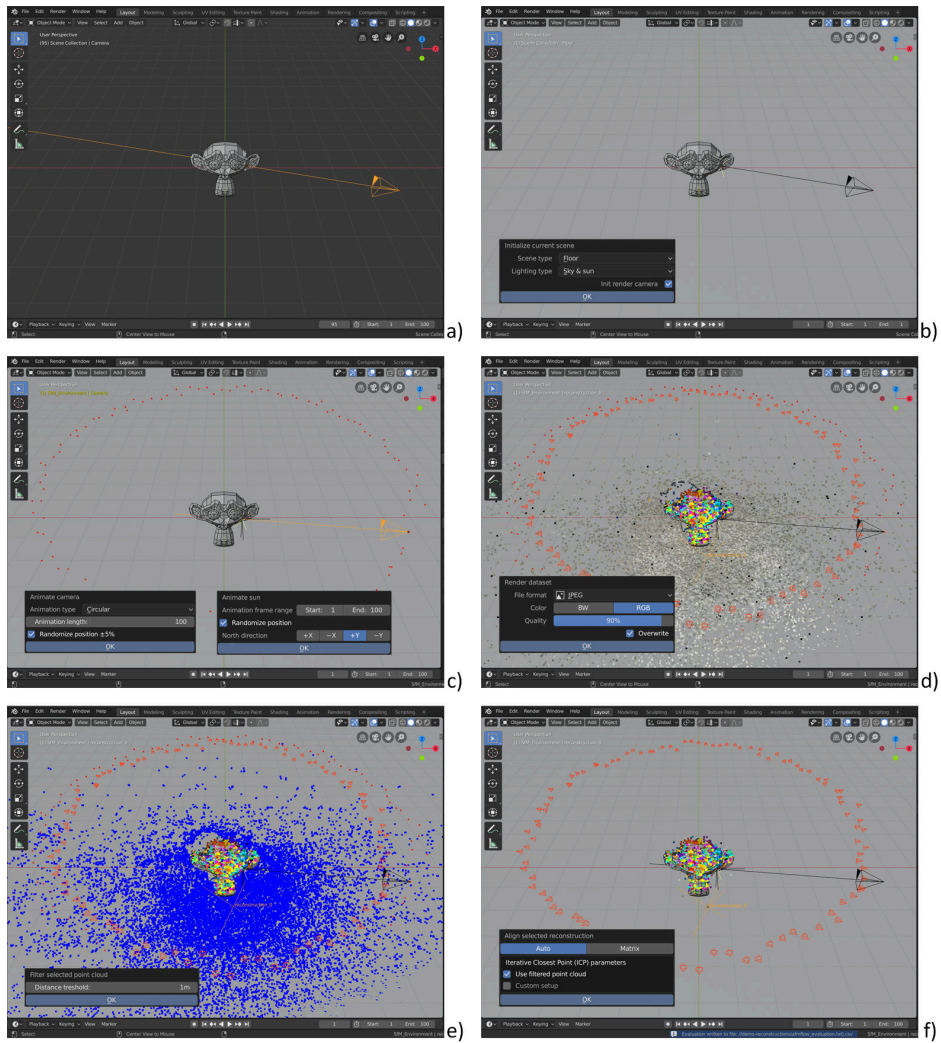


Figure B.1: Illustrative example of the software usage: a) Initial scene; b) 3D scene initialization; c) Camera and sun animation setup; d) Import 3D reconstruction result; e) Filter reconstructed point cloud; f) 3D reconstruction fine alignment.

7. Filter the reconstructed point cloud using the filter functionality (Figure B.1e) to discard points whose distance is beyond a threshold. Discarded points are shown in blue by default.
8. Use the "Align selected reconstruction" to perform fine registration of the point cloud to the virtual scene. This functionality is visible in Figure B.1f. If the registration process is not satisfying, repeat it with a better manual registration/filtering or use different parameters for fine alignment.
9. Evaluate the reconstructed point cloud using the "Evaluate selected reconstruction" operator. Reconstruction evaluation follows the method defined in [32]; results will be written in file *sfmflow_evaluation.txt*. Evaluation dialog and sample results are visible in Figure B.2.

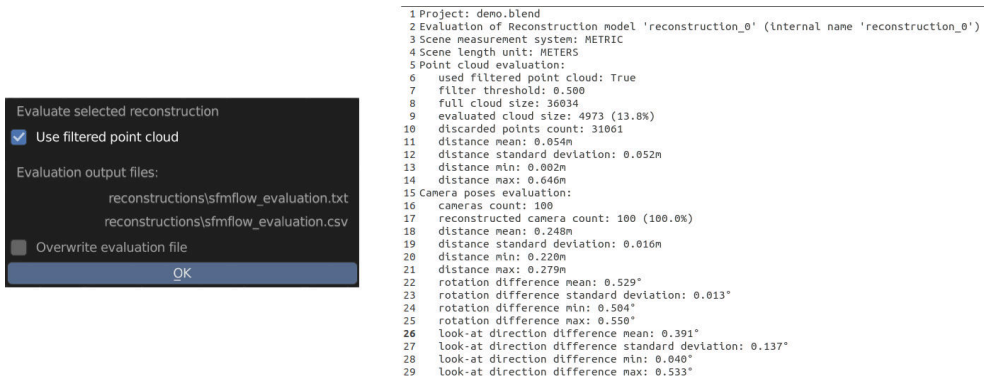


Figure B.2: 3D reconstruction evaluation.

The functionalities of *Sfm Flow* are further explained in Section 3.4.1.

C

Face size estimation algorithm

Algorithm 1 Face size estimation algorithm.

Input: the face image, the keypoints detected on the face, and the estimated distance for the ears in pixels.

Output: the estimated face size as the ear-to-ear distance in millimeters.

```
1: function FACE_SIZE_ESTIMATION(input_img, kpts, ear2ear_pixels)
2:   eye_kpts  $\leftarrow$  fetch_eyes_kpts(kpts)                                 $\triangleright$  left, right eyes
3:   padding  $\leftarrow$  10
4:   px2mm  $\leftarrow$  0
5:   detection_count  $\leftarrow$  0
6:   for eye_kpts in eyes_kpts do
7:     eye_img  $\leftarrow$  crop_image(input_image, eye_kpts, padding)
8:     eye_img  $\leftarrow$  gamma_correction(eye_img)
9:     eye_img  $\leftarrow$  eye_img[:, :, 0]                                     $\triangleright$  use red channel only
10:    mask  $\leftarrow$  drn_d_22_process(eye_img)                             $\triangleright$  predicts the iris segmentation mask
11:    iris  $\leftarrow$  detect_hough_circle(mask)                           $\triangleright$  fits a circle over the segmentation mask
12:    mm  $\leftarrow$  11.71 / (2 * iris.radius)                              $\triangleright$  size of a pixel in millimeters
13:    if mm is not 'nan' then                                          $\triangleright$  estimation successful
14:      px2mm  $\leftarrow$  px2mm + mm
15:      detection_count  $\leftarrow$  detection_count + 1
16:    end if
17:  end for
18:  if detection_count > 0 then
19:    px2mm  $\leftarrow$  px2mm / detection_count                           $\triangleright$  average size of a pixel in millimeters
20:  else
21:    raise Exception('Face size estimation failed!')
22:  end if
23:  return ear2ear_pixels * px2mm                                      $\triangleright$  ear to ear distance in millimeters
24: end function
```

Acknowledgements

This research was supported by grants from NVIDIA and utilized NVIDIA Quadro RTX 6000.

Bibliography

- [1] 3dMD. 3dmdface system. <https://3dmd.com/products/#!/face>, 2020. Accessed on 19 January 2023.
- [2] Aanæs, H., Dahl, A.L. and Steenstrup Pedersen, K. Interesting interest points. *International Journal of Computer Vision*, 97(1):18–35, 2012.
- [3] Ablavatski, A. and Grishchenko, I. Real-time ar self-expression with machine learning. <https://ai.googleblog.com/2019/03/real-time-ar-self-expression-with.html>, 2019. Accessed on 19 January 2023.
- [4] Agarwal, S., Mierle, K. and Others. Ceres solver. <http://ceres-solver.org>, 2012.
- [5] airnodes. Anna smart mirror. <https://www.miroir-anna.com/>, 2016.
- [6] Aittala, M., Weyrich, T. and Lehtinen, J. Practical svbrdf capture in the frequency domain. *ACM Trans. Graph.*, 32(4):110–1, 2013.
- [7] Aittala, M., Weyrich, T., Lehtinen, J. et al. Two-shot svbrdf capture for stationary materials. *ACM Trans. Graph.*, 34(4):110–1, 2015.
- [8] Alcantarilla, P.F. and Solutions, T. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell*, 34(7):1281–1298, 2011.
- [9] Amiri, A.J., Loo, S.Y. and Zhang, H. Semi-supervised monocular depth estimation with left-right consistency using deep neural network. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 602–607. IEEE, 2019.
- [10] Andone, D. and Frydenberg, M. Experiences in online collaborative learning with augmented reality. *eLearning & Software for Education*, 2, 2017.
- [11] Andreu, Y., Chiarugi, F., Colantonio, S., Giannakakis, G. et al. Wize mirror-a smart, multisensory cardio-metabolic risk monitoring system. *Elsevier Computer Vision and Image Understanding*, 148:3–22, 2016.
- [12] Arandjelović, R. and Zisserman, A. Three things everyone should know to improve object retrieval. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2911–2918. IEEE, 2012.
- [13] Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R. et al. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)*, 45(6):891–923, 1998.
- [14] Athira, S., Francis, F., Raphel, R., Sachin, N. et al. Smart mirror: A novel framework for interactive display. In *International Conference on Circuit, Power and Computing*

- Technologies (ICCPCT)*, pages 1–6. IEEE, 2016.
- [15] Bai, Z., Cui, Z., Liu, X. and Tan, P. Riggable 3d face reconstruction via in-network optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6216–6225, June 2021.
- [16] Bakula, K., Mills, J. and Remondino, F. A review of benchmarking in photogrammetry and remote sensing. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2019.
- [17] Barron, J.T. and Malik, J. Shape, albedo, and illumination from a single image of an unknown object. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 334–341. IEEE, 2012.
- [18] Barroso-Laguna, A., Riba, E., Ponsa, D. and Mikolajczyk, K. Key. net: Keypoint detection by handcrafted and learned cnn filters. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5836–5844, 2019.
- [19] Barsky, S. and Petrou, M. The 4-source photometric stereo technique for three-dimensional surfaces in the presence of highlights and shadows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1239–1252, 2003.
- [20] Bay, H., Ess, A., Tuytelaars, T. and Van Gool, L. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.
- [21] Beckmann, P. and Spizzichino, A. The scattering of electromagnetic waves from rough surfaces. *Norwood*, 1987.
- [22] Beddiar, D.R., Nini, B., Sabokrou, M. and Hadid, A. Vision-based human activity recognition: a survey. *Multimedia Tools and Applications*, 79(41):30509–30555, 2020.
- [23] Bederson, B.B. Audio augmented reality: a prototype automated tour guide. In *Conference companion on Human factors in computing systems*, pages 210–211, 1995.
- [24] Bellavia, F., Colombo, C., Morelli, L. and Remondino, F. Challenges in image matching for cultural heritage: an overview and perspective. *Proc. FAPER2022, Springer LNCS*, 2022.
- [25] Bellavia, F., Morelli, L., Menna, F. and Remondino, F. Image orientation with a hybrid pipeline robust to rotations and wide-baselines. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 46:73–80, 2022.
- [26] Bernardini, F., Bajaj, C.L., Chen, J. and Schikore, D.R. Automatic reconstruction of 3d cad models from digital scans. *International Journal of Computational Geometry & Applications*, 9(04n05):327–369, 1999.
- [27] Besl, P.J. and McKay, N.D. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International

- Society for Optics and Photonics, 1992.
- [28] Bhat, S.F., Alhashim, I. and Wonka, P. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021.
- [29] Bhosale, S., Deshmukh, P., Daware, M., Pawar, R. et al. An industrial purpose smart mirror for mood detection. *IJETT*, 6(2), 2019.
- [30] Bianco, S., Ciocca, G., Guarnera, G.C., Scaggianti, A. et al. Scoring recognizability of faces for security applications. In *Image Processing: Machine Vision Applications VII*, volume 9024, page 90240L. SPIE, 2014.
- [31] Bianco, S., Celona, L. and Napoletano, P. Visual-based sentiment logging in magic smart mirrors. In *International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*, pages 1–4. IEEE, 2018.
- [32] Bianco, S., Ciocca, G. and Marelli, D. Evaluating the performance of structure from motion pipelines. *Journal of Imaging*, 4(8), 2018.
- [33] Bianco, S., Celona, L., Ciocca, G., Marelli, D. et al. A smart mirror for emotion monitoring in home environments. *Sensors*, 21(22):7453, 2021.
- [34] Blanz, V., Vetter, T. et al. A morphable model for the synthesis of 3d faces. In *Siggraph*, volume 99, pages 187–194, 1999.
- [35] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. URL <http://www.blender.org>.
- [36] Blinn, J.F. Models of light reflection for computer synthesized pictures. In *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pages 192–198, 1977.
- [37] Bonetti, F., Warnaby, G. and Quinn, L. Augmented reality and virtual reality in physical and online retailing: A review, synthesis and research agenda. *Augmented reality and virtual reality*, pages 119–132, 2018.
- [38] Bottani, E. and Vignali, G. Augmented reality technology in the manufacturing industry: A review of the last decade. *IIEE Transactions*, 51(3):284–310, 2019.
- [39] Brooke, J. et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.
- [40] Brooks Jr, F.P. Walkthrough—a dynamic graphics system for simulating virtual buildings. In *Proceedings of the 1986 workshop on Interactive 3D graphics*, pages 9–21, 1987.
- [41] Brown, D.C. Decentering distortion of lenses. *Photogrammetric Engineering and*

- Remote Sensing*, 1966.
- [42] Bruder, G., Steinicke, F. and Hinrichs, K.H. Arch-explore: A natural user interface for immersive architectural walkthroughs. In *2009 IEEE Symposium on 3D User Interfaces*, pages 75–82. IEEE, 2009.
- [43] Bujnak, M., Kukulova, Z. and Pajdla, T. A general solution to the p4p problem for camera with unknown focal length. In *Computer Vision and Pattern Recognition, 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.
- [44] CareOS. Careos - the first smart health & beauty platform for the bathroom. <https://care-os.com/themis/>, 2021.
- [45] Carlbom, I., Terzopoulos, D. and Harris, K.M. Computer-assisted registration, segmentation, and 3d reconstruction from images of neuronal tissue sections. *IEEE Transactions on medical imaging*, 13(2):351–362, 1994.
- [46] Carman, A. The himirror plus scanned my face and told me i have wrinkles. <https://www.theverge.com/2017/1/4/14166064/himirror-plus-scan-smart-mirror-ces-2017/>, 2017.
- [47] Chaabane, A.M., Sabri, O. and Parguel, B. Competitive advertising within store flyers: a win–win strategy? *Journal of Retailing and Consumer Services*, 17(6): 478–486, 2010.
- [48] Chakra groups inc. Mango mirror. <https://www.mangomirror.com/>, 2018.
- [49] Chatzopoulos, D., Bermejo, C., Huang, Z. and Hui, P. Mobile augmented reality survey: From where we are to where we go. *IEEE Access*, 5:6917–6950, 2017.
- [50] Chen, L. and Heipke, C. Deep learning feature representation for image matching under large viewpoint and viewing direction change. *ISPRS Journal of Photogrammetry and Remote Sensing*, 190:94–112, 2022.
- [51] Chen, W., Fu, Z., Yang, D. and Deng, J. Single-image depth perception in the wild. *Advances in neural information processing systems*, 29, 2016.
- [52] Chen, Y. and Medioni, G. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992.
- [53] Cheng, J., Leng, C., Wu, J., Cui, H. et al. Fast and accurate image matching with cascade hashing for 3d reconstruction. In *2014 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2014.
- [54] Chhugani, J., Purnomo, B., Krishnan, S., Cohen, J. et al. vlod: High-fidelity walkthrough of large virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 11(1):35–47, 2005.
- [55] Chum, O. and Matas, J. Matching with prosac-progressive sample consensus. In

-
- Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 220–226. IEEE, 2005.
- [56] Chum, O., Matas, J. and Kittler, J. Locally optimized ransac. In *Joint Pattern Recognition Symposium*, pages 236–243. Springer, 2003.
- [57] Chum, O., Werner, T. and Matas, J. Two-view geometry estimation unaffected by a dominant plane. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 772–779. IEEE, 2005.
- [58] Colombo, A., Cusano, C. and Schettini, R. Three-dimensional occlusion detection and restoration of partially occluded faces. *Journal of Mathematical Imaging and Vision*, 40(1):105–119, 2011.
- [59] Colombo, A., Cusano, C. and Schettini, R. Umb-db: A database of partially occluded 3d faces. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 2113–2119. IEEE, 2011.
- [60] Cook, R.L. and Torrance, K.E. A reflectance model for computer graphics. *ACM Transactions on Graphics (ToG)*, 1(1):7–24, 1982.
- [61] Crist, R. Griffin technology takes a stab at the smart mirror at ces 2017. <https://www.cnet.com/reviews/griffin-technology-connected-mirror-preview/>, 2017.
- [62] Darrell, T., Gordon, G., Woodfill, J. and Harville, M. A virtual mirror interface using real-time robust face tracking. In *International Conference on Automatic Face and Gesture Recognition*, pages 616–621. IEEE, 1998.
- [63] Deng, Y., Yang, J., Xu, S., Chen, D. et al. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *IEEE Computer Vision and Pattern Recognition Workshops*, 2019.
- [64] Deschaintre, V., Aittala, M., Durand, F., Drettakis, G. et al. Single-image svbrdf capture with a rendering-aware deep network. *ACM Transactions on Graphics (ToG)*, 37(4):1–15, 2018.
- [65] Deschaintre, V., Aittala, M., Durand, F., Drettakis, G. et al. Flexible svbrdf capture with a multi-image deep network. In *Computer graphics forum*, volume 38, pages 1–13. Wiley Online Library, 2019.
- [66] DITTO Technologies. Ditto virtual try-on. <https://ditto.com/virtual-try-on/>, 2021. Accessed on 20 September 2021.
- [67] Drbohlav, O. and Chantler, M. On optimal light configurations in photometric stereo. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1707–1712. IEEE, 2005.
- [68] Du, J., Shi, Y., Mei, C., Quarles, J. et al. Communication by interaction: A

- multiplayer vr environment for building walkthroughs. In *Construction Research Congress*, volume 2016, pages 2281–2290, 2016.
- [69] Dupuy, J. and Jakob, W. An adaptive parameterization for efficient material acquisition and rendering. *ACM Transactions on graphics (TOG)*, 37(6):1–14, 2018.
- [70] Eastlick, M.A., Feinberg, R. and Trappey, C. Information overload in mail catalog shopping: how many catalogs are too many? *Journal of Direct Marketing*, 7(4): 14–19, 1993.
- [71] Eberly, D. Distance between point and triangle in 3d. *Magic Software*, <http://www.magic-software.com/Documentation/pt3tri3.pdf>, 1999.
- [72] Eisert, P., Rurainsky, J. and Fechteler, P. Virtual mirror: Real-time tracking of shoes in augmented reality environments. In *2007 IEEE International Conference on Image Processing*, volume 2, pages II–557. IEEE, 2007.
- [73] Epic Games Inc. Epic games releases realityscan ios app for 3d scanning. <https://www.epicgames.com/site/en-US/news/epic-games-releases-free-realityscan-ios-app-for-3d-scanning>, 2022. Accessed on 19 January 2023.
- [74] Farella, E., Morelli, L., Remondino, F., Mills, J. et al. The eurosdr time benchmark for historical aerial images. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43:1175–1182, 2022.
- [75] Feng, Y., Wu, F., Shao, X., Wang, Y. et al. Joint 3d face reconstruction and dense alignment with position map regression network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 534–551, 2018.
- [76] Feng, Y., Feng, H., Black, M.J. and Bolkart, T. Learning an animatable detailed 3d face model from in-the-wild images. *ACM Trans. Graph.*, 40(4), 2021.
- [77] Filip, J. and Vávra, R. Template-based sampling of anisotropic brdfs. In *Computer Graphics Forum*, volume 33, pages 91–99. Wiley Online Library, 2014.
- [78] Filip, J., Vávra, R. and Havlíček, M. Effective acquisition of dense anisotropic brdf. In *2014 22nd International Conference on Pattern Recognition*, pages 2047–2052. IEEE, 2014.
- [79] Finlayson, G.D., Mackiewicz, M. and Hurlbert, A. Color correction using root-polynomial regression. *IEEE Transactions on Image Processing*, 24(5):1460–1470, 2015.
- [80] Fischler, M.A. and 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):381–395, 1981.

- [81] Fragoso, V., Sen, P., Rodriguez, S. and Turk, M. Evsac: accelerating hypotheses generation by modeling matching scores with extreme value theory. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2472–2479. IEEE, 2013.
- [82] Francken, Y., Cuypers, T., Mertens, T., Gielis, J. et al. High quality mesostructure acquisition using specularities. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7. IEEE, 2008.
- [83] Fuhrmann, S., Langguth, F. and Goesele, M. Mve-a multi-view reconstruction environment. In *GCH*, pages 11–18, 2014.
- [84] Furukawa, Y. and Ponce, J. Accurate, dense, and robust multi-view stereopsis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.
- [85] Furukawa, Y., Hernández, C. et al. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015.
- [86] FXGear Inc. Fxmirror. <http://www.fxmirror.net>, 2017. Accessed on 19 January 2023.
- [87] Gao, D., Li, X., Dong, Y., Peers, P. et al. Deep inverse rendering for high-resolution svbrdf estimation from an arbitrary number of images. *ACM Trans. Graph.*, 38(4): 134–1, 2019.
- [88] Gao, X.S., Hou, X.R., Tang, J. and Cheng, H.F. Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943, 2003.
- [89] García, I.C.A., Salmón, E.R.L., Riega, R.V. and Padilla, A.B. Implementation and customization of a smart mirror through a facial recognition authentication and a personalized news recommendation algorithm. In *International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 35–39. IEEE, 2017.
- [90] Garcia, M. and Oliveira, H. The influence of ground control points configuration and camera calibration for dtm and orthomosaic generation using imagery obtained from a low-cost uav. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 5(1), 2020.
- [91] Garnier, M. and Poncin, I. Do enriched digital catalogues offer compelling experiences, beyond websites? a comparative analysis through the ikea case. *Journal of Retailing and Consumer Services*, 47:361–369, 2019.
- [92] Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F.J. and Marín-Jiménez, M.J. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

- [93] Gartner. Gartner says 100 million consumers will shop in augmented reality online and in-store by 2020. <https://www.gartner.com/en/newsroom/press-releases/2019-04-01-gartner-says-100-million-consumers-will-shop-in-augme>, 2019. Accessed on 19 January 2023.
- [94] Gecer, B., Ploumpis, S., Kotsia, I. and Zafeiriou, S. Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1155–1164, 2019.
- [95] Gecer, B., Ploumpis, S., Kotsia, I. and Zafeiriou, S. Fast-ganfit: Generative adversarial network for high fidelity 3d face reconstruction. *arXiv preprint arXiv:2105.07474*, 2021.
- [96] Gehring, S. and Löchtfeld, M. Using smartphones for customizing products at the point of sale. *International Journal of Mobile Human Computer Interaction (IJMHCI)*, 4(2):59–66, 2012.
- [97] Gehring, S., Löchtfeld, M., Schöning, J., Gorecky, D. et al. Mobile product customization. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, pages 3463–3468. Association for Computing Machinery, 2010.
- [98] Gehrt, K.C. and Carter, K. An exploratory assessment of catalog shopping orientations: the existence of convenience and recreational segments. *Journal of Direct marketing*, 6(1):29–39, 1992.
- [99] Geiger, A., Lenz, P. and Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.
- [100] George, M., Kumar, V. and Grewal, D. Maximizing profits for a multi-category catalog retailer. *Journal of Retailing*, 89(4):374–396, 2013.
- [101] Gerke, M., Nex, F., Remondino, F., Jacobsen, K. et al. Orientation of oblique airborne image sets-experiences from the isprs/eurosdrr benchmark on multi-platform photogrammetry. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 41-B1*, 41:185–191, 2016.
- [102] Ghasemi, Y., Jeong, H., Choi, S.H., Park, K.B. et al. Deep learning-based object detection in augmented reality: A systematic review. *Computers in Industry*, 139:103661, 2022.
- [103] Ghosh, A., Chen, T., Peers, P., Wilson, C.A. et al. Estimating specular roughness and anisotropy from second order spherical gradient illumination. In *Computer Graphics Forum*, volume 28, pages 1161–1170. Wiley Online Library, 2009.
- [104] Ghosh, A., Heidrich, W., Achutha, S. and O’Toole, M. A basis illumination approach to brdf measurement. *International journal of computer vision*, 90(2):183–197, 2010.

- [105] Google. Google cardboard - google vr. <https://arvr.google.com/cardboard/>, 2014. Accessed on 19 January 2023.
- [106] Gorpas, D., Kampouris, C. and Malassiotis, S. Miniature photometric stereo system for textile surface structure reconstruction. In *Videometrics, Range Imaging, and Applications XII; and Automated Visual Inspection*, volume 8791, pages 271–282. SPIE, 2013.
- [107] Gottschalk, S., Yigitbas, E., Schmidt, E. and Engels, G. Model-based product configuration in augmented reality applications. In *International conference on human-centred software engineering*, pages 84–104. Springer, 2020.
- [108] Gross, R., Matthews, I., Cohn, J., Kanade, T. et al. Multi-pie. *Image and Vision Computing*, 28(5):807 – 813, 2010. doi: <https://doi.org/10.1016/j.imavis.2009.08.002>.
- [109] Guo, H., Okura, F., Shi, B., Funatomi, T. et al. Multispectral photometric stereo for spatially-varying spectral reflectances: A well posed problem? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 963–971, 2021.
- [110] Guo, J., Zhu, X., Yang, Y., Yang, F. et al. Towards fast, accurate and stable 3d dense face alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 152–168, 2020.
- [111] Guo, Y., Cai, J., Jiang, B., Zheng, J. et al. Cnn-based real-time dense face reconstruction with inverse-rendered photo-realistic face images. *IEEE transactions on pattern analysis and machine intelligence*, 41(6):1294–1307, 2018.
- [112] Guven, S., Oda, O., Podlaseck, M., Stavropoulos, H. et al. Social mobile augmented reality for retail. In *2009 IEEE International Conference on Pervasive Computing and Communications*, pages 1–3. IEEE, 2009.
- [113] Hall, T., Ciolfi, L., Bannon, L., Fraser, M. et al. The visitor as virtual archaeologist: explorations in mixed reality technology to enhance educational and social interaction in the museum. In *Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage*, pages 91–96, 2001.
- [114] Hamilton, A. and Brown, K. Photogrammetry and star wars battlefront. <https://www.ea.com/frostbite/news/photogrammetry-and-star-wars-battlefront>, 2016. Accessed on 19 January 2023.
- [115] Hammady, R., Ma, M., Strathern, C. and Mohamad, M. Design and development of a spatial mixed reality touring guide to the egyptian museum. *Multimedia Tools and Applications*, 79(5):3465–3494, 2020.
- [116] Han, D.I., tom Dieck, M.C. and Jung, T. User experience model for augmented reality applications in urban heritage tourism. *Journal of Heritage Tourism*, 13(1):

- 46–61, 2018.
- [117] Hanlon, M. Philips homelab creates mirror tv. <https://newatlas.com/philips-homelab-creates-mirror-tv/2003/>, 2004.
- [118] Hanlon, M. Ekko smart mirror puts a wealth of information right in front of your face. <https://www.digitaltrends.com/home/ekko-smart-mirror-integrates-functional-technology-everyday-object/>, 2016.
- [119] Hapticmedia. Functions and benefits of 3d configurators. <https://hapticmedia.com/services/?index=6>, 2021. Accessed on 19 January 2023.
- [120] Hartley, R. and Zisserman, A. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [121] Hartley, R.I. and Sturm, P. Triangulation. *Computer vision and image understanding*, 68(2):146–157, 1997.
- [122] Henriquez, P., Matuszewski, B.J., Andreu-Cabedo, Y., Bastiani, L. et al. Mirror mirror on the wall... an unobtrusive intelligent multisensory mirror for well-being status self-assessment and visualization. *IEEE Transactions on Multimedia*, 19(7): 1467–1481, 2017.
- [123] Hesch, J.A. and Roumeliotis, S.I. A direct least-squares (dls) method for pnp. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 383–390. IEEE, 2011.
- [124] Hoiem, D., Efros, A.A. and Hebert, M. Automatic photo pop-up. In *ACM SIGGRAPH 2005 Papers*, pages 577–584. Association for Computing Machinery, 2005.
- [125] Holz, T., Dragone, M., O’hare, G.M., Martin, A. et al. Mixed reality agents as museum guides. In *ABSHL’06: Agent-Based Systems for Human Learning, AAMAS 2006 Workshop*. ACM Press, 2006.
- [126] Horn, B.K. Closed-form solution of absolute orientation using unit quaternions. *JOSA A*, 4(4):629–642, 1987.
- [127] Huber, P., Kopp, P., Christmas, W., Räscher, M. et al. Real-time 3d face fitting and texture fusion on in-the-wild videos. *IEEE Signal Processing Letters*, 24(4):437–441, 2017.
- [128] Inter IKEA Group. After 70 successful years, ikea is turning the page. <https://about.ikea.com/en/newsroom/2020/12/07/after-70-successful-years-ikea-is-turning-the-page>, 2020. Accessed on 19 January 2023.
- [129] Inter IKEA Systems B.V. Say hej to ikea place. <https://www.ikea.com/au/en/customer-service/mobile-apps/say-hej-to-ikea-place-pub1f8af050>, 2017. Accessed on 19 January 2023.

- [130] International Organization for Standardization. Iso 4287 — geometrical product specifications (gps) - surface texture: Profile method - terms, definitions and surface texture parameters. *Geneve: International Organization for Standardization*, 32: 313–320, 1997.
- [131] Iyer, S.R., Basu, S., Yadav, S., Vijayanand, V.M. et al. Reasonably intelligent mirror. In *International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS)*, pages 302–306. IEEE, 2018.
- [132] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D. et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.
- [133] Jackson, A.S., Bulat, A., Argyriou, V. and Tzimiropoulos, G. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1031–1039, 2017.
- [134] Javornik, A. Augmented reality: Research agenda for studying the impact of its media characteristics on consumer behaviour. *Journal of Retailing and Consumer Services*, 30:252–261, 2016.
- [135] Jeeliz. Jeeliz virtual try-on. <https://github.com/jeeliz/jeelizGlassesVT0Widget>, 2018. Accessed on 19 January 2023.
- [136] Jensen, R., Dahl, A., Vogiatzis, G., Tola, E. et al. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406–413, 2014.
- [137] Jiang, H. and Learned-Miller, E. Face detection with the faster r-cnn. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 650–657. IEEE, 2017.
- [138] Jin, K., Deng, X., Huang, Z. and Chen, S. Design of the smart mirror based on raspberry pi. In *Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pages 1919–1923. IEEE, 2018.
- [139] Jin, Y., Mishkin, D., Mishchuk, A., Matas, J. et al. Image matching across wide baselines: From paper to practice. *International Journal of Computer Vision*, 129 (2):517–547, 2021.
- [140] Kajiya, J.T. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, 1986.
- [141] Kampouris, C., Zafeiriou, S., Ghosh, A. and Malassiotis, S. Fine-grained material classification using micro-geometry and reflectance. In *European Conference on Computer Vision*, pages 778–792. Springer, 2016.

- [142] Kang, H.J., Shin, J.h. and Ponto, K. How 3d virtual reality stores can shape consumer purchase decisions: the roles of informativeness and playfulness. *Journal of Interactive Marketing*, 49:70–85, 2020.
- [143] Karras, T., Laine, S. and Aila, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019.
- [144] Kawale, J. and Chaudhari, P. Iot based design of intelligent mirror using raspberry pi. In *International Conference for Convergence in Technology (I2CT)*, pages 1–4. IEEE, 2019.
- [145] Kaya, M. and Ünay, D. Dressboard: An embedded virtual try-on system for ties and bowties. *Journal of Signal Processing Systems*, 73(2):143–152, 2013.
- [146] Kerrigan, D., Trokielewicz, M., Czajka, A. and Bowyer, K.W. Iris recognition with image segmentation employing retrained off-the-shelf deep neural networks. In *2019 International Conference on Biometrics (ICB)*, pages 1–7. IEEE, 2019.
- [147] Kim, J. and Forsythe, S. Adoption of virtual try-on technology for online apparel shopping. *Journal of Interactive Marketing*, 22(2):45–59, 2008.
- [148] Kim, Y.g. and Kim, W.j. Implementation of augmented reality system for smartphone advertisements. *international journal of multimedia and ubiquitous engineering*, 9(2): 385–392, 2014.
- [149] King, D.E. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.
- [150] Knapitsch, A., Park, J., Zhou, Q.Y. and Koltun, V. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4): 1–13, 2017.
- [151] Kukeleva, Z., Bujnak, M. and Pajdla, T. Real-time solution to the absolute pose problem with unknown radial distortion and focal length. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2816–2823. IEEE, 2013.
- [152] Lafortune, E.P., Foo, S.C., Torrance, K.E. and Greenberg, D.P. Non-linear approximation of reflectance functions. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 117–126, 1997.
- [153] Lambert, J. *Photometria sive de mensura de gradibus luminis, colorum et umbrae*. Augustae Vindelicorum, 1760.
- [154] Launceston City Council. City of Launceston - 3D Scan Atlas. <https://s3-ap-southeast-2.amazonaws.com/launceston/atlas/index.html>, 2013. Accessed on 19 January 2023.

- [155] Lee, J.H., Han, M.K., Ko, D.W. and Suh, I.H. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019.
- [156] Lepetit, V., Moreno-Noguer, F. and Fua, P. Epanp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009.
- [157] Lévy, P. *Cyberculture: rapport au conseil de l'Europe*. Odile Jacob, 1997.
- [158] Ley, A., Hänsch, R. and Hellwich, O. Syb3r: A realistic synthetic benchmark for 3d reconstruction from images. In *European Conference on Computer Vision*, pages 236–251. Springer, 2016.
- [159] Li, M., Zhou, Z., Wu, Z., Shi, B. et al. Multi-view photometric stereo: A robust solution and benchmark dataset for spatially varying isotropic materials. *IEEE Transactions on Image Processing*, 29:4159–4173, 2020.
- [160] Li, T., Bolkart, T., Black, M.J., Li, H. et al. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017.
- [161] Li, X., Dong, Y., Peers, P. and Tong, X. Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. *ACM Transactions on Graphics (ToG)*, 36(4):1–11, 2017.
- [162] Li, Z. and Snavely, N. Megadepth: Learning single-view depth prediction from internet photos. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2041–2050, 2018.
- [163] Li, Z., Xu, Z., Ramamoorthi, R., Sunkavalli, K. et al. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018.
- [164] Liarokapis, F. and White, M. Augmented reality techniques for museum environments. *Mediterranean Journal of Computers and Networks*, 1(2):95–102, 2005.
- [165] Lin, J., Yuan, Y., Shao, T. and Zhou, K. Towards high-fidelity 3d face reconstruction from in-the-wild images using graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5891–5900, 2020.
- [166] Lindstrom, P. Triangulation made easy. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1554–1561. IEEE, 2010.
- [167] Liu, C., Narasimhan, S.G. and Dubrawski, A.W. Near-light photometric stereo using circularly placed point light sources. In *2018 IEEE International Conference on Computational Photography (ICCP)*, pages 1–10. IEEE, 2018.

- [168] Liu, C., Yang, J., Ceylan, D., Yumer, E. et al. Planenet: Piece-wise planar reconstruction from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2579–2588, 2018.
- [169] Liu, Y., Jia, J., Fu, J., Ma, Y. et al. Magic mirror: A virtual fashion consultant. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 680–683, 2016.
- [170] Liu, Z., Zhu, J., Bu, J. and Chen, C. A survey of human pose estimation: the body parts parsing based methods. *Journal of Visual Communication and Image Representation*, 32:10–19, 2015.
- [171] Löchtefeld, M., Schöning, J., Rohs, M. and Krüger, A. Marauders light: replacing the wand with a mobile camera projector unit. In *Proceedings of the 8th International Conference on Mobile and Ubiquitous Multimedia*, pages 1–4, 2009.
- [172] Löchtefeld, M., Gehring, S., Schöning, J. and Krüger, A. Shelftorchlight: Augmenting a shelf using a camera projector unit. In *Adjunct Proceedings of the Eighth International Conference on Pervasive Computing*, pages 1–4, 2010.
- [173] Löchtefeld, M., Böhmer, M., Daiber, F. and Gehring, S. Augmented reality-based advertising strategies for paper leaflets. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 1015–1022, 2013.
- [174] Logothetis, F., Mecca, R. and Cipolla, R. Semi-calibrated near field photometric stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 941–950, 2017.
- [175] Lourakis, M. and Argyros, A. The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Technical report, Technical Report 340, Institute of Computer Science-FORTH, Heraklion, Crete, Greece, 2004.
- [176] Lowe, D.G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [177] Luxottica Group. Virtual mirror. <http://www.luxottica.com/en/virtual-mirror-technology-arrives-valentinocom>, 2020. Accessed on 19 January 2023.
- [178] Mack, E. Toshiba’s smart mirror concept puts the future on display. <https://newatlas.com/toshiba-smart-mirror-concept-ces-2014/30574/>, 2014.
- [179] Maiwald, F., Lehmann, C. and Lazariv, T. Fully automated pose estimation of historical images in the context of 4d geographic information systems utilizing machine learning methods. *ISPRS International Journal of Geo-Information*, 10(11): 748, 2021.

- [180] Malzbender, T., Wilburn, B., Gelb, D. and Ambrisco, B. Surface enhancement using real-time photometric stereo and reflectance transformation. *Rendering techniques*, 2006:245–250, 2006.
- [181] Marelli, D., Bianco, S. and Ciocca, G. Ivl-synthsfm-v2: A synthetic dataset with exact ground truth for the evaluation of 3d reconstruction pipelines. *Data in Brief*, page 105041, 2020.
- [182] Marelli, D., Bianco, S. and Ciocca, G. Sfm flow: A comprehensive toolset for the evaluation of 3d reconstruction pipelines. *SoftwareX*, 17:100931, 2022.
- [183] Marelli, D., Morelli, L., Farella, E.M., Bianco, S. et al. Enrich: multi-purpose datasets for benchmarking in computer vision and photogrammetry. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2023. (minor review).
- [184] Marschner, S.R., Westin, S.H., Lafortune, E.P., Torrance, K.E. et al. Image-based brdf measurement including human skin. In *Eurographics Workshop on Rendering Techniques*, pages 131–144. Springer, 1999.
- [185] Mase, K., Kadobayashi, R. and Nakatsu, R. Meta-museum: A supportive augmented-reality environment for knowledge sharing. In *ATR workshop on social agents: humans and machines*, pages 107–110. CiteSeerX PA, USA, 1996.
- [186] Matusik, W. *A data-driven reflectance model*. PhD thesis, Massachusetts Institute of Technology, 2003.
- [187] Max-Planck-Gesellschaft. NoW Challenge. <https://now.is.tue.mpg.de/>, 2020. Accessed on 19 January 2023.
- [188] Meagher, D.J. *Octree encoding: A new technique for the representation, manipulation and display of arbitrary 3-d objects by computer*. Electrical and Systems Engineering Department Rensselaer Polytechnic Institute Image Processing Laboratory, 1980.
- [189] Mehta, D., Sridhar, S., Sotnychenko, O., Rhodin, H. et al. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017.
- [190] MemoMi. Memory mirror. <https://memorymirror.com/>, 2015. Accessed on 19 January 2023.
- [191] Milgram, P. and Kishino, F. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.
- [192] Mishkin, D., Radenovic, F. and Matas, J. Repeatability is not enough: Learning affine regions via discriminability. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 284–300, 2018.
- [193] Miyashita, T., Meier, P., Tachikawa, T., Orlic, S. et al. An augmented reality museum

- guide. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 103–106. IEEE, 2008.
- [194] Mohamed, A.S.A., Wahab, M.A., Suhaily, S. and Arasu, D.B.L. Smart mirror design powered by raspberry pi. In *Artificial Intelligence and Cloud Computing Conference*, pages 166–173, 2018.
- [195] Moisan, L., Moulon, P. and Monasse, P. Automatic homographic registration of a pair of images, with a contrario elimination of outliers. *Image Processing On Line*, 2: 56–73, 2012.
- [196] Moons, T., Van Gool, L., Vergauwen, M. et al. 3d reconstruction from multiple images part 1: Principles. *Foundations and trends® in Computer Graphics and Vision*, 4(4):287–404, 2010.
- [197] Moulon, P., Monasse, P., Marlet, R. and Others. OpenMVG. an open multiple view geometry library. <https://github.com/openMVG/openMVG>, 2013.
- [198] Muja, M. and Lowe, D. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP 2009 - Proceedings of the 4th International Conference on Computer Vision Theory and Applications*, volume 1, pages 331–340, 01 2009.
- [199] Mukaigawa, Y., Sumino, K. and Yagi, Y. Multiplexed illumination for measuring brdf using an ellipsoidal mirror and a projector. In *Asian Conference on Computer Vision*, pages 246–257. Springer, 2007.
- [200] Muratov, O., Slynko, Y., Chernov, V., Lyubimtseva, M. et al. 3dcapture: 3d reconstruction for a smartphone. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 75–82, 2016.
- [201] Naik, N., Zhao, S., Velten, A., Raskar, R. et al. Single view reflectance capture using multiplexed scattering and time-of-flight imaging. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 1–10, 2011.
- [202] Nelder, J.A. and Mead, R. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.
- [203] Ngan, A., Durand, F. and Matusik, W. Experimental analysis of brdf models. *Rendering Techniques*, 2005(16th):2, 2005.
- [204] Nguyen, D.T., Li, W. and Ogunbona, P.O. Human detection from images and videos: A survey. *Pattern Recognition*, 51:148–175, 2016.
- [205] Nguyen, T.V. and Liu, L. Smart mirror: Intelligent makeup recommendation and synthesis. In *International Conference on Multimedia*, pages 1253–1254. ACM, 2017.
- [206] Nicodemus, F.E. Directional reflectance and emissivity of an opaque surface. *Applied optics*, 4(7):767–775, 1965.

- [207] Nielsen, J. and Landauer, T.K. A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 206–213, 1993.
- [208] Nikolenko, S.I. et al. *Synthetic data for deep learning*. Springer, 2021.
- [209] Nocerino, E., Lago, F., Morabito, D., Remondino, F. et al. A smartphone-based 3d pipeline for the creative industry-the replicate eu project. *3D Virtual Reconstruction and Visualization of Complex Architectures*, 42:535–541, 2017.
- [210] Noh, Z., Sunar, M.S. and Pan, Z. *A Review on Augmented Reality for Virtual Heritage System*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-03364-3.
- [211] Olson, J.L., Krum, D.M., Suma, E.A. and Bolas, M. A design for a smartphone-based head mounted display. In *2011 IEEE Virtual Reality Conference*, pages 233–234. IEEE, 2011.
- [212] Oniga, V.E., Breaban, A.I., Pfeifer, N. and Chirila, C. Determining the suitable number of ground control points for uas images georeferencing by varying number and spatial distribution. *Remote Sensing*, 12(5):876, 2020.
- [213] ONNX Runtime developers. ONNX Runtime, 11 2018. URL <https://github.com/microsoft/onnxruntime>.
- [214] Ono, Y., Trulls, E., Fua, P. and Yi, K.M. Lf-net: Learning local features from images. *Advances in neural information processing systems*, 31, 2018.
- [215] OpenCV. Open source computer vision library, 2015.
- [216] Oren, M. and Nayar, S.K. Generalization of lambert’s reflectance model. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 239–246, 1994.
- [217] Ostrowski, W. and Bakula, K. Towards efficiency of oblique images orientation. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-3/W4:91–96, 2016.
- [218] Ouellet, E., Boller, B., Corriveau-Lecavalier, N., Cloutier, S. et al. The virtual shop: A new immersive virtual reality environment and scenario for the assessment of everyday memory. *Journal of neuroscience methods*, 303:126–135, 2018.
- [219] Özdemir, E., Toschi, I. and Remondino, F. A multi-purpose benchmark for photogrammetric urban 3d reconstruction in a controlled environment. In *Evaluation and Benchmarking Sensors, Systems and Geospatial Data in Photogrammetry and Remote Sensing*, volume 42, pages 53–60, 2019.
- [220] Panasonic. Digital concierge - advanced smart mirror with ibm watson. <https://>

- [//channel.panasonic.com/contents/19698/](https://channel.panasonic.com/contents/19698/), 2017.
- [221] Pantano, E., Rese, A. and Baier, D. Enhancing the online decision-making process by using augmented reality: A two country comparison of youth markets. *Journal of Retailing and Consumer Services*, 38:81–95, 2017.
- [222] Parihar, U.S., Gujarathi, A., Mehta, K., Tourani, S. et al. Rord: Rotation-robust descriptors and orthographic views for local feature matching. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1593–1600. IEEE, 2021.
- [223] Park, Y., Lepetit, V. and Woo, W. Multiple 3d object tracking for augmented reality. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 117–120. IEEE, 2008.
- [224] Paysan, P., Knothe, R., Amberg, B., Romdhani, S. et al. A 3d face model for pose and illumination invariant face recognition. In *2009 sixth IEEE international conference on advanced video and signal based surveillance*, pages 296–301. IEEE, 2009.
- [225] Peppas, M., Morelli, L., Mills, J., Penna, N. et al. Handcrafted and learning-based tie point features—comparison using the euroSDR rPAS benchmark dataset. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43:1183–1190, 2022.
- [226] Perfect Corp. Youcam makeup. <https://www.perfectcorp.com/app/ykc>, 2014. Accessed on 19 January 2023.
- [227] Perfect Corp. Makeup ar. <https://plugins.makeupar.com>, 2015. Accessed on 19 January 2023.
- [228] Phong, B.T. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.
- [229] Pietraschke, M. and Blanz, V. Automated 3d face reconstruction from multiple images using quality measures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3418–3427, 2016.
- [230] Plata, C., Nieves, J.L., Valero, E.M. and Romero, J. Trichromatic red-green-blue camera used for the recovery of albedo and reflectance of rough-textured surfaces under different illumination conditions. *Applied Optics*, 48(19):3643–3653, 2009.
- [231] Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H. et al. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5): 1–36, 2018.
- [232] Poznanski, A. Visual revolution of the vanishing of ethan carter.

- <http://www.theastronauts.com/2014/03/visual-revolution-vanishing-ethan-carter/>, 2014. Accessed on 19 January 2023.
- [233] Pultar, M. Improving the hardnet descriptor. *arXiv preprint arXiv:2007.09699*, 2020.
- [234] Qiu, W., Zhong, F., Zhang, Y., Qiao, S. et al. Unrealcv: Virtual worlds for computer vision. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1221–1224, 2017.
- [235] Raguram, R., Frahm, J.M. and Pollefeys, M. A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. In *European Conference on Computer Vision*, pages 500–513. Springer, 2008.
- [236] Ranftl, R., Lasinger, K., Hafner, D., Schindler, K. et al. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 2020.
- [237] Ranftl, R., Bochkovskiy, A. and Koltun, V. Vision transformers for dense prediction. *ArXiv preprint*, 2021.
- [238] Ranjan, A., Bolkart, T., Sanyal, S. and Black, M.J. Generating 3d faces using convolutional mesh autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 704–720, 2018.
- [239] Rashid, Z., Peig, E. and Pous, R. Bringing online shopping experience to offline retail through augmented reality and rfid. In *2015 5th International Conference on the Internet of Things (IOT)*, pages 45–51. IEEE, 2015.
- [240] Rasouli, A. Deep learning for vision-based prediction: A survey. *arXiv preprint arXiv:2007.00095*, 2020.
- [241] Remondino, F. Heritage recording and 3d modeling with photogrammetry and 3d scanning. *Remote Sensing*, 3(6):1104–1138, 2011.
- [242] Remondino, F., Menna, F. and Morelli, L. Evaluating hand-crafted and learning-based features for photogrammetric applications. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43:549–556, 2021.
- [243] Remondino, F., Morelli, L., Stathopoulou, E., Elhashash, M. et al. Aerial triangulation with learning-based tie points. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43:77–84, 2022.
- [244] Riviere, J., Peers, P. and Ghosh, A. Mobile surface reflectometry. In *ACM SIGGRAPH 2014 Posters*, pages 1–1. Association for Computing Machinery, 2014.
- [245] Roth, J., Tong, Y. and Liu, X. Adaptive 3d face reconstruction from unconstrained photo collections. In *Proceedings of the IEEE Conference on Computer Vision and*

- Pattern Recognition*, pages 4197–4206, 2016.
- [246] Rousseeuw, P.J. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984.
- [247] Rüfer, F., Schröder, A. and Erb, C. White-to-white corneal diameter: normal values in healthy humans obtained with the orbscan ii topography system. *Cornea*, 24(3): 259–261, 2005.
- [248] Rump, M., Müller, G., Sarlette, R., Koch, D. et al. Photo-realistic rendering of metallic car paint from image-based measurements. In *Computer Graphics Forum*, volume 27, pages 527–536. Wiley Online Library, 2008.
- [249] Rupnik, E., Nex, F., Toschi, I. and Remondino, F. Aerial multi-camera systems: Accuracy and block triangulation issues. *ISPRS Journal of Photogrammetry and Remote Sensing*, 101:233–246, 2015.
- [250] Saakes, D., Yeo, H.S., Noh, S.T., Han, G. et al. Mirror mirror: An on-body t-shirt design system. In *CHI Conference on Human Factors in Computing Systems*, pages 6058–6063, 2016.
- [251] Safilo. Virtualeyes. <https://www.uqido.com/progetti/safilo-virtualeyes/>, 2020. Accessed on 19 January 2023.
- [252] Sanna, A., Manuri, F., Lamberti, F., Paravati, G. et al. Using handheld devices to support augmented reality-based maintenance and assembly tasks. In *2015 IEEE international conference on consumer electronics (ICCE)*, pages 178–179. IEEE, 2015.
- [253] Sanyal, S., Bolkart, T., Feng, H. and Black, M. Learning to regress 3d face shape and expression from an image without 3d supervision. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 7763–7772, June 2019.
- [254] Saurer, O., Fraundorfer, F. and Pollefeys, M. Omnitour: Semi-automatic generation of interactive virtual tours from omnidirectional video. In *Proc. 3DPVT2010 (Int. Symp. on 3D Data Processing, Visualization and Transmission)*, 2010.
- [255] Sauro, J. Measuring usability with the system usability scale (sus), 2011. URL <http://www.measuringusability.com/sus.php>.
- [256] Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G. et al. High-resolution stereo datasets with subpixel-accurate ground truth. In *German conference on pattern recognition*, pages 31–42. Springer, 2014.
- [257] Schlick, C. An inexpensive brdf model for physically-based rendering. In *Computer graphics forum*, volume 13, pages 233–246. Wiley Online Library, 1994.
- [258] Schmitt, C., Donne, S., Riegler, G., Koltun, V. et al. On joint estimation of pose, geometry and svbrdf from a handheld scanner. In *Proceedings of the IEEE/CVF*

-
- Conference on Computer Vision and Pattern Recognition*, pages 3493–3503, 2020.
- [259] Schonberger, J.L. and Frahm, J.M. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.
- [260] Schönberger, J.L., Price, T., Sattler, T., Frahm, J.M. et al. A vote-and-verify strategy for fast spatial verification in image retrieval. In *Asian Conference on Computer Vision*, pages 321–337. Springer, 2016.
- [261] Schonberger, J.L., Hardmeier, H., Sattler, T. and Pollefeys, M. Comparative evaluation of hand-crafted and learned local features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1482–1491, 2017.
- [262] Schops, T., Schonberger, J.L., Galliani, S., Sattler, T. et al. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3260–3269, 2017.
- [263] Seitz, S.M., Curless, B., Diebel, J., Scharstein, D. et al. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 1, pages 519–528. IEEE, 2006.
- [264] Sethukkarasi, C., HariKrishnan, V., PalAmutha, K. and Pitchian, R. Interactive mirror for smart home. *International Journal on Advances in Intelligent Systems*, 9 (1 & 2):148–160, 2016.
- [265] Sharma, G., Wu, W. and Dalal, E.N. The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur*, 30(1):21–30, 2005.
- [266] Shenzhen Normand Electronic Co.,Ltd. Sk6812rgbw datasheet. <https://www.normandle.com/upload/201805/SK6812RGBX-XX%20Datasheet.pdf>, 2019. Accessed on 19 January 2023.
- [267] Shih, Y.C. A virtual walk through london: Culture learning through a cultural immersion experience. *Computer Assisted Language Learning*, 28(5):407–428, 2015.
- [268] Silberman, N., Hoiem, D., Kohli, P. and Fergus, R. Indoor segmentation and support inference from rgb-d images. In *European conference on computer vision*, pages 746–760. Springer, 2012.
- [269] Sinthanayothin, C., Wongwean, N. and Bholsithi, W. Interactive virtual 3d gallery

- using motion detection of mobile device. In *International Conference on Mobile IT Convergence*, pages 120–125. IEEE, 2011.
- [270] Snavely, N., Seitz, S.M. and Szeliski, R. Photo tourism: exploring photo collections in 3d. In *ACM transactions on graphics (TOG)*, volume 25, pages 835–846. ACM, 2006.
- [271] Soga, A. Virtual show, go in!: walk-through system and vr goggles of a temple for museum exhibits. In *2015 International Conference on Culture and Computing (Culture Computing)*, pages 199–200. IEEE, 2015.
- [272] Sony Corporation. Sony imx219pqh5-c datasheet. https://github.com/rellimmot/Sony-IMX219-Raspberry-Pi-V2-CMOS/blob/bb4a45eaad8b433c2f29aaa9c06592b4efd7552f/RASPBERRY%20PI%20CAMERA%20V2%20DATASHEET%20IMX219PQH5_7.0.0_Datasheet_XXX.PDF, 2014. Accessed on 19 January 2023.
- [273] Spence, A. and Chantler, M. Optimal illumination for three-image photometric stereo acquisition of texture. In *Proceedings of the 3rd International Workshop on Texture Analysis and Synthesis*, pages 89–94. Citeseer, 2003.
- [274] Starck, J. and Hilton, A. Surface capture for performance-based animation. *IEEE computer graphics and applications*, 27(3), 2007.
- [275] Stathopoulou, E.K. and Remondino, F. Open-source image-based 3d reconstruction pipelines: Review, comparison and evaluation. In *6th International Workshop LowCost 3D-Sensors, Algorithms, Applications*, pages 331–338, 2019.
- [276] Statista. Retail e-commerce sales worldwide from 2014 to 2024. <https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/>, 2020. Accessed on 19 January 2023.
- [277] Stewenius, H., Engels, C. and Nistér, D. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4):284–294, 2006.
- [278] Strecha, C., Von Hansen, W., Van Gool, L., Fua, P. et al. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. Ieee, 2008.
- [279] Sun, Y., Wang, X. and Tang, X. Deep convolutional network cascade for facial point detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3476–3483, 2013.
- [280] Sutherland, I.E. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 757–764, 1968.

- [281] Suzuki, S. et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985.
- [282] Sweeney, C. Theia multiview geometry library: Tutorial & reference. <http://theia-sfm.org>, 2015.
- [283] Tater, L., Pranjale, S., Lade, S., Nimbalkar, A. et al. Iot based assistive smart mirror with human emotion recognition system. *International Journal of Engineering Research & Technology (IJERT)*, 9(2), 2020.
- [284] Tefera, Y., Poiesi, F., Morabito, D., Remondino, F. et al. 3dnow: Image-based 3d reconstruction and modeling via web. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 42(2), 2018.
- [285] Thomas, B., Close, B., Donoghue, J., Squires, J. et al. Arquake: An outdoor/indoor augmented reality first person application. In *Digest of Papers. Fourth International Symposium on Wearable Computers*, pages 139–146. IEEE, 2000.
- [286] Thomas, P.C. and David, W. Augmented reality: An application of heads-up display technology to manual manufacturing processes. In *Hawaii international conference on system sciences*, volume 2. ACM SIGCHI Bulletin, 1992.
- [287] Torrance, K.E. and Sparrow, E.M. Off-specular peaks in the directional distribution of reflected thermal radiation. *Journal of Heat Transfer*, 1966.
- [288] Torrance, K.E. and Sparrow, E.M. Theory for off-specular reflection from roughened surfaces. *Josa*, 57(9):1105–1114, 1967.
- [289] Tosi, F., Aleotti, F., Poggi, M. and Mattoccia, S. Learning monocular depth estimation infusing traditional stereo knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9799–9809, 2019.
- [290] Tran, A.T., Hassner, T., Masi, I., Paz, E. et al. Extreme 3D face reconstruction: Seeing through occlusions. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3935–3944, 2018.
- [291] Tremblay, J., Prakash, A., Acuna, D., Brophy, M. et al. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 969–977, 2018.
- [292] Triggs, B., McLauchlan, P.F., Hartley, R.I. and Fitzgibbon, A.W. Bundle adjustment - a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.
- [293] Tsunashima, H., Arase, K., Lam, A. and Kataoka, H. Uvirt—unsupervised virtual try-on using disentangled clothing and person features. *Sensors*, 20(19):5647, Oct

- 2020.
- [294] Tuan Tran, A., Hassner, T., Masi, I. and Medioni, G. Regressing robust and discriminative 3d morphable models with a very deep neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5163–5172, 2017.
- [295] Uchiyama, H. and Marchand, E. Object detection and pose tracking for augmented reality: Recent approaches. In *18th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, 2012.
- [296] Ulvi, A. The effect of the distribution and numbers of ground control points on the precision of producing orthophoto maps with an unmanned aerial vehicle. *Journal of Asian Architecture and Building Engineering*, 20(6):806–817, 2021.
- [297] United States. National Bureau of Standards and Nicodemus, Fred Edwin. *Geometrical considerations and nomenclature for reflectance*, volume 160. Citeseer, 1977.
- [298] Unity Technologies. Unity real-time development platform. <https://unity.com/>, 2005. Accessed on 19 January 2023.
- [299] Unity Technologies. Unity Perception package. <https://github.com/Unity-Technologies/com.unity.perception>, 2020.
- [300] Vasiljevic, I., Kolkin, N., Zhang, S., Luo, R. et al. DIODE: A Dense Indoor and Outdoor DEpth Dataset. *CoRR*, abs/1908.00463, 2019.
- [301] Villanueva, J. and Blanco, A. Optimization of ground control point (gcp) configuration for unmanned aerial vehicle (uav) survey using structure from motion (sfm). *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:167–174, 2019.
- [302] Viola, P. and Jones, M.J. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.
- [303] Vlahakis, V., Ioannidis, M., Karigiannis, J., Tsotros, M. et al. Archeoguide: an augmented reality guide for archaeological sites. *IEEE Computer Graphics and Applications*, 22(5):52–60, 2002.
- [304] Voir. Digital makeup. <https://voir.me/>, 2018. Accessed on 19 January 2023.
- [305] Von Hollen, S. and Reeh, B. Smart mirror devices. In *International Conference on Innovations for Community Services*, pages 194–204. Springer, 2018.
- [306] Walter, B., Marschner, S.R., Li, H. and Torrance, K.E. Microfacet models for refraction through rough surfaces. *Rendering techniques*, 2007:18th, 2007.
- [307] Wang, C.P., Snavely, N. and Marschner, S. Estimating dual-scale properties of

- glossy surfaces from step-edge lighting. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 1–12, 2011.
- [308] Wang, L., Villamil, R., Samarasekera, S. and Kumar, R. Magic mirror: A virtual handbag shopping system. In *Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 19–24. IEEE, 2012.
- [309] Wang, S., Shen, X. and Yu, K. Real-time 3d face reconstruction from single image using end-to-end cnn regression. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3293–3297, 2021.
- [310] Wang, Y., Lu, Y., Xie, Z. and Lu, G. Deep unsupervised 3d sfm face reconstruction based on massive landmark bundle adjustment. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1350–1358, 2021.
- [311] Ward, G.J. Measuring and modeling anisotropic reflection. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 265–272, 1992.
- [312] Welpöner, M., Stathopoulou, E. and Remondino, F. Monocular depth prediction in photogrammetric applications. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43:469–476, 2022.
- [313] White, M., Mourkoussis, N., Darcy, J., Petridis, P. et al. Arco-an architecture for digitization, management and presentation of virtual exhibitions. In *Proceedings Computer Graphics International, 2004.*, pages 622–625. IEEE, 2004.
- [314] Wiwatwattana, N., Sukaphat, S., Putwanpen, T., Thongnuch, S. et al. Augmenting for purchasing with mobile: Usage and design scenario for ice dessert. In *IISA 2014, The 5th International Conference on Information, Intelligence, Systems and Applications*, pages 446–450. IEEE, 2014.
- [315] Woodham, R.J. Photometric method for determining surface orientation from multiple images. *Optical engineering*, 19(1):139–144, 1980.
- [316] Wu, C. VisualSFM: A visual structure from motion system. <http://ccwu.me/vsfm/>, 2011.
- [317] Wu, C. Towards linear-time incremental structure from motion. In *3D Vision-3DV 2013, 2013 International conference on*, pages 127–134. IEEE, 2013.
- [318] Wu, C., Agarwal, S., Curless, B. and Seitz, S.M. Multicore bundle adjustment. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3057–3064. IEEE, 2011.
- [319] Wu, S., Ruppel, C. and Vedaldi, A. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *Proceedings of the IEEE/CVF*

- Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2020.
- [320] Xiang, J. and Zhu, G. Joint face detection and facial expression recognition with mtcnn. In *2017 4th international conference on information science and control engineering (ICISCE)*, pages 424–427. IEEE, 2017.
- [321] Xie, W., Dai, C. and Wang, C.C. Photometric stereo with near point lighting: A solution by mesh deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4585–4593, 2015.
- [322] XL Tech Apps. Glassify try on virtual glasses. <https://play.google.com/store/apps/details?id=com.xl.apps.virtual.glass.tryon>, 2020. Accessed on 19 January 2023.
- [323] Xu, S., Yang, J., Chen, D., Wen, F. et al. Deep 3d portrait from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7710–7720, 2020.
- [324] Yang, R.P., Liu, Z.T., Zheng, L.D., Wu, J.P. et al. Intelligent mirror system based on facial expression recognition and color emotion adaptation—imirror. In *Chinese Control Conference (CCC)*, pages 3227–3232. IEEE, 2018.
- [325] Yao, Y., Luo, Z., Li, S., Zhang, J. et al. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1790–1799, 2020.
- [326] Ye, W., Li, X., Dong, Y., Peers, P. et al. Single image surface appearance modeling with self-augmented cnns and inexact supervision. In *Computer Graphics Forum*, volume 37, pages 201–211. Wiley Online Library, 2018.
- [327] Yeo, U.C., Park, S.H., Moon, J.W., An, S.W. et al. Smart mirror of personal environment using voice recognition. *Journal of the Korea institute of electronic communication sciences*, 14(1):199–204, 2019.
- [328] Yi, K.M., Trulls, E., Ono, Y., Lepetit, V. et al. Learning to find good correspondences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2666–2674, 2018.
- [329] Yu, F., Koltun, V. and Funkhouser, T. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017.
- [330] Yu, Y.C., You, S.D. and Tsai, D.R. Magic mirror table for social-emotion alleviation in the smart home. *IEEE Transactions on Consumer Electronics*, 58(1):126–131, 2012.
- [331] Yuniarti, A. and Suciati, N. A review of deep learning techniques for 3d recon-

- struction of 2d images. In *2019 12th International Conference on Information & Communication Technology and System (ICTS)*, pages 327–331. IEEE, 2019.
- [332] Zafeiriou, S., Zhang, C. and Zhang, Z. A survey on face detection in the wild: past, present and future. *Computer Vision and Image Understanding*, 138:1–24, 2015.
- [333] Zhang, S., Zhu, X., Lei, Z., Shi, H. et al. Faceboxes: A cpu real-time face detector with high accuracy. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–9. IEEE, 2017.
- [334] Zhang, S., Zhu, X., Lei, Z., Shi, H. et al. S3fd: Single shot scale-invariant face detector. In *Proceedings of the IEEE international conference on computer vision*, pages 192–201, 2017.
- [335] Zhao, L., Huang, S. and Dissanayake, G. Linear sfm: A hierarchical approach to solving structure-from-motion problems by decoupling the linear and nonlinear components. *ISPRS Journal of Photogrammetry and Remote Sensing*, 141:275–289, 2018.
- [336] Zhao, X., Wu, X., Miao, J., Chen, W. et al. Alike: Accurate and lightweight keypoint detection and descriptor extraction. *IEEE Transactions on Multimedia*, 2022.
- [337] Zhou, M., Ding, Y., Ji, Y., Young, S.S. et al. Shape and reflectance reconstruction using concentric multi-spectral light field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(7):1594–1605, 2020.
- [338] Zhou, Y., Hu, L., Xing, J., Chen, W. et al. Hairnet: Single-view hair reconstruction using convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 235–251, September 2018.
- [339] Zhu, X., Yang, F., Huang, D., Yu, C. et al. Beyond 3dmm space: Towards fine-grained 3d face reconstruction. In *European Conference on Computer Vision*, pages 343–358. Springer, 2020.
- [340] Zulkifli, A.N., Alnagrat, A.J.A. and Mat, R.C. Development and evaluation of i-brochure: A mobile augmented reality application. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 8(10):145–150, 2016.

“Ok, that’s it, turn off your computer
and do something constructive.”

(Monkey Island 2: LeChuck’s Revenge)