POLITECNICO
MILANO 1863

POLITECNICO DI MILANO
Department of Mechanical Engineering
Doctoral programme in Engineering Design and Manufacturing
for the Industry of the Future
Cycle XXXV

Overcoming Mixed Reality adoption barriers in design through a computer vision-based approach for content authoring

Thesis submitted for the degree of doctor in
Mechanical Engineering at Politecnico di Milano

to be defended by

Eduardo Andrés Piñones Zuleta

Supervisors:
Prof. Gaetano Cascini
Prof. Giandomenico Caruso

Tutor:
Barbara Previtali

Coordinator of doctoral program:
Prof. Andrea Bernasconi

Declaration

I declare that the work presented in this thesis has not been submitted for any other degree or professional qualification and is the result of my independent work.

Eduardo Andrés Piñones Zuleta

31-01-2023

Summary

Prototypes are fundamental in the product development process as different types of prototypes can take different roles depending on the needs of the developers and the expected decision outcomes. Moreover, they can significantly impact a design project's success by working as a risk-reduction tool when used effectively. Due to this, several technologies have been developed to support the prototyping process. Among them, mixed prototypes have demonstrated to be valuable technology in product development.

This type of prototypes comprises a physical part that allows for realistic user interactions and a digital replica that work as a canvas to allow real-time modifications. In order to create them the pose of the physical elements are tracked within the environment (i.e., through computer vision algorithms). Then, the pose information is used to align the digital model, allowing the projection or rendering of the digital elements over its surface with a corrected perspective. This effect can be achieved using head-mounted displays or projector-based augmented reality, the latter being the main method used during the research.

Although several benefits have been found on the use of mixed prototypes created using projector-based augmented reality, it has yet to achieve a successful technology adoption due to a highly complex content authoring and setup process. In this context, the present thesis aims to reduce the existing adoption barriers of mixed prototyping and allow subjects involved in the design process to take advantage of the benefits of this technology. The main objectives are:

O1: To develop a workflow and supporting tool that can be used to streamline the mixed prototyping content authoring process.

O2: To develop a workflow and supporting tool that reduces the 3D modelling competence requirements to conduct the mixed prototyping authoring process.

O3: To develop a workflow and supporting tool that allows subjects with low competence in 3D modelling to achieve a mesh segmentation of similar quality to one made by an experienced user using standard tools.

The literature review shows that methods to support mixed reality content authoring have been developed in various contexts. Within these methods, those focused on creating 3D models that match the geometry of physical objects are the best suited to support the authoring process of mixed prototyping. However, it was found that most of the existing methods are limited to the use of primitive geometries and the creation of other simple shapes. Hence, they do not match the requirements of mixed prototyping, where the virtual and physical prototypes must be geometrically equal. Some methods based on surface reconstruction techniques allowed the creation of geometrically accurate replicas, using the results primarily for visualization or interactions with the environment. These methods have the potential to support the authoring of mixed prototypes. However, the segmentation of the models, needed to allow changes in the colour of different parts or the placement of graphical elements on the surface, has yet to be addressed. By focusing on the literature on image and mesh segmentation methods,

several solutions exist that could automatically execute this task can be found (e.g., using geometrical features or machine learning to create a part-based segmentation). However, as most of them rely on the use of geometric information to segment the 3D model, either directly or indirectly, they are not entirely suitable for mixed prototyping, at least in two opposite circumstances:

- When two or more geometrical elements of the object must remain unsegmented to be treated as a unique part, e.g., the housing of a device that will be manufactured using the same material.

- When one geometrical element of the object must be segmented to be treated as independent parts, e.g., a cylindrical bottle composed of two different materials.

Considering the ability of surface reconstruction methods to replicate objects and the texture information they can provide, the first research question addresses the technical suitability of an approach that takes advantage of the texture information to generate a segmented model suitable for mixed prototyping. Moreover, as in any other supported process, it is essential to understand that the obtained ease of use has an associated cost related to reduced control over the results. Hence, the second research question addresses the differences in the results obtained using the proposed method against manual segmentation, which is the current method used to prepare 3D models for mixed prototyping. Finally, although the impact of the proposed approach on the overall technological adoption can be challenging to assess, the third research question aims to do this through the evaluation of the impact on the execution time, perceived ease of use, and reduction of the knowledge requirements in 3D modelling.

This thesis presents an integrated workflow of content authoring for mixed prototyping spanning from the availability of a physical prototype to exporting a 3D model suitable for mixed prototyping. The main contribution is a texture-based mesh segmentation algorithm to generate part-based segmentations of reconstructed surface models. The integrated workflow is composed of multiple sequential stages. First, a physical prototype preparation stage ensures that the object has texture information that can aid the segmentation, that is, high-contrast features that define the boundaries of the output segments. Next, surface reconstruction creates a digital replica of the object with its texture information. The model is prepared for the projection system by applying the required transformations (e.g., rotation, location, or scale). Then, a feature detection algorithm is used to extract the information related to the boundaries of the segments from the original texture and uses it to select the corresponding vertices in the 3D mesh. Finally, all the polygons within the detected features are individually assigned to the main segments detected, and new UV maps are generated for each segment. Hence, producing a 3D model that complies with all the requirements of mixed prototyping. Two implementations of the supporting algorithms were developed, one integrated into an existing 3D modelling software and the other as a standalone application.

The validation plan comprises three testing stages, each with different objectives and generating supporting data to answer the research questions. The first testing stage focuses on the technical suitability of the proposed approach. Consequently, the research activities begin with a simple setup, adding new variables and increasing the complexity

of the input at each step. First, synthetic input data is used, reducing the variables and complexity of the input model when compared to a reconstructed model and easing the development of the algorithms. After the core functionalities are implemented, the complete workflow is tested in a more realistic scenario using an actual physical prototype with added graphical marks. Finally, the evaluation is extended to multiple objects with a wide range of geometrical features and segmentation requirements. The results demonstrated the technical suitability of a texture-based mesh segmentation approach to process models used in mixed prototyping and its integration within the complete content authoring workflow.

The second testing stage extends the evaluation of the functionality of the texture-based mesh segmentation approach to unmodified physical prototypes, i.e., using already existing texture features. Moreover, it also focuses on the proposed approach's usability and ease of use of the developed implementations. The results show that computer vision is a suitable method for feature detection in this context and that the resulting segmentation using the proposed approach is acceptable in mixed prototyping. Additionally, the subjects, who had low expertise in 3D modelling, chose parameters that generated a properly segmented 3D model for mixed prototyping without manually intervening in the mesh. Hence, demonstrating the capability of the approach to reduce the 3D modelling competence requirements for mixed prototyping content authoring.

The third testing stage compares the results of the texture-based mesh segmentation method against the standard tools available in 3D modelling software. The comparison was focused on the quality of results relative to reference segmentations made by advanced users, the time required to execute the segmentation process, and the perceived ease of use of each method. The results show that the segmentations generated through the proposed approach are slightly less accurate than the manual segmentation. Nevertheless, they could still be suitable for applications where a low error level is acceptable. Moreover, the proposed approach demonstrated significant benefits in terms of time reduction, resulting in an average processing time of nearly 1/3 the time required for manual segmentation. Finally, regarding the perceived ease of use, the subjects perceived the proposed method as easier to use than the standard modelling tools to execute the mesh segmentation.

The research process was composed of five stages to accomplish the previously mentioned objectives. The first stage comprises two descriptive studies of the practice and research contexts, gathering information to support the design decisions, detect the existing research gaps and define the research questions. Using this information, a new workflow to support the mixed prototyping content authoring process in the second stage. The third stage defines a validation plan to verify the contributions of the proposed approach and answer the research questions. The fourth stage comprises the execution of the validation plan with the analysis of the obtained results. Finally, the fifth stage discusses the overall results of the experimental activities concerning the research questions, the limitations of the proposed approach, and the conclusions of the complete research activity.

Overall, the multiple research activities to validate the proposed texture-based mesh segmentation approach demonstrated its suitability to support the mixed prototyping content authoring process. The three main objectives of this thesis were achieved, with only a small loss in the accuracy of the results to gain considerable improvements in processing time and reduction of the 3D modelling knowledge required to conduct the authoring process. The outcomes of this research contribute to understand the potential that a texture-based mesh segmentation can have and the limitations of that must be considered. This thesis is a step forward to reducing the existing adoption barriers of mixed prototyping and allowing users to take advantage of the benefits of this technology.

Acknowledgments

I would like to thank all the people that have supported me during the Ph.D., without them, this dissertation would not be possible.

I acknowledge the generous financial support from the Chilean government through ANID (National Research and Development Agency).

I deeply thank my wife Claudia, who believed in me moving to another country far from home looking for new opportunities to keep growing, and who was there for me even during the most challenging times.

I give my sincere thanks to my supervisors, Gaetano Cascini and Giandomenico Caruso, who guided me during my research process and kindly shared their knowledge and experiences. They were always thinking of ways to help me improve my research and support all my activities.

To the other members of our research team. Federico Morosi, who shared all his expertise in mixed prototyping and the SPARK platform, where I tested the results of my research. And to Niccolò Becattini, who always provided valuable feedback on my testing activities.

I would also like to thank the professors Gabriele Guidi, Laura Micoli and Victor Urrutia, for allowing me to participate in their courses and gather valuable information for my research.

And finally, thanks to all the people that participated in my tests who provided their time and experience.

Glossary

AR	Augmented Reality
CAD	Computer-Aided Design
MP	Mixed Prototype
MR	Mixed Reality
PEoU	Perceived Ease of Use
PU	Perceived Usefulness
SAR	Spatial Augmented Reality
TAM	Technology Acceptance Model
TUI	Tangible User Interfaces

Table of contents

List of Figures

List of tables

Chapter 1: Introduction

Prototypes are fundamental in the product development process as different types of prototypes can take different roles depending on the needs of the developers and the expected decision outcomes (X. Zhou & Rau, 2019). Using them effectively can significantly impact a design project's success (Camburn et al., 2017) by working as a risk reduction tool (Ulrich & Eppinger, 2015). However, the use of prototypes has some drawbacks that must be considered. Designers commonly create several prototypes to test and refine until a satisfactory design is reached (Christie et al., 2012). Although prototypes can be made for a low cost at the initial stages of development, this is not true for more advanced stages where a higher fidelity is required (Liker & Pereira, 2018). Due to this, the prototyping process can become expensive.

Additionally, high-fidelity prototypes usually represent only one design variation, and it can be challenging to iterate over them (Ulrich & Eppinger, 2015), making the manufacturing of several prototypes almost mandatory, further increasing the cost of this process. Moreover, whenever a prototype is evaluated, and changes are applied, time must be spent applying those changes before continuing with the design evaluation. Hence, increasing the required time to complete the product development process.

These problems of the prototyping process, added to the continuous development of computer technologies that increased the interest of developers and users in technologies like virtual reality, augmented reality, and mixed reality (Ladwig & Geiger, 2018), have fostered the creation of new methods of prototyping. Among them, mixed prototyping has been demonstrated to solve the problems mentioned above and offer several other benefits to the product development process. However, due to a highly complex content authoring process, it has yet to achieve successful technological adoption, preventing designers from taking advantage of this technology.

In this context, this thesis aims to propose a solution to improve the technological adoption of mixed prototyping and evaluate its impact compared to the current state.

This chapter briefly introduces the context of mixed prototyping, followed by the related technology adoption challenges, the research aims and objectives, the research approach and finally, the structure of the thesis.

1.1 Mixed prototyping

It has been challenging to reach a consensus on what should be considered Mixed Reality (Speicher et al., 2019). However, the "traditional" definition proposed by Milgram & Kishino (1994) is a mix of real and virtual objects on a spectrum that spans from a fully real to a fully virtual world. In this context, mixed prototypes are physical prototypes with the capability of synchronising the digital and physical domains in near real-time, anchoring virtual information into their surface (Kent et al., 2021). In some cases, most of the mixed prototype can be digital, with only the parts that the user will touch being physical (Barbieri et al., 2013). While on other cases, the surface of the physical object can be used as a canvas to project the digital elements (See figure 1.1).

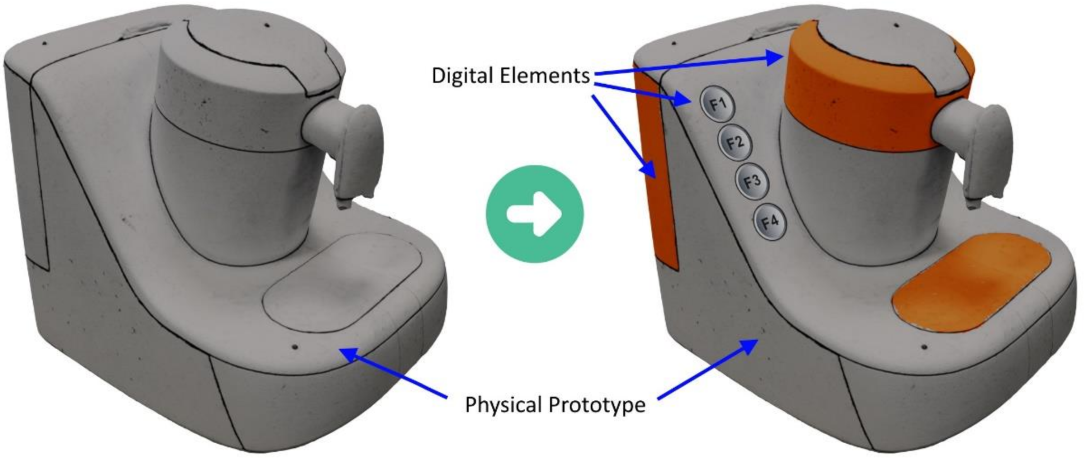


Figure 1.1 - Physical Prototype (left) and Mixed Prototype (right)

On one side, physical prototypes can provide passive haptic feedback during the interactions and naturally provide depth information, allowing designers to examine their physical look and feel, such as the dimensions and ergonomic fit (Akaoka et al., 2010). Moreover, multiple persons can simultaneously view and interact with physical prototypes, easing the design process in collaborative setups. Nevertheless, this type of prototype can be expensive from a time and cost perspective and usually does not allow more modifications (Verlinden et al., 2003), especially in the later stages of development.

Conversely, virtual prototypes allow designers to create and evaluate multiple design variants within a virtual environment reducing the time and cost of the product development process (Carulli et al., 2013). Moreover, in recent years, rendering engines have reached elevated levels of realism. Nevertheless, fully virtual platforms still lack haptic feedback, and when standard 2D displays are used, the depth information is lost, two aspects that play an essential role during the design process (Verlinden et al., 2003).

By using physical and virtual prototype characteristics, mixed prototypes can take advantage of their benefits while overcoming some weaknesses.

The creation of this type of prototype can be supported by different augmented reality technologies to add digital elements, such as see-through displays and projectors. Moreover, aside from a static physical prototype, haptic devices can also be used to simulate the physical response of buttons and other physical user interfaces (Kang et al., 2021), further improving their functionality. Considering this, the system architecture of mixed prototyping platforms can vary depending on the specific objectives of the design process that will be supported.

Mixed prototypes have allowed developers to avoid the need for physical prototypes; exhibit multiple designs of a product similar to the real one without the limitations of space (M. ki Park et al., 2015); create fully functional prototypes without the need for electronics, displays or physical buttons (Marner et al., 2011); create on-the-fly perceptual deformation and surface material manipulation during design evaluation (Takezawa et al., 2019); help find usability problems related to the appearance and functions of the product (X. Zhou & Rau, 2019); and enhance communication during co-creative design sessions (O'Hare et al., 2018). In conjunction, these benefits allow for speeding up the creation of design iterations and reduce the cost of prototyping, being especially suitable to support the design re-evaluation stage once the geometry has been defined (X. Zhou & Rau, 2019).

1.2 Mixed prototyping content authoring workflow

Since mixed prototyping is in early stages of development (Cascini et al., 2020), no standard method supports the complete workflow of content authoring on all implementations. Nevertheless, as presented in Figure 1.2, a general workflow that generates a connection between a physical prototype and a 3D model to create a usable mixed prototype can still be outlined.

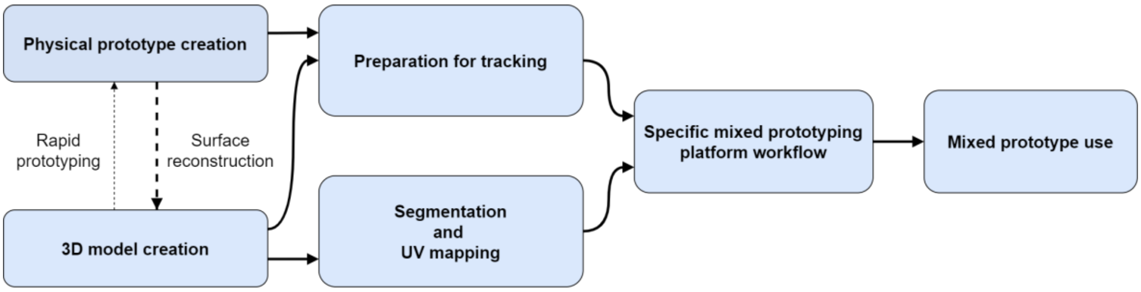


Figure 1.2 - General workflow of content authoring for mixed prototyping

In mixed prototyping, a 3D model of the physical prototype (or of the relevant parts to be evaluated) is necessary to correctly project the virtual information over its surface. On one side, if the 3D model is created first, the physical prototype can be manufactured, e.g., through rapid prototyping technologies. On the other side, if the physical prototype is created first, the 3D model can be created, e.g., through CAD or surface reconstruction methods. In some applications where the physical prototype to be augmented is based on primitive shapes, the prototype can be easily replicated through CAD software. However, this is not always an option when dealing with freeform geometries such as handcrafted prototypes, which could require considerable time and skill to reconstruct. Furthermore, it is likely that, in the latter case, the resulting digital models will not correspond to the prototypes manufactured through manual techniques due to geometrical imperfections. Additionally, some prototypes made of soft materials could change their shape after user interaction, worsening the discrepancy between the physical and digital prototypes. In this perspective, and given the limitations of CAD-supported workflows, suitable opportunities derive from surface reconstruction methods such as photogrammetry and structured light scanning. These techniques can reconstruct the majority of an object's geometry within 0.1mm of the reference, and which utility has already been evaluated with promising results in the product prototyping context (Gebler et al., 2021; Misal et al., 2019).

Once the 3D model is ready, regardless of the method used to create it, mixed prototyping applications focused on appearance modification also need to segment the digital model into several parts. This requirement allows the parts of the 3D model to receive a different treatment, e.g., to augment the corresponding portions of the physical model with diverse textures, annotations, and interaction modalities. Moreover, mapping bidimensional graphical elements on the surface of the physical object require a process known as UV mapping. This process correlates the coordinates of the 2D object and the 3D coordinates of the vertices and polygons that make up the 3D model.

Additionally, to enable the user to physically interact with the mixed prototype (e.g., moving it to see different perspectives) and coherently map the digital content in real-time, a tracking technology, which streams data of the position and rotation of the object, is needed. Tracking technologies use computer vision to detect fiducial markers, infrared markers or natural features of the object to use as input information to obtain a pose estimation (van Lopik et al., 2020). After the segmentation of the 3D model, UV mapping, and tracking preparation, the model is ready for integration into the mixed prototyping platform and be used during design sessions. Some other sub-steps, such as calibration of the system, the definition of graphical assets and the creation of an initial design layout, could be required according to the technologies and functionalities of the platform (O'Hare et al., 2018).

1.3 Research problem

Although the benefits of mixed prototyping and mixed reality technologies have been demonstrated, and users that have interacted with them agree that it could be helpful as a design tool (O'Hare et al., 2018), these technologies are still not widely adopted within product development processes as several reasons impact negatively the user acceptance.

From a final use standpoint, some of these reasons include a lack of realism in the surface reproduction (Cascini et al., 2020), a limited field of view (Kruijff et al., 2010), high latency on the tracking data (Morosi et al., 2018), and its dependence on the surface properties of the physical prototype (Raskar et al., 1999).

From a content authoring standpoint, both the physical and the virtual counterparts have an intrinsic realisation cost regarding skills, time and resources needed to create them (Kent et al., 2021). Moreover, since both must work coordinated, their creation is not trivial. Indeed, some of the reasons negatively affecting the user acceptance of this technology are the requirement of high levels of competence for content authoring (Bhattacharya & Winer, 2019) and a challenging set-up process for running the mixed prototyping platform consistently (Giunta et al., 2019).

Research is being conducted to solve some of those problems. However, there is a discrepancy between the research and industrial interests. While most research focuses on technological challenges instead of organisational ones, they are equally relevant to the industry. This difference of interests becomes even more noticeable regarding user acceptance (Masood & Egger, 2019). Due to this situation, currently, content authoring depends mainly on the developers of the technology or IT experts. Therefore the scale-up becomes difficult due to time requirements and content creation costs (Masood & Egger, 2020), counteracting some of the main benefits of mixed prototyping.

1.4 Research Focus

To properly take advantage of mixed prototyping and improve user acceptance, attention should be put on easing the content authoring process. As stated by previous research, there is a need to develop innovative tools that are more intuitive to create augmented experiences (de Souza Cardoso et al., 2020). Moreover, these solutions should be developed with the user's knowledge requirements in mind to avoid hindering the efficiency of the technology itself (Egger & Masood, 2020).

Furthermore, solutions focused on the initial steps of the general workflow of content authoring could improve user acceptance for various applications due to its independence from the mixed prototyping platform to be used. Within the initial steps, the mesh segmentation and UV mapping stand out with the highest potential for improvement and further development. This part of the process is time-consuming, requiring specific knowledge of multiple 3D modelling operations to achieve a satisfactory result (Tarini et al., 2017). Moreover, a solution focused on this part of the process has a high potential to aid the content creation of other applications supported by 3D content.

Considering this, the main objectives of the research towards the authoring process of mixed prototypes are:

O1: To develop a workflow and supporting tool that can be used to streamline the mixed prototyping content authoring process.

O2: To develop a workflow and supporting tool that reduces the 3D modelling competence requirements to conduct the mixed prototyping authoring process.

O3: To develop a workflow and supporting tool that allows subjects with low competence in 3D modelling to achieve a mesh segmentation of similar quality to one made by an experienced user using standard tools.

1.5 Research approach

This thesis will seek a contribution to practice and the existing body of knowledge. To do so, as recommended by Isaksson et al. (2020), a parallel validation journey (See figure 1.3) that takes into consideration both perspectives guided the research process.

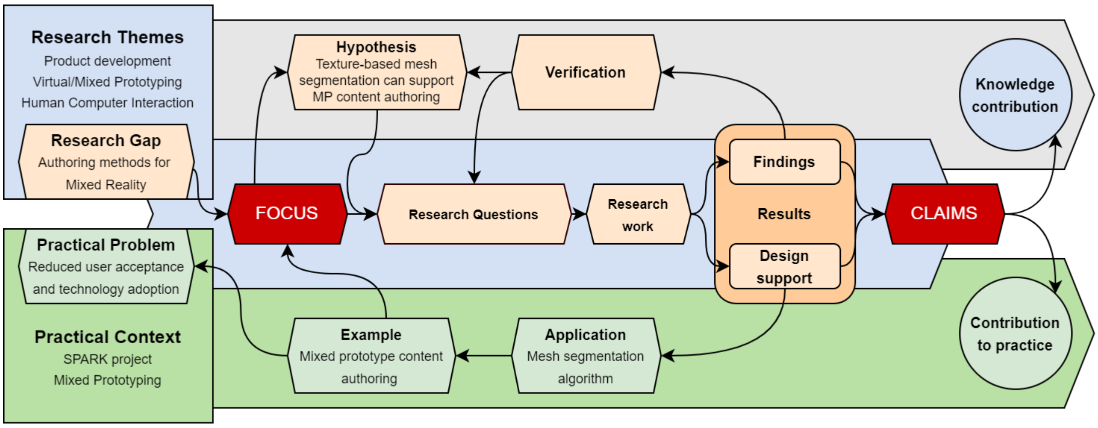


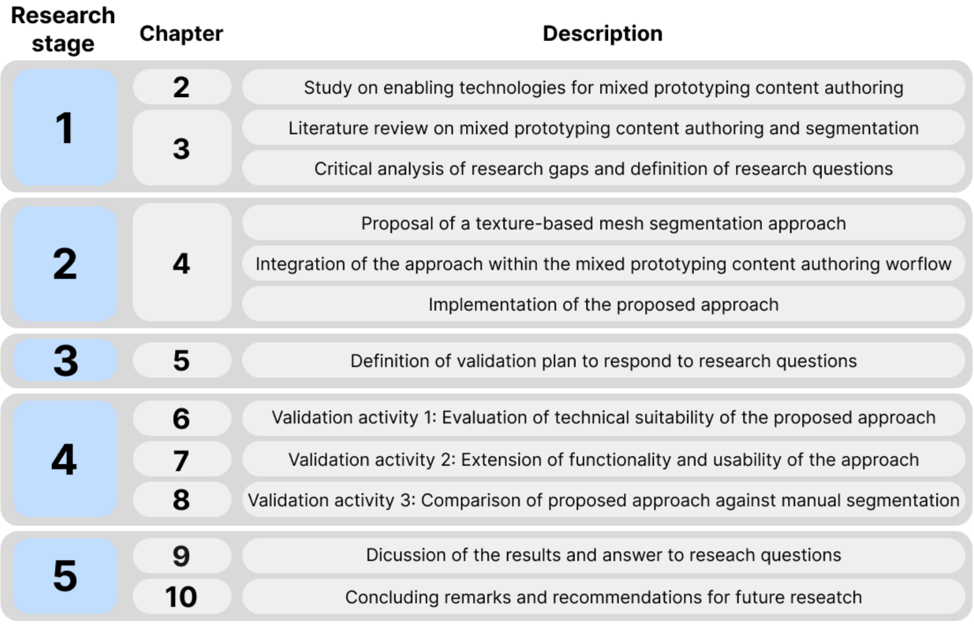
Figure 1.3 - PhD valitation journey

As shown in figure 1.4, the research process considers five stages. The first stage comprises two descriptive studies to gather information on the practical and research contexts. The first study focuses on the practical context, analysing the core elements of a mixed prototyping platform and the technical knowledge and tools needed to conduct the mixed prototyping content authoring process. The second study, on the other hand, focuses on the existing literature on methods that support the creation of mixed reality experiences, as well as the segmentation and UV mapping of 3D models. While the first study will support design decisions related to the technical development of the proposed approach, the second study, through critical analysis, will allow the detection of existing literature gaps and support the definition of the research questions.

The second stage considers the knowledge from the practice and research contexts analysis and proposes a new workflow to support the mixed prototyping content authoring process. Within this workflow, the primary novel contribution is a texture-based mesh segmentation algorithm to generate part-based segmentations of reconstructed surface models.

The third stage defines a validation plan to verify the contributions of the proposed approach and answer the research questions. Accordingly, multiple experimental activities are included focusing on various parts of the workflow, input objects and knowledge of the test subjects.

The fourth stage comprises the execution of the validation activities with the analysis of the obtained results. Finally, the fifth stage discusses the overall results of the experimental activities concerning the research questions, the limitations of the proposed approach and the conclusions of the complete research activity.



Research stage	Chapter	Description
1	2	Study on enabling technologies for mixed prototyping content authoring
	3	Literature review on mixed prototyping content authoring and segmentation
		Critical analysis of research gaps and definition of research questions
2	4	Proposal of a texture-based mesh segmentation approach
		Integration of the approach within the mixed prototyping content authoring worflow
		Implementation of the proposed approach
3	5	Definition of validation plan to respond to research questions
4	6	Validation activity 1: Evaluation of technical suitability of the proposed approach
	7	Validation activity 2: Extension of functionality and usability of the approach
	8	Validation activity 3: Comparison of proposed approach against manual segmentation
5	9	Dicussion of the results and answer to reseach questions
	10	Concluding remarks and recommendations for future reseatch

Figure 1.4 - Overview of research process

1.6 Thesis structure

Following this introductory chapter, this thesis includes two background chapters, five chapters disclosing the thesis research, a discussion, and a conclusion chapter.

Chapter 2 presents the enabling technologies for the development of the thesis research. It starts from the SPARK platform, the mixed prototyping platform where some validation tests were conducted. Then, information on 3D modelling and the functions needed in the authoring process is presented. Finally, two surface reconstruction methods capable of generating textured models are explained, as they provide the primary input data for the proposed method.

Chapter 3 presents a literature review on mixed prototyping content authoring methods in mixed reality. Additionally, the literature review is extended to segmentation methods to understand their limitations in mixed prototyping and how they can be used to improve the content authoring process. Finally, the chapter concludes with a critical analysis to identify the research gaps and define the research questions.

Chapter 4 proposes a computer-vision-based method to improve the mixed prototyping content authoring process. First, the complete workflow is explained, spanning from the availability of a physical prototype to exporting a 3D model suitable for mixed prototyping. The chapter ends with the presentation of two implementations of the proposed method developed during the thesis.

Chapter 5 presents the validation plan to respond to the previously defined research questions, including the motivations and methods for three testing activities. Their corresponding results are presented in chapters 6 to 8.

Chapter 6 applies a series of tests to the proposed approach to evaluate its technical suitability within mixed prototyping. First, part of the proposed approach is evaluated with synthetic input data to develop the core functionality of the algorithm and better understand its capacity to segment a model using texture features. Then, the complete workflow is evaluated with an actual physical prototype to evaluate the tool's robustness to handle complex geometries and input textures generated through surface reconstruction methods. Finally, a test includes users and multiple objects to evaluate its behaviour with various objects and gather early usability feedback.

Chapter 7 tests the proposed approach in the context of fashion design and with users with low competence in 3D modelling to evaluate two new perspectives. First, the capacity of the proposed approach to properly segment physical prototypes without needing extra modifications, i.e., using already existing texture features. And second, the user interaction with the developed algorithm to validate its impact on reducing the required 3D modelling knowledge to prepare a segmented model for mixed prototyping.

Chapter 8 compares the results of mesh segmentations with the proposed approach against manually done segmentations. The assessment includes the time required to finish the process, the quality from a geometrical standpoint, and the perceived ease of use.

Chapter 9 presents a general discussion of the research outcomes in response to the research questions, the identified limitations related to the research methodology and the proposed approach.

Chapter 10 concludes the thesis by presenting a summary of the findings related to the research focus and objectives; the implications for the scientific community, users, and the industry; and recommendations for future research.

Chapter 2: Background and enabling technologies

This chapter aims to provide the knowledge necessary to understand the choices made while developing the proposed mixed prototyping content authoring approach.

The chapter starts by presenting the SPARK platform, the mixed prototyping system the author referred to for developing the content authoring approach. In particular, the development context, the platform's aim, technology, and relationship with the research problem are explained. Then a short description of 3D modelling is provided with the primary operations that will be used or automated by the proposed approach. Finally, surface reconstruction methods are described as they generate the primary input data for the proposed method.

2.1 SPARK platform

2.1.1 Context

The mixed prototyping platform called SPARK (See figure 2.1) is installed within the Virtual Prototyping Laboratory, where the author developed most of its work. This platform aims to enhance the communication within co-creative design sessions by allowing the different stakeholders (e.g., designers, engineers, final users, and marketing people) to simultaneously visualise and modify different design variants of a mixed prototype.

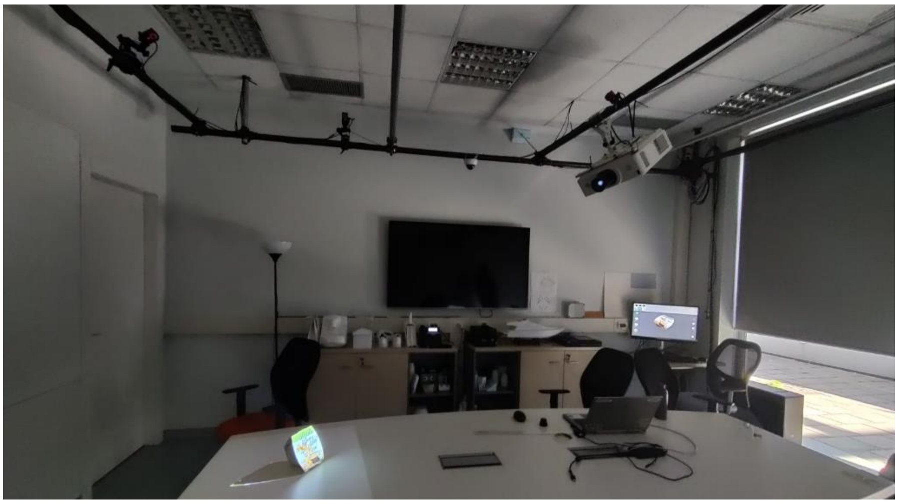


Figure 2.1 - SPARK room at the Virtual Prototyping Laboratory of the Politecnico di Milano

This platform was developed with funding from the European Union's Horizon 2020 Research and Innovation programme (Grant code H2020_ICT_2015_SPARK_688417). It was a joint project between the Politecnico di Milano, the University of Bath and the Grenoble Engineering and Management Institute.

2.1.2 Spatial Augmented Reality

The SPARK platform uses SAR (Spatial Augmented Reality), also known as projector-based augmented reality, which was first introduced by Raskar et al. (1999).The SPARK platform uses projectors to add digital information onto the surface of physical objects, such as prototypes. This technology can show perceptually undistorted graphics on tridimensional surfaces in the environment by pre-warping and colour-adjusting the virtual data (J. Zhou et al., 2012). Figure 2.2 shows a physical prototype of a soup container before and after the projector-based augmentation by the SPARK platform.

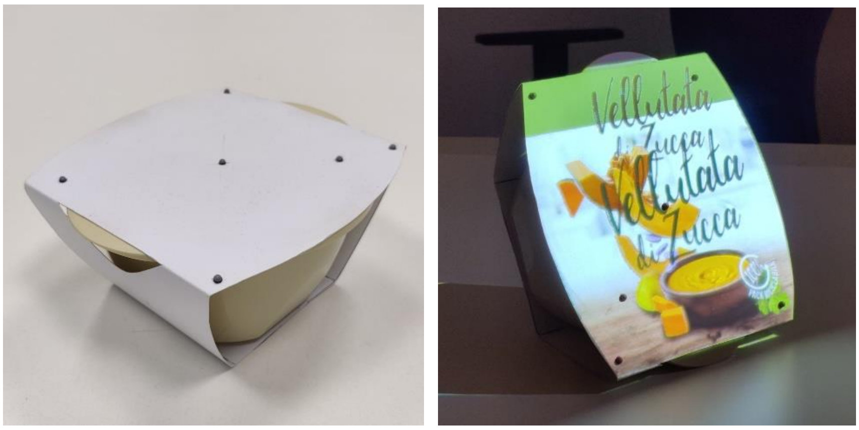


Figure 2.2 - Mixed prototype before (left) and after (right) projector-based augmentation

There are several benefits of this technology when it is compared to other mixed reality alternatives. Unlike head-worn and hand-held AR, SAR does not place the technology on the user but instead integrates it into their surrounding environment (Bimber & Raskar, 2005). Hence, it allows the view of the mixed prototype from any angle simultaneously by all the participants without the need for wearables or other visual devices, making it ideal for collaborative tasks (Marner & Thomas, 2014). This benefit is demonstrated in a comparative analysis between SAR, AR and conventional conditions (O'Hare et al., 2018), where the SAR condition scored well on the collaboration and exploration aspects of the Creativity Support Index developed by Cherry and Latulipe (2014).

Regarding physical interactions, in contrast to CAD and other AR display technologies, SAR allows users to physically touch the virtual information, where surfaces provide passive haptic feedback and all stereoscopic depth cues are naturally provided (Marner et al., 2011). This is an advantage studied in early mixed reality research, which demonstrated that the ability to touch virtual objects and information could enhance user experience (Hoffman et al., 1998) and user performance (Ware & Rose, 1999).

SAR systems can visually manipulate the user's perception of some surface characteristics, such as the bending stiffness of fabric (Punpongsanon et al., 2018). Moreover, they can leverage the human brain's multisensory integration of visual and haptic sensations to manipulate the user's perception of the softness of physical objects (Punpongsanon et al., 2015).

Finally, regarding the cost of these systems, contrary to those based on Head Mounted Devices, SAR systems can be made with low-cost off-the-shelf components (Giunta et al., 2018). Therefore, its adoption in industrial scenarios is continuously growing due to brighter, lighter and cheaper projectors (Uva et al., 2018). Although, it is worth noting that Head Mounted Devices for AR and MR are following a similar trend with higher resolution, lighter and cheaper devices being developed. Hence, could potentially be a more competitive alternative in the following years.

Thanks to the characteristics of SAR technology, several examples of mixed prototyping platforms using it can be found in the literature. These include WARP (Verlinden et al., 2003), which demonstrated its functionality with a small car model by allowing the user to change its appearance and the lighting environment. DisplayObjects (Akaoka et al., 2010) proposed a system capable of prototyping functional physical interfaces on models made from low-cost materials such as polystyrene foam, paper or cardboard. Marner et al. (2011; 2010, 2014) explored its application on interactive user interfaces and large-scale objects such as dashboards and cabinets. Furthermore, park et al. (2015) evaluated its suitability for appearance evaluation considering more complex surface characteristics.

Additionally, although the focus of the thesis is not to solve the specific problems and limitations of SAR, they have been included to give the readers a complete overview of this technology.

The main limitation of SAR is the incapability to drastically change the perceived shape of the mixed prototype due to the dependence on the unchanging surface of a physical object (Giunta et al., 2018). Research has been done to simultaneously modify a physical and virtual object (Marner & Thomas, 2010). However, usually, this is not possible on objects with complex geometries due to the lack of precision and methods to retrieve the new physical geometry in real-time.

Moreover, although the system can be used with a single small projector, this limits the number of viewpoints that can view augmentations (Marner et al., 2011). Multiple projectors can be used to create more extensive areas of projection. However, this introduces the problem of overlapping projections and increases system requirements such as cabling, signal routing, networking, and environmental considerations such as air conditioning, projector mounting systems, and lighting (Marner et al., 2011). Furthermore, any physical object placed in the working area will cast shadows, causing occlusion of the projection in the surface of the interest object or projections on unwanted surfaces. This situation is especially relevant during physical interactions, where the user's hands can occlude the added digital information (Akaoka et al., 2010).

Finally, regarding the projection quality, the incident angle of the projected light changes the physical shape of the pixels and the corresponding covered area, affecting the resulting brightness of the surface and the colour fidelity (Kruijff et al., 2010). In addition, there is no clear focal distance when projecting onto three-dimensional objects or a flat surface at an angle. Therefore, some parts of the augmentations will appear sharp, and others will be out of focus (Bimber et al., 2005). Moreover, while systems have been developed where the augmentation is displayed at the correct location while the object or the projector moves (Ehnes et al., 2004), registration (i.e., the correspondence between the physical object and the projection) usually includes some lag or drift. This characteristic is often not acceptable for many users (Kruijff et al., 2010), as a prior investigation revealed that a latency of more than six milliseconds is perceivable by human observers in interactive SAR systems (Ng et al., 2012).

2.1.3 System architecture

Although several versions of the SPARK platform exist, they all have a similar system architecture. The main difference is the maximum size of the mixed prototype that can be augmented and the degrees of freedom of it during user interactions. The version used during the project allowed mixed prototypes within a size range from 10 to 60 centimetres approximately and complete freedom of movement and rotations, i.e., 6 degrees of freedom.

As shown in figure 2.3, several elements need to work together to generate a functional mixed prototype of these characteristics:

1. The tracking system comprises several synchronised cameras that provide information to calculate the 3D location and rotation of groups of retro-reflective infrared markers. These markers are then placed on the surface of the mixed prototype, allowing the system to know its position and rotation in real time.
2. The projector-camera system is used to project the virtual elements over the surface of the mixed prototype.
3. The application offers a graphical user interface where the users can change some elements of the mixed prototype, such as the surface texture, colours and other 2D graphical elements.
4. The SAR controller manages the communication between all the other systems and connects to the information system, where 3D models of the prototypes, 2D assets and the history of different design sessions can be saved.

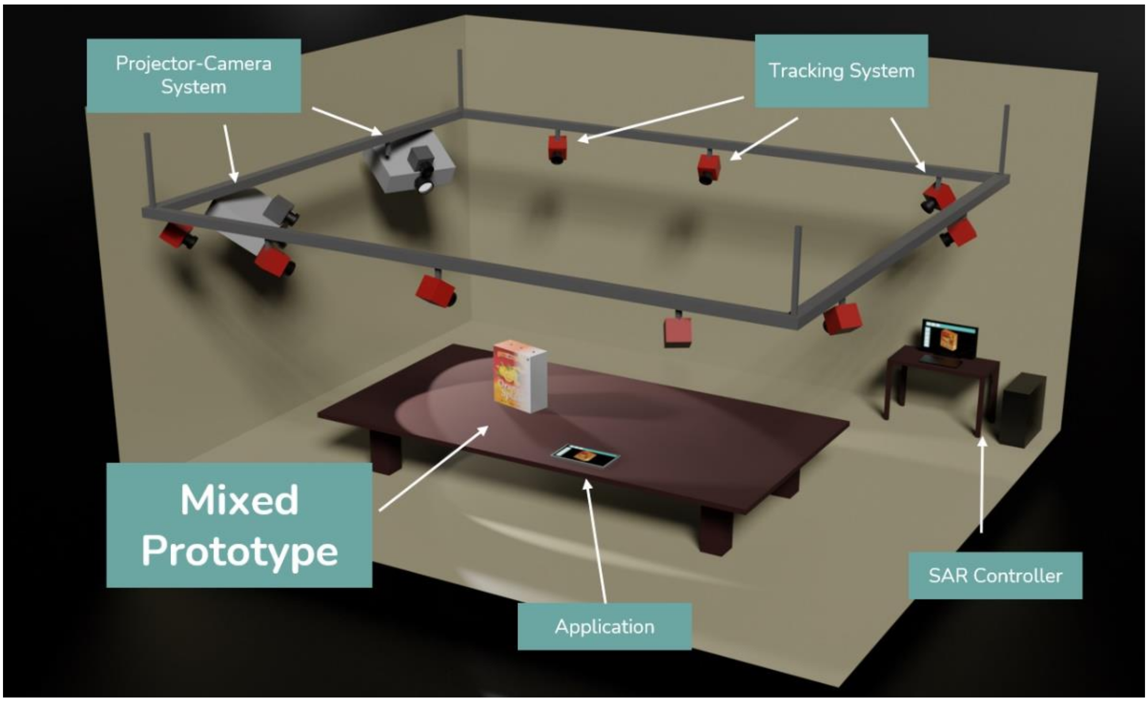


Figure 2.3 - Main elements of the SPARK mixed prototyping platform

2.1.4 Surface requirements for projector-based mixed prototypes

Four factors determine the quality of the colour reproduction on the surface of the mixed prototypes: the technology of the projector, the ambient illumination, the rendering algorithms implemented in the software of the mixed prototyping platform, and the surface characteristics of the augmented prototype. Nevertheless, from a content authoring perspective, in most cases, the platform setup already determines the first three factors. Therefore, the only one that can be controlled on a per-prototype basis is the augmented prototype's surface.

For an accurate projection of digital information, a light-coloured object with smooth geometry is ideal, as it is practically impossible to render vivid images on highly specular, low reflectance or dark surfaces (Raskar et al., 1999). Although, regarding this last characteristic, the complete opposite, a perfectly white object, is not necessarily the best choice. When the projection surface gets darker, the brightness of the image displayed by the projector lowers, making the result more realistic and closer to the reference materials that are being simulated (Morosi & Caruso, 2020). Consequently, any material could be used if its natural properties comply with these requirements or some treatment is applied on its surface to get such characteristics. However, from a practical perspective, as shown in figure 2.4, applying white or light grey paint with a mate finish should be enough for good-quality colour reproduction.

Figure 2.4 - Physical prototype painted with a white matte finish

2.1.5 Relation to the main research problem

As stated in section 1.3, where the research problem was presented, several technologies under the umbrella of mixed reality have their adoption hindered by highly specialised content authoring processes, and the SPARK platform is no exception. Research carried out with the platform showed that the usage of the platform offers benefits related to communication and reduction of cost and time required for product development (Cascini et al., 2020; Giunta et al., 2019; Morosi & Caruso, 2020; O'Hare et al., 2018). Nevertheless, underlying problems related to the preparation process were also found. Regarding this, professional design companies stated that the system's reliability during the preparation of new content was a problem (O'Hare et al., 2018), and test subjects mentioned that the setup process for running the mixed prototyping platform consistently was challenging (Giunta et al., 2019). Moreover, even though the platform has been in development for many years, the researchers considered it is still in the early stages of development due to these and other technology-related limitations (Cascini et al., 2020).

2.2 3D modelling

Here, 3D modelling is used broadly and refers to a set of operations that can be applied to a 3D model to change its characteristics. These operations are usually correlated to a specific type of 3D model and can serve different purposes. Some of these include changing the geometry, either at a macro or surface level; segmentation of the 3D model on independent parts; applying texture information; creating a rig for movement animation, among others.

However, this thesis will focus on two 3D modelling operations, mesh segmentation and texture mapping, which will be applied over wireframe models. Their definition and relation to the research are explained in the following subsections.

2.2.1 Wireframe models

In computer graphics, a wireframe model is a mathematical representation of a geometrical object. This representation comprises several elements that allow the visualisation of the resulting 3D model. The most basic element is the vertices, points in the 3D space defined by the three coordinates X, Y and Z. Moreover, each of these vertices has a unique identifier or index. However, this information is not enough to reconstruct the surface of the 3D model, as there is no way to know how the vertices are connected to create an edge or a face. To provide this information, the second element that defines a 3D model is the faces. Each face is composed of a list of the vertex's indexes that must be connected sequentially to create a surface or polygon. Ideally, to ensure that each face is planar and there is no ambiguity in its shape, each face must be composed of only three vertices. Nevertheless, faces can be composed of a higher number of vertices. Using these two sets of information, shapes as simple as a box or as complex as a body can be described as the ones shown in figure 2.5.

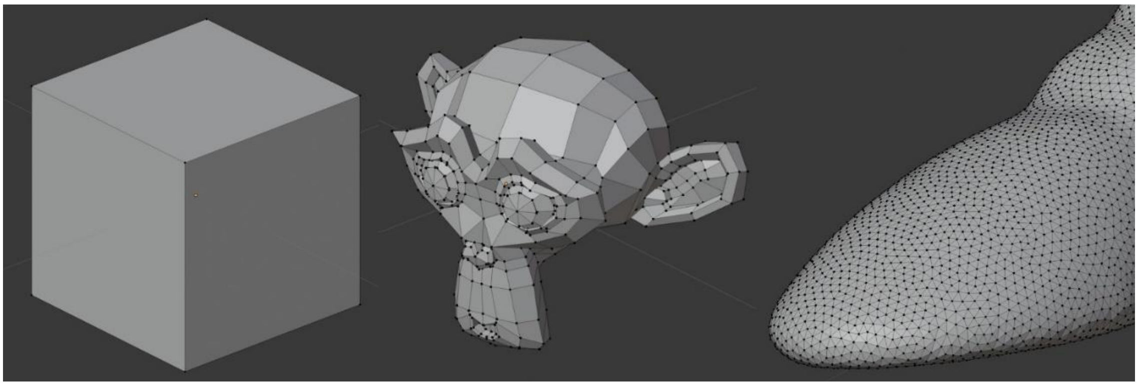


Figure 2.5 - Examples of wireframe models

If all the 3D model faces are connected and form a watertight surface, it is categorised as a solid model. Otherwise, they are considered only a surface, and it is not possible to calculate their volume or execute other operations, such as physical simulations or Boolean operations.

Wireframe modelling offer some unique benefits when compared to other methods such as solid modelling or NURBS (Non-Uniform Rational B-spline), the main being the editing freedom and versatility. Since the position of the vertices and faces can be modified independently, this type of modelling is well suited to represent complex geometries such as the ones of handcrafted or freeform objects. However, a trade-off is that many vertices and faces are needed to represent those complex surfaces accurately.

It is worth noting that the values of the vertices and faces that describe the 3D model are handled internally by the 3D modelling software, and the users typically interact with them through different tools rather than changing the values manually.

In the context of the research, this is the typical output from surface reconstruction methods, which are going to be used at the beginning of the mixed prototyping content authoring process due to their capacity to accurately replicate the geometry of physical prototypes.

2.2.2 Mesh segmentation

A 3D model that includes all the geometrical information of an object, usually referred to as a 3D object, can be helpful, as it allows us to quickly apply changes to all the faces while maintaining a connection between them. Some of these changes include the size, rotation, location, materials, or even more complex operations such as remeshing, duplication of the object or displacement of its surface. Nevertheless, some use cases can benefit from a 3D model separated into multiple parts or 3D objects, allowing us to apply the modifications to each part independently. For example, the 3D model of a car would not allow to independently move the doors or the wheels if these elements were part of a unique 3D object that is joined to the car's chassis. Moreover, although the application of multiple materials to a 3D object is possible, this process can be more complex than applying one material per 3D object, as the assignment of the materials must be done at the polygon level. Figure 2.6 shows an example of a 3D model before and after segmentation, where each colour indicates an independent 3D object.

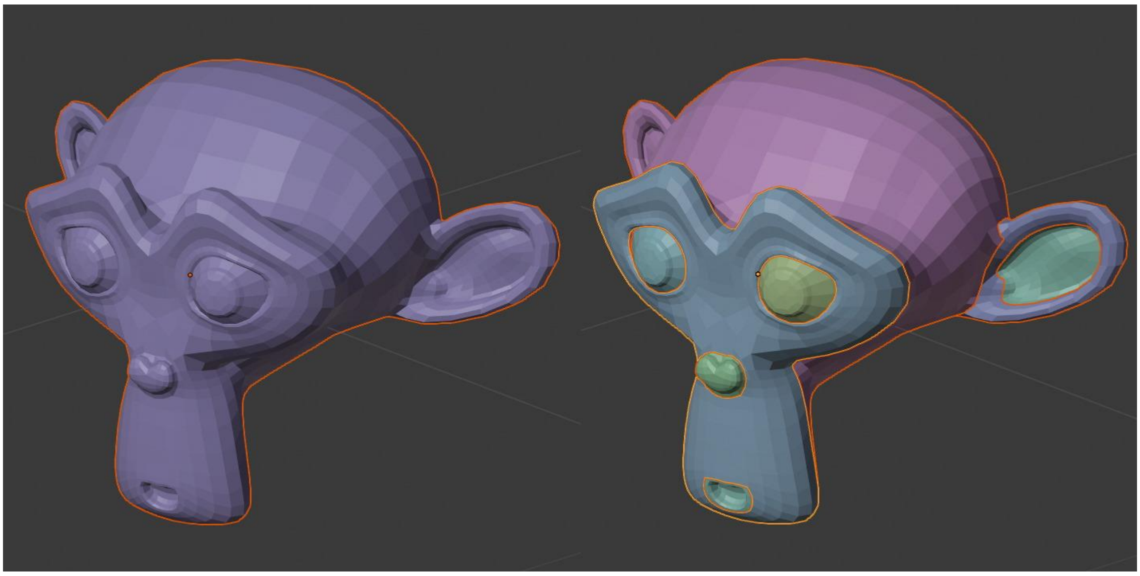


Figure 2.6 - Example of unsegmented model (left) and segmented model (right)

The process of mesh segmentation can be carried out using different techniques and tools integrated into 3D modelling software and geometry processing libraries. Nevertheless, at its core, this process requires two steps: selecting the faces of a 3D object that must be separated and creating one or more independent 3D objects using those faces. When a mesh segmentation is done, if two resulting 3D objects share the same boundary, the vertices in that boundary are duplicated and assigned to each segment. An alternative way to define a mesh segmentation of a 3D model without modifying the original mesh is creating a list of values equal to the number of faces in the 3D model, assigning to each an index corresponding to the segment they belong.

Regarding the relation of this operation with the research, although mesh segmentation is not a mandatory process to prepare a 3D model to be used in a mixed prototyping platform, it is a required step if several parts of a mixed prototype are going to be edited independently. A segmented 3D model will allow the users to apply different treatments to each part, including changing surface characteristics (i.e., colours, smoothness, glossiness, specularity or metalness) or different interaction modalities if the mixed prototyping platform allows it.

2.2.3 Texture mapping

In computer graphics, texture mapping is the process by which high-frequency signals such as the surface colours, glossiness and other surface parameters are defined over the surface of a 3D model (Tarini et al., 2017) (See figure 2.7). Rather than storing this information directly in the 3D model, the principle of texture mapping is to create multiple 2D images and then define a mapping operation between the 2D images and the faces in the 3D model. These images are known as texture maps, and several of them can be applied over the same 3D model to allow the rendering of realistic surfaces. Each texture map typically has information of one of the surface parameters. However, multiple parameters can also be encoded on different channels of the same texture map.

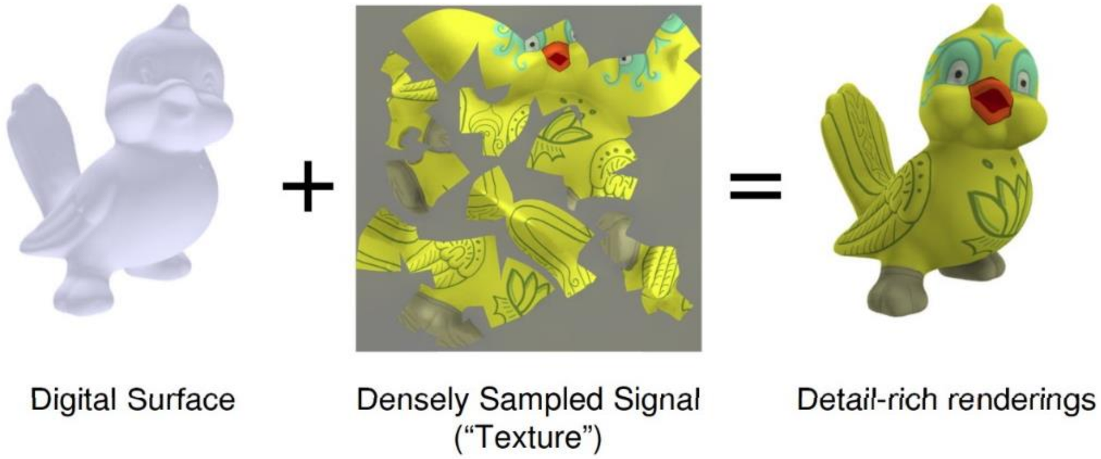


Figure 2.7 - General idea of texture mapping (Tarini et al., 2017)

The mapping between the 2D texture maps and the 3D model is known as UV mapping because a corresponding UV coordinate is assigned for each vertex in the 3D model. Figure 2.8 shows an example of a 3D model with its corresponding UV map.

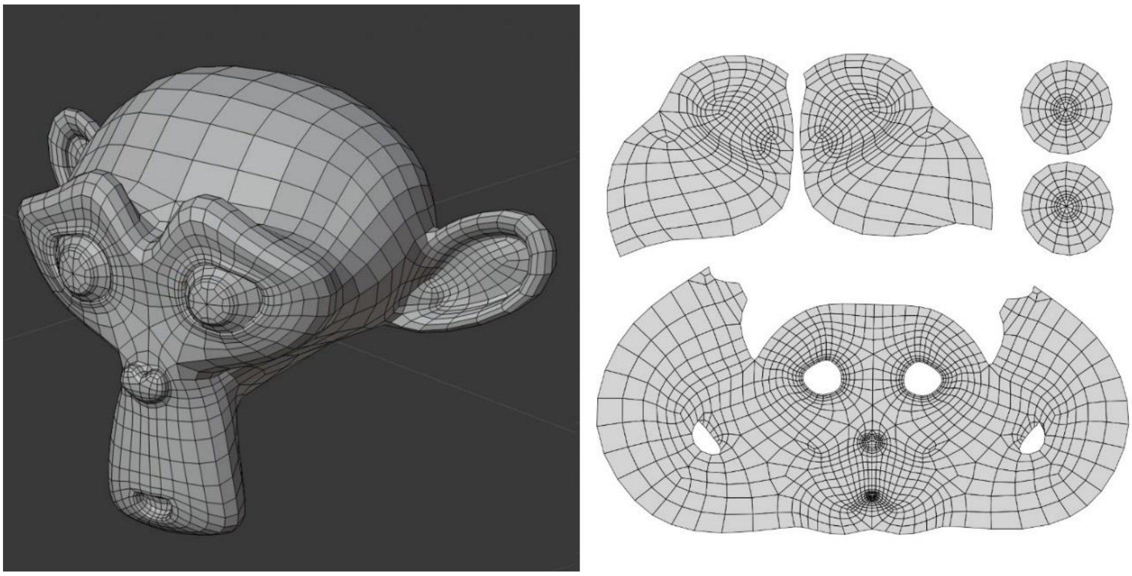


Figure 2.8 - Example of 3D model (left) with its corresponding UV map (right)

The UV mapping process can be carried out automatically or supported by the user's input and should consider several characteristics depending on the objective. One such characteristic is a reduced distortion of the textures when rendered over the 3D model. To achieve this, if the 3D model has a complex geometry, the mapped mesh in the UV map must be segmented and extended in a 2D plane, trying to maintain the shape of the faces. Conversely, other methods could focus on efficiently using the space in the texture maps. In this case, a mapping can be created that uses most of the surface of the texture map; however, this will usually result in a distorted UV map.

Regarding the relationship of texture mapping with the research, it is twofold:

First, 3D models with their corresponding texture mapping are needed during mixed prototyping to allow the modification of their surface characteristics. In this case, rather than having a static or pre-made texture map, the surface information can be modified and visualised in real time. The textures and other graphical elements are applied to a 2D canvas projected over the surface of the mixed prototype using the information from the UV map. Moreover, like the benefits of having a segmented model, an independent texture map for each segment allows a more straightforward editing process eliminating the possibility of applying changes to other segments by mistake.

Second, the information provided by a UV map can be used to correlate specific surface features of a physical prototype to vertices in the 3D model. Hence, this relationship makes possible to use those features as an input to execute other operations over the geometry, such as the mesh segmentation, one of the main operations to be automated by the proposed approach. Moreover, in mixed prototyping, as in most applications, a minimal distortion of the UV map is needed to project the virtual information properly. Therefore, it is a high-priority requirement of the resulting 3D model from the proposed approach.

2.3 Surface reconstruction methods

Surface reconstruction in computer science refers to different methods capable of recreating a tridimensional shape starting from various input data such as pictures, computed tomography scans, and depth information. The availability of such methods has increased, and a wide range of solutions applying different techniques, levels of performance and prices have been developed (Savio et al., 2007). Nevertheless, not all of them are suitable for the specific context of mixed prototyping or the requirements of the proposed approach. In particular, the proposed method for mixed prototyping content authoring requires texture information to extract surface features. Therefore, the method selected must generate the 3D model and the corresponding texture of the physical prototype.

2.3.1 Context

Early research in mixed prototyping, such as the work of Akaoka et al. (2010) already recognised the potential surface reconstruction methods to support the content authoring process more than ten years ago. However, it was considered too laborious due to the mesh clean-up process, and other methods based on primitive geometries were preferred. Nevertheless, current software for surface reconstruction has improved. It is not rare for them to include multiple post-processing steps to reduce the noise in the output 3D models, close holes in the surface of the geometry, generate clean 3D meshes of specific resolution or export models directly to use on other CAD software (Artec3D, 2022a).

Within all the available surface reconstruction technologies, photogrammetry and structured light scanning have been demonstrated to be well-suited to support prototyping processes. As presented in the work of Gebler et al. (2021), both technologies can reconstruct most of the geometry of physical prototypes with a tolerance of 0.1 mm, and the digitisation can be carried out in less than an hour, being acceptable in this context. Moreover, both can generate textured 3D models, making them suitable for the proposed method.

It is worth noting that other surface reconstruction methods could match these requirements. However, it was considered that photogrammetry and structured light scanning provided a representative sample of what future users of mixed prototyping would use, as there is a wide range of implementations available in the market, and they can easily be applied to products of different sizes.

The following subsections briefly explain how these technologies work, how the data acquisition and processing are carried out, and the specific software and devices used during the thesis.

2.3.2 Photogrammetry

As defined by the book “Computer Vision, A Reference Guide” (Schindler & Förstner, 2021), photogrammetry is “the science and technology of obtaining information about the physical environment from images, with a focus on applications in surveying, mapping, and high-precision metrology”.

From a technical standpoint, the photogrammetry process is an automated application of structure-from-motion theory (Ullman & Brenner, 1979), which is the estimation of 3D structures from sequences of 2D images of an object in motion, or that is seen from different perspectives. Essentially, the same process that the human brain does each time it estimates the depth or geometry of our surroundings using the 2D information provided by our two eyes. This process requires computer vision techniques such as pattern recognition, feature matching, semantic segmentation, and complex geometrical calculations to generate the output 3D model (Schindler & Förstner, 2021).

As expected from a technology that uses 2D images as an input, the main requirement is a camera. In theory, any camera can be used to generate the input images for the photogrammetric process, even a smartphone. Nevertheless, the resolution of the images, the dynamic range, the quality of the lenses and the accurate information from the sensor will impact the quality of the reconstructed model. For this reason, professional cameras are preferred.

Data acquisition is the first step in using this technology. To carry out this process, the user must take multiple photos of the object of interest and the number of photos needed depends on the object's complexity and size. Nevertheless, as shown in figure 2.9, an example of this process could consider taking pictures at five different heights and then moving 20 degrees around the object until doing a full circle, effectively taking 90 pictures. Moreover, sections occluded by other parts of the object during the previous process or with high amounts of detail could require more pictures to be reconstructed correctly.

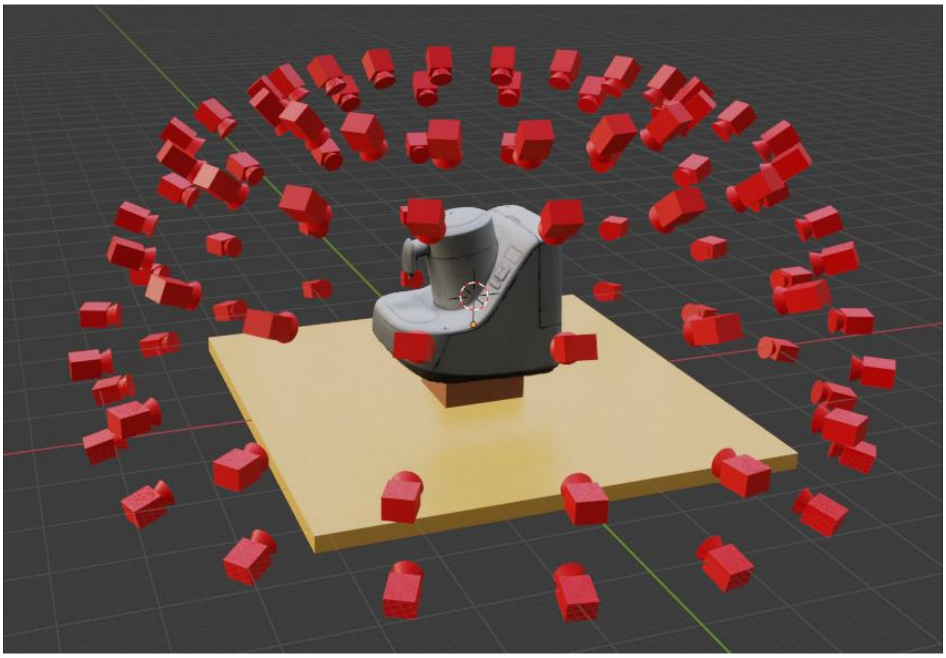
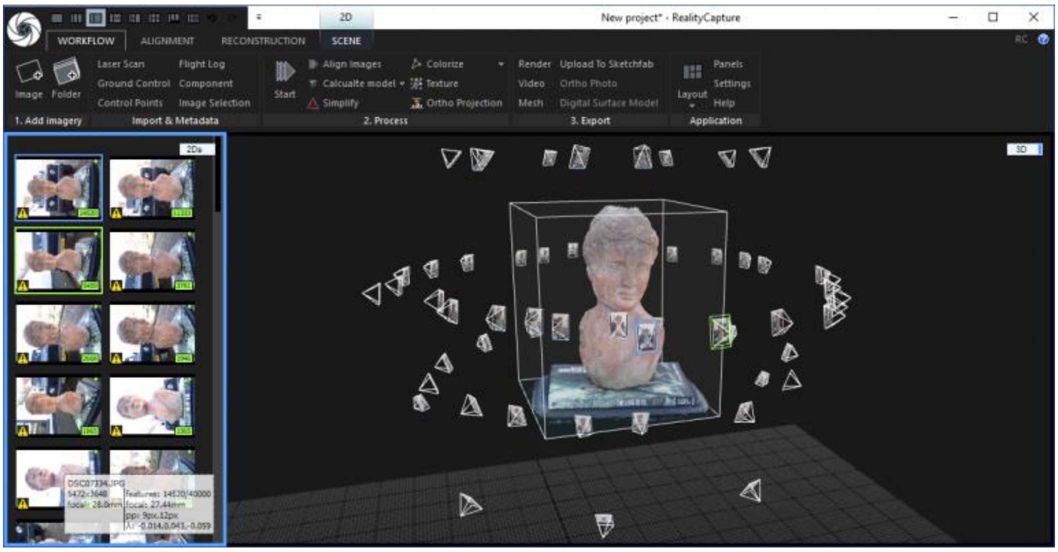


Figure 2.9 - Example of camera placement during data aqcuisition in photogrammetry

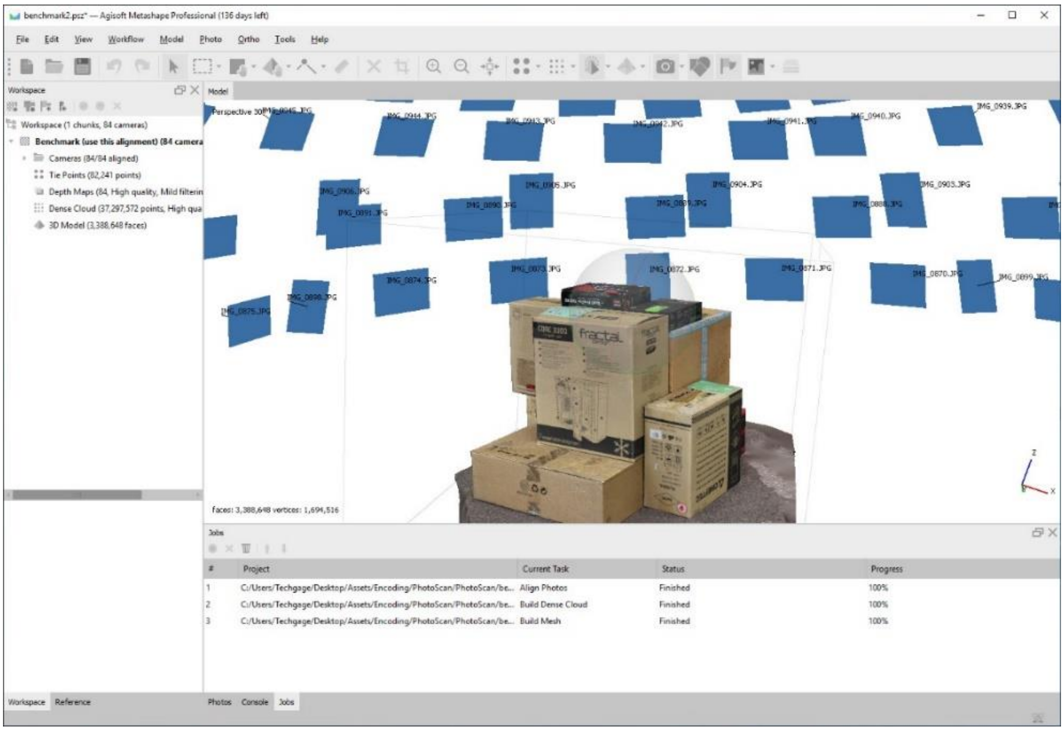
Once the images have been taken, the information is imported into photogrammetry software. Next, points of interest are extracted from the images to generate a correlation between them and reconstruct the camera's position in 3D space. Then the images are analysed to generate a point cloud or 3D model. Finally, a UV map is created for the 3D model, and the textures present in the input 2D images are projected over it to create a texture that covers the entire surface. The 3D model and the texture can then be exported to other software.

During the thesis, RealityCapture (Capturing Reality, 2022) and Metashape (Agisoft, 2022) were used to reconstruct 3D models using photogrammetry (See Figures 2.10 & 2.11). Additionally, since this surface reconstruction technique was mainly used during an activity with students, the cameras used for data acquisition included a wide range of devices, from smartphones to professional cameras with interchangeable lenses.



DSC07334.JPG
5472x3648
focal: 28.0mm
features: 14520/40000
focal: 27.44mm
pp: 9px,12px

Figure 2.10 - User interface of RealityCapture



Workspace (1 chunks, 84 cameras)
Benchmark (use this alignment) (84 camera
Cameras (84/84 aligned)
Tie Points (82,241 points)
Depth Maps (84, High quality, Mild filterin
Dense Cloud (37,297,572 points, High qua
3D Model (3,388,648 faces)

faces: 3,388,648 vertices: 1,694,516

#	Project	Current Task	Status	Progress
1	C:/Users/Techgage/Desktop/Assets/Encoding/PhotoScan/PhotoScan/be...	Align Photos	Finished	100%
2	C:/Users/Techgage/Desktop/Assets/Encoding/PhotoScan/PhotoScan/be...	Build Dense Cloud	Finished	100%
3	C:/Users/Techgage/Desktop/Assets/Encoding/PhotoScan/PhotoScan/be...	Build Mesh	Finished	100%

Figure 2.11 - User interface of Metashape

2.3.3 Structured light scanning

This technology, like photogrammetry, uses 2D images to reconstruct the surface of an object of interest. However, it adds extra information to support this process. The structured light scanning technology considers the projection of a pattern (i.e., black & white stripes or checkerboards) of visible light onto the surface of an artefact. Once this pattern reaches the surface of the object of interest, it is distorted due to its geometry, and its effect is captured using a camera. Knowing some internal parameters of the camera and projector, as well as their relative position, it is possible to use the information from the pattern distortion in the captured images to estimate the object's geometry (Gebler et al., 2021). In this way, each time a pattern is projected over an object and a picture of it is taken, it is possible to get an image with a corresponding z-depth map (i.e., a 2D array that indicates how far away from the camera each of the pixels in the image is). Multiple sets of images, in addition to the z-depth map, can then be joined to reconstruct the complete geometry of the object. Moreover, if additional images are taken without the projected pattern, this information can be used to project the colours over the resulting 3D model.

As previously mentioned, the main components of this technology are a camera and a projector. These can be independent of one another and used together through a calibration process to calculate the needed parameters, or can be integrated into a unique device where the parameters and relationships between them are known beforehand.

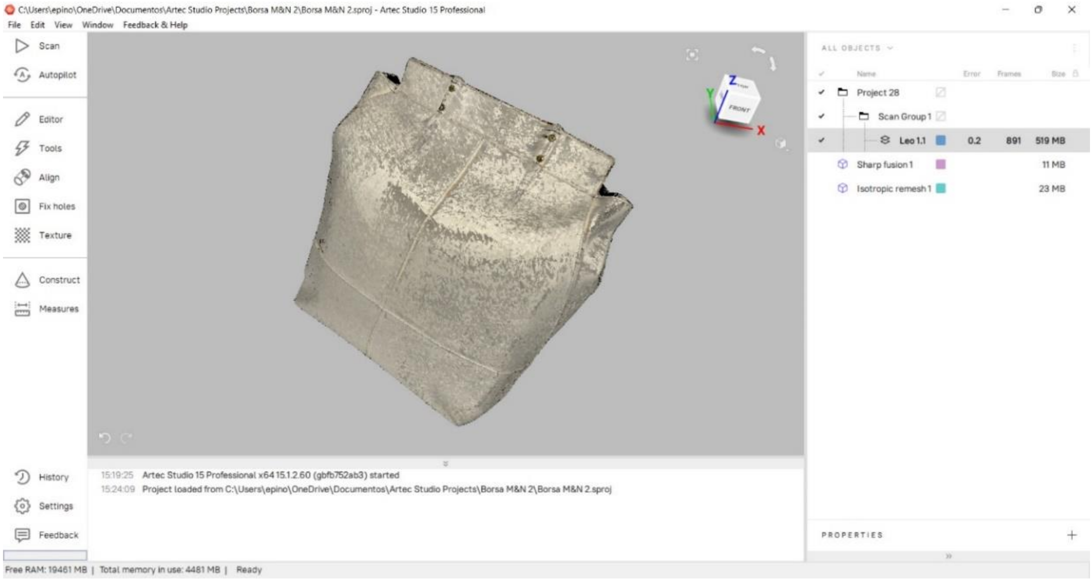
Depending on the configuration, the use of this technology can vary slightly. In some cases, the camera-projector system will remain static, and the object of interest will be placed over a controlled rotating tabletop, allowing it to reconstruct all sides. Alternatively, the camera-projector system can be integrated into a unique device. In that case, the object of interest can remain static, and the structured light scanning device can be moved around the object to reconstruct the surface.

Once the surface data is acquired, the following step is processing the information to create a unique 3D model. This process is usually done in software developed by the same developers of the device used for data acquisition. After the data is imported into the software, the individual reconstructions are aligned, and the geometrical information is cleaned up by deleting outliers in the surface or other unwanted elements. Next, the 3D model is remeshed to the desired resolution, a UV map is created, and finally, the colour information is projected over the surface to generate a textured 3D model that can be exported to be used on other software.

During the thesis, the Artec Leo (Artec3D, 2022b) was used. It is a portable professional surface reconstruction device with a resolution of up to 0.2 millimetres (See figure 2.12). Moreover, the specific configuration of this device included a diffuse light ring that was used during the capture of the colour images. Hence, allowing the generation of a textured 3D model without shadows.

Figure 2.12 - Artec Leo surface reconstruction device

Accordingly, the processing of the acquired data was done in Artec Studio (See figure 2.13).



C:\Users\epino\OneDrive\Documentos\Artec Studio Projects\Borsa M&N 2\Borsa M&N 2.sproj - Artec Studio 15 Professional

✓	Name	Error	Frames	Size
✓	Project 28			
✓	Scan Group 1			
✓	**Leo 1.1**	**0.2**	**891**	**519 MB**
	Sharp fusion 1			11 MB
	Isotropic remesh 1			23 MB

```
15:19:25   Artec Studio 15 Professional x64 15.1.2.60 (gbfb752ab3) started
15:24:09   Project loaded from C:\Users\epino\OneDrive\Documentos\Artec Studio Projects\Borsa M&N 2\Borsa M&N 2.sproj
```

Free RAM: 19461 MB | Total memory in use: 4481 MB | Ready

Figure 2.13 - User interface of Artec Studio

2.3.4 Surface requirements for reconstructed object

Since photogrammetry and structured light scanning are based on optical information, the superficial characteristics of the objects to be reconstructed directly affect the quality of the results.

Surfaces with a homogeneous plain colour should be avoided, as the algorithms for surface reconstruction require some reference points in the input data to calculate the correlation between them. For example, suppose multiple images are taken from a perfectly white wall. In that case, even if the images have been taken from different perspectives, the algorithm will not be able to reconstruct a planar geometry properly due to missing reference points. On the other hand, if a surface has a repeating pattern, the surface reconstruction algorithms will detect the needed reference features, but they could be confused, producing reconstruction errors.

Another surface characteristic that can produce reconstruction errors is reflectivity, such as the one found on metallic or glossy objects. Each time the camera perspective is changed, the reflections on the surface of the objects will also change. Hence, it is impossible to have a stable reference feature that can then be used to reconstruct the object. Similarly, when an object is translucent, such as glass or plastic, it is impossible to capture stable reference features at the object's surface to generate the reconstruction.

Opaque surfaces are preferred when reconstructing the surface of an object to minimise the previously mentioned effect.

Additionally, only for photogrammetry, ambient illumination is also an element to take into consideration to achieve good results. The ideal case would be to have a diffuse source of light that illuminates the object of interest from all possible angles without generating harsh shadows. This requirement is needed for two reasons. The first reason is that the capture device must stay in the same setup during the whole process of acquisition of images to maintain a stable input. Hence, if the camera is set up for a darker scene, well-illuminated areas will be overexposed, effectively losing data. Conversely, if the camera is set up for well-illuminated areas, the darker sections will be underexposed, generating the same problem. The second reason is related to the proposed approach. Since the texture information will be used as an input for the segmentation, if shadows generate high contrast features, they could be detected as false positives, negatively affecting the output segmentation.

2.3.5 Output data

2.3.5.1 3D model

Usually, surface reconstruction software offers several options to manipulate the generated 3D model before it is exported to be used in other software. Hence, it is not possible to define some of its characteristics. Nevertheless, since its objective is to replicate the geometry of a physical object closely, the output 3D model tends to have a high resolution (i.e., over 100.000 polygons) to consider small geometrical features. Moreover, they also tend to have an isometric mesh, meaning that each polygon of the 3D model has a similar area and length of its edges (See figure 2.14). Due to this characteristic, the resulting models are not well optimised. This feature is particularly evident in smooth or planar parts of the 3D model, which could be accurately defined with a reduced number of polygons, but use a high number of polygons.

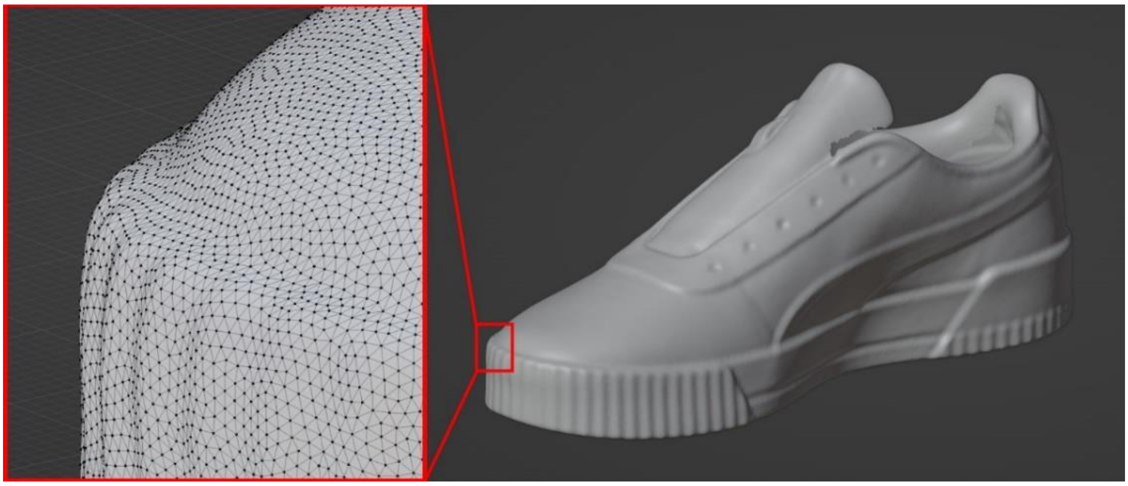


Figure 2.14 - Example of surface reconstructed model with closeup of its mesh

Regarding the output format, the most common are OBJ, FBX and STL.

Additionally, suppose a texture is to be applied in the 3D model. In that case, a UV map is created to correlate the information between the 3D texture and its position on the surface of the 3D model. The specific algorithm to generate the UV map will change between different software. Nevertheless, one of the most commonly used algorithms is the least squares conformal maps (Lévy et al., 2002). The UV map information is integrated into the exported 3D model regardless of the method used.

2.3.5.2 Texture

Surface reconstruction software typically generates textures in a square aspect ratio. These textures usually have a width that is a multiple of a power of 2, and some examples include 512, 1024, 2048, and 4096 pixels. Hence, each resolution level quadruples the amount of data in the texture. Moreover, as shown in figure 2.15, in the context of mixed prototyping, textures surface reconstructed models have specific characteristics that must be taken into consideration, such as:

1. Fragmentation: although the resulting image depends on the adopted software application, textures of reconstructed surface models are typically fragmented into several pieces to reduce their distortion. Consequently, a unique feature (i.e., the surface of a part of the prototype) will often be fragmented and distributed within the complete image.
2. Inpainting: To reduce rendering errors and account for missing data during the texturing of the 3D model, software also includes digital inpainting capabilities (Jam et al., 2021) that can affect the behaviour of image segmentation algorithms, for example, by graphically joining two segments that correspond to different parts of an object, but have the same texture patterns.
3. Homogeneous texture: usually, most of the surface of the mixed prototype has a homogeneous texture with a neutral colour that serves as a clean canvas to add digital information onto the surface of a physical prototype. This is typically required in projector-based augmented-reality applications, where the resulting surface colour depends directly on the original colour of the physical prototype (Morosi & Caruso, 2020).

Figure 2.15 - Characteristics of surface reconstructed textures of mixed prototypes

Additionally, some surface reconstruction software allows adjustments to the texture before exporting it. These adjustments include brightness, contrast, hue, and the reduction of reflection artifacts.

Regarding the output format, the most common are JPG and PNG.

Chapter 3: Literature review

This chapter presents the context of the thesis by giving an overview of the related research areas involved, the gaps that need to be addressed and their relevance. Hence, being the base for developing novel research contributions and achieve the objectives presented in section 1.4. The aim is to gather information on the existing methods developed to support the mixed reality content authoring process and to evaluate their potential to support the content authoring of mixed prototypes. The chapter begins with an explanation of the methodology used to collect the relevant literature and continues with a literature review on mixed reality content authoring methods presenting current research focused on easing this part of the process. Additionally, the literature review is extended to the segmentation methods to understand their limitations in mixed prototyping and how they can be used to improve the content authoring process. Finally, the chapter concludes with a critical analysis to identify the research gaps and define the research questions.

3.1 Literature review methodology

Snowballing was used to conduct the literature review (Wohlin, 2014). It is an iterative method that, after selecting a start set of relevant publications, uses the reference lists of each paper and their citations to identify additional papers to be considered for further analysis. The complete workflow to apply this method can be seen in Figure 3.1.

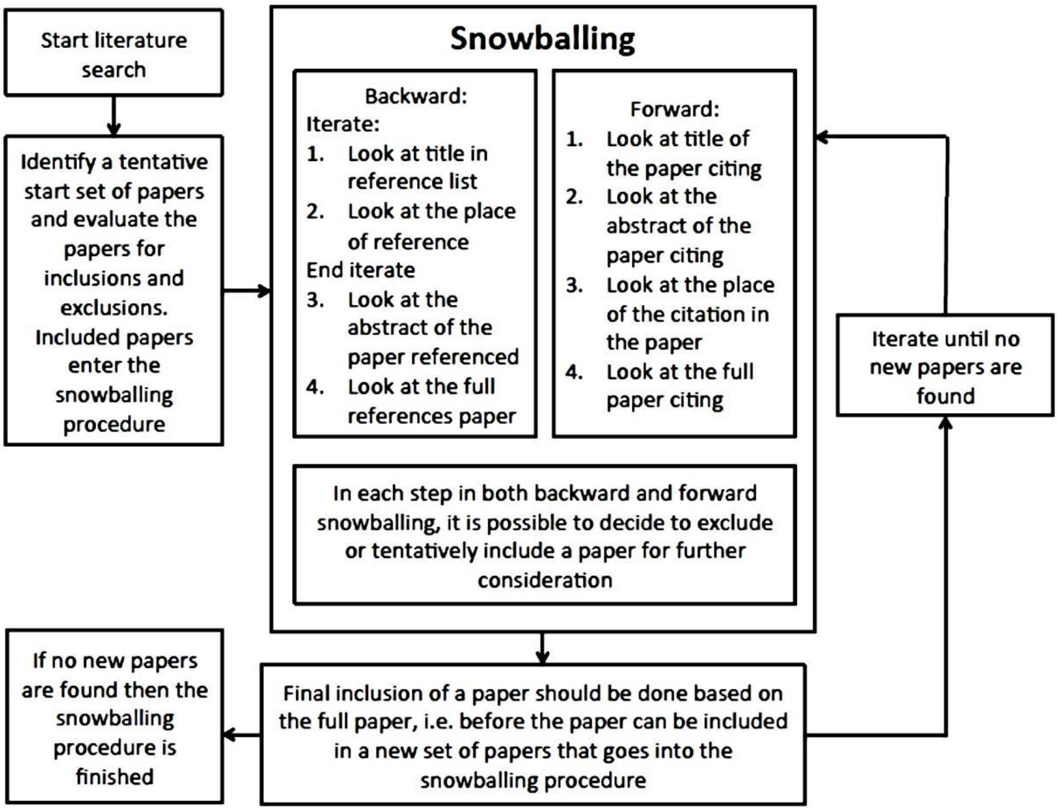


Figure 3.1 - Snowballing procedure (Wohlin, 2014)

Snowballing can provide a substantial knowledge increase in the topic of interest (Wnuk, 2017) while providing a similar efficiency to methods based on database searches

(Badampudi et al., 2015). Moreover, this approach has the added benefit of a reduced effort when updating the systematic literature review (Felizardo et al., 2016). Hence, being particularly useful in a multiyear endeavour such as a PhD, allowing to maintain an overview of the evolution of the research context.

3.2 Mixed Reality content authoring methods

In previous research on mixed reality applied to design tasks, the content has been created mainly by the developers and not by the final users (Masood & Egger, 2020). Several mixed reality implementations have also identified this limitation, making the users' upskilling one of the main barriers to their successful adoption (Davila et al., 2020). In this context, methods to support mixed reality content authoring have been developed in a wide variety of contexts, including assembly, maintenance, cultural dissemination, design, teaching, storytelling, presentations and training (Cubillo et al., 2015; Egger & Masood, 2020; Gimeno et al., 2013b; Harazono et al., 2022; Koleva et al., 2009; Mendes et al., 2018; Speicher et al., 2021; Vera & Sánchez, 2016; Zhu et al., 2013). While each method aims to simplify the authoring process, they also try to solve it from different perspectives. The following subsections present an overview of the most prominent trends in the literature review, their potential application in mixed prototyping and limitations in this context. It is worth noting that these are not exclusive, and one specific method or authoring platform could use several of the presented trends to aid the authoring process.

3.2.1 Programming simplification

One of the main barriers to adopting MR technology has been the high programming skill level to create such applications. As such, initial efforts focused on simplifying the creation of MR applications for the developers of the technologies rather than final users. The first objective was the organisation of all the concepts involved in creating MR. Some early examples of these efforts include Hampshire et al. (2006), which proposed an initial taxonomy of MR applications, addressing the different levels of abstraction for defining the relation between the real and virtual worlds. Similarly, APRIL (Ledermann & Schmalstieg, 2005), an authoring platform for MR content provided concepts and techniques independent of specific applications or target hardware platforms.

In this context, the core elements of mixed reality technologies, such as rendering, tracking, and user interactions, could be rearranged to create a wide range of MR experiences. Consequently, there was a proliferation of component-based methods to support the creation of mixed reality applications. Following this direction, the work of Bauer et al. (2001) and Dörner et al. (2003) proposed a flexible and modular software framework that allows components to be adapted and reused between applications. Grimm et al. (2002) developed AMIRE, a generic framework to simplify the communication between the MR components. Park et al. (2011) developed AR-Room, a series of deployable components for core augmented reality technologies, modules for hardware abstraction, and an authoring toolkit for rapid content design. More recently, Rumiński et al. (2014a, 2014b) developed CARL, a Contextual Augmented Reality Language that enables modularisation of the structure of AR scenes and the presented AR content.

Whether or not these proposed frameworks and programming libraries have been widely adopted by developers or are still in use is out of the scope of this literature review. However, the authoring of mixed reality applications has reached some level of maturity in this regard. One of the leading indicators of this is the shift in the focus to developing solutions to support MR platforms where the final users could create their content rather than depending on the developers.

Several tools and programming approaches have been developed within this new focus towards end-user content authoring. Among these, we can find the work by Castillo et al. (2016) which developed a node-based add-on to create AR experiences with reduced programming knowledge. The work by Radu & MacIntyre (2009), integrated typical MR functions in Scratch, a high-level block-based visual programming language developed for children. StoryMakAR (Glenn et al., 2020), allows children to create basic interactions between 3D characters and physical devices through a simple mobile interface. Meta-AR-app (Villanueva et al., 2020), a MR platform which allows the creation of AR animations through a drag-and-drop interface. AuthoAR (Lucrecia et al., 2013), that simplify the creation of AR learning experiences by offering content templates that the teachers can adapt to their needs. SUGAR (Gimeno et al., 2013a, 2013b), a platform supported by depth-sensing cameras to gather extra 3D information about the environment to help users overcome occlusion problems during content creation in industrial environments. Finally, the work of Knopfle et al. (2005), that proposed a template based approach where skilled developers created base templates that then could adapted and reused by non-skilled users.

While these methods are undoubtedly helpful in supporting the users' autonomy during content authoring, their application is limited in mixed prototyping for two reasons. First, they are supported by pre-made content, which would significantly limit the ability of the users to try new product shapes during prototyping. Moreover, they support the creation of interactions between different components of MR experiences. However, none of them focuses on preparing the 3D model, which is the step that allows the prototypes to provide further interactions.

3.2.2 Tangible user interfaces

Creating a link between physical and digital objects is one of the main features of MR technologies. For this reason, several methods have been proposed to make it as simple as possible, allowing final users to easily link 3D content to physical markers (Barone Rodrigues et al., 2017; Coquillart et al., 2004; Jee et al., 2011; Koleva et al., 2009). Furthermore, this process, initially limited to mostly visualising experiences, was extended to allow the users to interact with the digital content through physical elements, a method known as TUI (Tangible User Interfaces).

Some examples of these developments are the work of Ha & Wo (2007), which proposed a TUI capable of doing rough modification to AR 3D scenes, creating dynamic light sources, moving the 3D models and supporting functionality tests. ARtalet (Ha et al., 2010), an authoring tool for augmented digital books where the user can add 3D elements, such as objects and animated characters over the pages using a TUI. AR-jig (Anabuki & Ishii, 2007), a handheld TUI to support 3D modelling by replicating a physical curve as a tool in the digital space. ARGroove (Billinghurst et al., 2008), a AR platform for the creation of music where TUI used to control sound loops, filters and effects in a composition. Furthermore, Ha et al. (2012) used computer vision to track the path of TUI in the physical space and animate the spatial movement of 3D content in an AR environment.

The use of TUI in mixed prototyping can allow a more interactive augmented experience where some functions can be simulated and digital elements can be adjusted directly on the object. From this perspective, methods allowing users to integrate those functions easily can improve the content authoring process. However, their capacity to edit digital elements is typically limited to changes in placement and simple geometrical modifications. Therefore, they provide limited support if the 3D model of the mixed prototype is not already available.

3.2.3 Immersive authoring

Closely related to the use of TUI, another trend for MR content creation has been the utilisation of immersive authoring. This authoring method is based on the logic of “what you experience is what you get” (G. A. Lee et al., 2005). Hence, creating the MR experiences directly in the medium where it will be experienced rather than moving back and forth between a 2D screen and an AR display each time a change is made. One of the main benefits of this method is its capacity to allow users to concurrently design and test the MR experiences (Nebeling et al., 2020).

Due to the popularity of these methods, several examples can be found in the literature. These developments include multiple research projects that proposed smartphone-based authoring tools to create in-situ mobile AR experiences (Z. Chen et al., 2020; Jung et al., 2012; Y. Yang et al., 2016). Yu et al. (2017) use hand tracking and a head-mounted display to allow authors to place, interact and animate the digital content within an AR environment. LibrARy (Ifrim et al., 2021), a platform to enrich cultural places that allow the creation of collaborative spaces where users can execute augmented interactions through 3D virtual objects. XRDirector (Nebeling et al., 2020), a role-based collaborative immersive authoring system that allows designers to interact using AR and VR devices as puppets to manipulate virtual objects in 3D physical space. Furthermore, 360theater (Speicher et al., 2021), a tool for rapid prototyping of MR experiences, takes physical dioramas into a virtual space by enhancing 360° video capture with 3D models and simulating spatial interactions.

Moreover, within the immersive authoring methods, we can find platforms that simulate where the MR experience will be used. Hence, allowing authors to work in a space like the real one but with the virtual environment flexibility. Some of these methods include Corsican Twin (Prouzeau et al., 2020), a tool to support the authoring in-situ augmented reality data visualisations in virtual reality using digital twins. CAVE-AR (Cavallo & Forbes, 2019), uses a CAVE (Cave Automatic Virtual Environment) to simulate the expected conditions where the AR will be experienced, including tracking information, geographical information, architectural features, and sensor data. And ScalAR (Qian et al., 2022), a platform that enable designers to author semantically adaptive AR experiences in Virtual Reality.

These methods can support the creation of a wide variety of augmented experiences and content. However, they mostly rely on pre-made content to create such experiences. Furthermore, although some of these methods can create 3D models of objects or physical spaces, they are not intended to create accurate replicas of their physical counterparts. Hence, offering insufficient support to create mixed prototypes.

3.2.4 Creation through demonstration

Another trend that can be found in MR content authoring is the creation of content through user demonstration. These methods record the user actions and the environment through different sensors to extract relevant objects and interactions. The gathered information is then used to automatically create an MR experience to support the repetition of this process, hence being especially useful for supporting tutorials, maintenance, and assembly procedures. The main benefit of these methods is the similarity with the actual procedure to be supported, allowing a more natural authoring process that drastically reduces the need for manual creation and adjustment of the MR interactions.

Among the examples of these methods, we can find several supported by RGB cameras. For example, AREDA (Bhattacharya & Winer, 2019) uses computer vision to analyse an assembly process conducted by an expert and separate it into steps that can be easily edited to create an AR experience. ProcessAR (Chidambaram et al., 2021), a system supported by computer vision and eye-tracking to identify the objects and tools the author is looking at during the realisation of a procedure. TutorialLens (Kong et al., 2021), a platform for creating AR-supported tutorials by tracking the user’s interactions with physical interfaces while the actions are being narrated. Furthermore, the work of Van Lopik et al. (2020) proposed a template-based interface to support AR content creation on shop floors using the camera to record the user’s point of view and the corresponding interactions.

Moreover, with the increased availability of cameras with depth-sensing capabilities, multiple developments have opted to use the additional information acquired from the demonstrated processes to support the authoring process. Some examples include SpatialProto (Müller et al., 2021). This platform records users’ demonstrations using physical objects, which later are extracted from the background to create a digital replay of their movements in MR. In addition, Stanescu et al. (2022) use the provided depth information to detect volumetric changes related to different states of a product during an assembly demonstration. Moreover, Zhang et al. (2022) propose a method to support MR experiments demonstrations that registers hand movements during interactions and uses machine learning to create adaptive animations when grabbing objects.

Finally, examples supported by immersive authoring can be found, such as GhostAR (Cao et al., 2019), a platform for creating and editing human-robot collaborative tasks through demonstrative role-playing in AR.

In mixed prototyping, these methods could help develop and test complex interactions between the final user and the product and help designers define user interfaces’ placement. However, they would offer limited support in defining other product characteristics.

3.2.5 Retargeting existing data

When dealing with material to support procedures such as assembly, we can find many formats, from printed documentation to videos and MR experiences. This supporting material is typically created for its specific medium, requiring a complete remake when changing the information to another format. In this context, and due to the high availability of the material in older formats, a proposed method to ease MR content authoring has been retargeting existing data to MR experiences.

Some efforts in this direction, focused on the communication aspect, include the work of Gatullo et al. (2017), which proposed a method to digitalise technical documentation using language processing to distinguish tasks and convert them to AR graphical instructions. Moreover, Scurati et al. (2018) identified the most frequent maintenance actions used in manuals and converted them into graphical symbols for AR experiences.

Examples of a direct transformation of format are the work of Mohr et al. (2017), which proposed a method to retarget video tutorials to MR experiences, showing different interactions in the 3D space. Moreover, 360proto (Nebeling & Madier, 2019), presents a platform that allows users to capture sketched environments and user interfaces, and visualise them in 3D space where the interactions can be created.

However, since, in many cases, 3D models are also available to support the MR experience; several methods have used the retargeted data to guide the interactions. Examples of this are the method proposed by Mohr et al. (2015), which uses as an input a CAD model and printed documentation and extracts from it the different steps and positions of the object's parts using computer vision to create a MR experience. Salonen et al. (2009) proposed a data pipeline to convert CAD data to AR assembly instructions using markers. ARComposer (Shekhar et al., 2019), a platform that uses text as an input to retrieve 3D models from a database and create MR animations composed of multiple objects with diverse relationships to each other. Wang et al. (2013), proposed a method to reuse assembly-related information generated during design (formalised as an IDEF0 graph) to create MR experiences. Furthermore, Zogopoulos et al. (2022) proposed a method to identify disassembly sequences using a CAD model as an input and create AR experiences with reduced user intervention.

These methods ease the authoring processes of augmented experiences where information is available beforehand. Therefore, implementing similar solutions in mixed prototyping, where some characteristics are not necessarily defined, could be challenging. Nevertheless, this could be an alternative if the developed product comprises standardised components.

3.2.6 3D models preparation

The last authoring trend in MR found during the literature review relates to methods focused on supporting the creation of 3D models to be used in MR experiences. Two of the main benefits of this approach are the reduction of the dependency on an existing database of 3D models to create the experiences and the reduction of 3D modelling skills to prepare custom content.

Efforts in this direction include methods that can transform 2D content into 3D models usable in MR experiences. Among these methods, Bergig et al. (2009) and Hagbi et al. (2010) used computer vision to analyse sketches of simple geometrical shapes, generate a corresponding 3D model visualisation in AR, and detect graphical symbols to automate the creation of interactions among objects. MagicToon (Feng et al., 2017) proposed a system developed for children capable of generating textured cartoon 3D models of objects and characters starting from 2D sketches and the capacity to animate them using a simple user interface. Furthermore, Yoo & Lee (2014) proposed a method supported by touchscreen input where gestures resembling primitive geometries were used to add such geometries to the MR space.

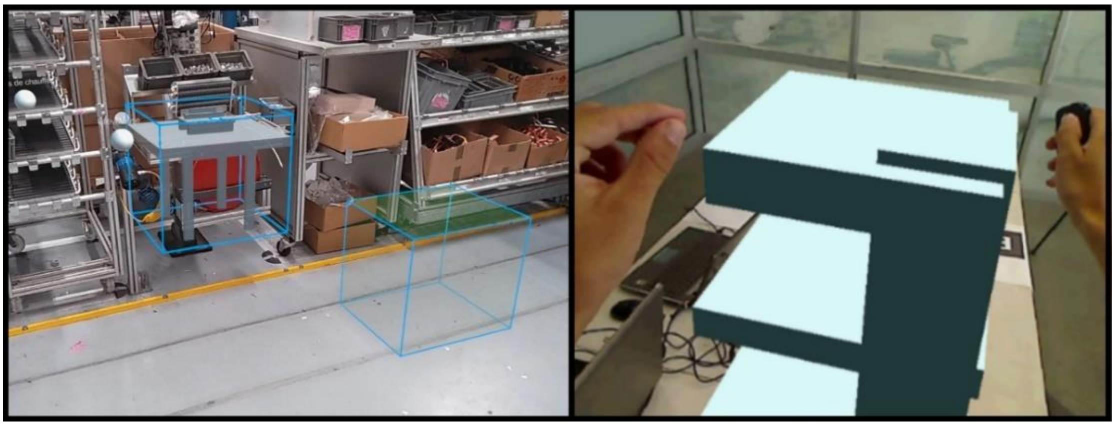


Figure 3.2 - WAAT by Bégout et al. (2020) (left), and AIR MODELLING by Arroyave-Tobón et al. (2015) (right)

Nevertheless, most of the methods found were developed with an immersive authoring approach, allowing users to create and modify the 3D models in the MR space. Some examples include the work of Langlotz et al. (2012), which proposed using natural feature detection to simulate a planar working surface where an intersection point projected from a handheld device can be used to draw shapes and use them to create simple 3D models. Mobi3DSketch (Kwan & Fu, 2019) and project PRONTO (Leiva et al., 2020), two mobile applications that use multiple sensors to recognise the device's position in the 3D space and allow users to create interactive 3D sketches in situ. The work of Tang et al. (2015), a platform supported by markers, where each of them represented a primitive 3D geometry that could be duplicated and placed in the 3D space to create more complex 3D objects. Arroyave-Tobón et al. (2015) and Han et al. (2020) proposed methods to support prototyping during early design stages, which used depth-sensing cameras and computer vision to recognise hand-gestures and used as the interaction means to create and modify 3D models in the 3D space. Furthermore, WAAT (Bégout et al., 2020) proposed a platform for quickly creating and dynamically modifying simple 3D models of workstations in MR and testing the placement of such objects for guided experiences (See figure 3.1).

Whitin these methods, some have taken advantage of surface reconstruction technologies to gather extra data about the environment and use it as a reference during the modelling process. In this direction, Wan et al. proposed MRStudio (Wan et al., 2011), a platform for aircraft cockpit designs that generated a point cloud reconstruction of the environment to help them place relevant and create 3D models and graphical elements. Yang et al. (2013) integrated several surface reconstruction methods, such as Structure From Motion and Clustering Views for Multi-View Stereo, to allow a reconstruction of objects and environments that could then be visualised in MR. Izadi et al. (2011) proposed KinectFusion, a platform that used depth information to reconstruct objects and the environment to support the creation of physically accurate interactions with the environment. Finally, Huo et al. (2016) developed Window-Shaping, a mobile interface for design ideation that gathers the environment's geometrical data to allow the direct creation of 3D shapes on and around physical objects (See figure 3.3).

Figure 3.3 - mobile surface reconstruction system by Yang et al. (2013) (left), and Window-shaping by Huo et al. (2016) (right)

Within the studied trends, the methods focused on creating 3D models have a high potential to support mixed prototyping content authoring, as one of the main requirements is the availability of a 3D model replicating the physical counterpart. However, the ability of most methods to execute this task is limited to simple geometries, making them unsuitable for mixed prototyping. Moreover, although some of the presented methods integrate surface reconstruction algorithms, allowing them to generate more accurate replicas, the 3D models were mainly used for visualisation or support physically accurate interactions. Hence, they do not include the preparation of the models for more complex interactions as the ones of a mixed prototype.

3.3 Segmentation methods

Mesh segmentation is not a mandatory requirement of 3D models to be used in a mixed prototyping platform. However, it is essential if users need to edit various parts of the prototype independently. By segmenting the 3D model, users can apply distinct treatments to each part, such as altering surface characteristics like color, smoothness, glossiness, specularity, or metalness. Additionally, if the mixed prototyping platform allows it, users can utilize this characteristic to distinguish different interaction modalities.

This section presents an overview of segmentation methods applied to 3D meshes to understand their limitations in mixed prototyping and how they can be used as a base for new developments. Moreover, 2D mesh segmentation methods are also evaluated due to the availability of texture information when the 3D model is generated through surface reconstruction methods.

3.3.1 Mesh segmentation and UV mapping

The mesh segmentation and UV mapping processes are not unique to the mixed prototyping workflow, and many applications with 3D rendering capabilities require this step in their content authoring workflow. Thus, a wide variety of methods can be found in the literature. The existent mesh segmentation algorithms can be categorised as surface-based and part-based (Shamir, 2008) (see figure 3.4). On the one hand, surface-based segmentation methods aim to optimise some characteristics of the mesh or the corresponding parametrisation, such as reducing the distortion of the texture mapping. On the other hand, part-based segmentation methods aim to decompose the input mesh into meaningful parts based on human perception. Considering this, the latter is more suitable for mixed prototyping as it would allow for the distinction of the different parts of a product, the same that must be modified independently during a design session.

Figure 3.4 - Example of part-based segmentation (Y. Lee et al., 2005) (left) and surface-based segmentation (Sander et al., 2001)(right)

Initially, most of these algorithms directly used geometrical features in the mesh as an input to separate them into meaningful components (Katz & Tal, 2003; Rodrigues et al., 2015). More recently, there has been an increased focus on machine learning-based algorithms that are supported by labelled data and comparisons from multiple objects of the same category to achieve better results (George et al., 2018; Shu et al., 2022; W. Tang & Qiu, 2021). These algorithms demonstrated excellent performance on well-established datasets such as the Princeton segmentation benchmark (X. Chen et al., 2009), which includes 3D models of products. However, as most of them rely on the use of geometric information to segment the 3D model, either directly or indirectly, they are not entirely suitable for mixed prototyping, at least in two opposite circumstances:

- When two or more geometrical elements of the object must remain unsegmented to be treated as a unique part, e.g., the housing of a device that will be manufactured using the same material.
- When one geometrical element of the object must be segmented to be treated as independent parts, e.g., a cylindrical bottle composed of two different materials.

Moreover, although the Princeton Segmentation Benchmarks (X. Chen et al., 2009) and similar datasets such as Thingi10K (Q. Zhou & Jacobson, 2016), offer a great variety of object categories, they are also low-resolution with elementary geometries. They, therefore, have a limited capacity to reflect the actual behaviour of the algorithms on reconstructed surface models such as the ones used in mixed prototyping. As a result, most of the mesh segmentation methods found do not consider the specific requirements of these 3D models nor take advantage of the extra information they can provide through the reconstructed texture.

When dealing with texture information, UV mapping methods are typically the ones to be used. Some have been specifically developed and tested to deal with reconstructed surface models (Lévy et al., 2002; Maggiordomo et al., 2021). However, they mainly focus on reducing the deformation of the parametrisation rather than segmenting the models in a meaningful way. During the state-of-the-art analysis, the only work that has reported carrying out a part-based segmentation using the texture as an input is in the area of cultural heritage (Inzerillo et al., 2019). Nevertheless, in the previously mentioned work, the texture features used for the model segmentation were not extracted by automated means. Instead, the user selected the segments before proceeding with the automated segmentation and UV mapping process.

All these limitations have created an increased demand for robust techniques designed to process and manipulate reconstructed surface textured 3D models (Maggiordomo et al., 2020). In this context, tools derived from computer vision and image segmentation are suitable solutions due to the availability of new texture information that can be exploited as valuable input for segmentation. The following section will describe image segmentation methods and how their capabilities relate to the specific requirements of textures of reconstructed surface models.

3.3.2 Image segmentation

Image segmentation is "the partition of an image into a set of nonoverlapping regions whose union is the entire image" (Haralick & Shapiro, 1992). During the last decades, there has been an increasing interest in its research and development. One of the main reasons for this is that, as pointed out by Guo et al. (2018) "have the potential to make major contributions across the wide field of visual understanding from image classification to image synthesis; from object recognition to object modelling; from high-performance indexing to relevance feedback and interactive search".

There are more than a thousand different methods published (Mikes & Haindl, 2021), and similar to the case of mesh segmentation, the main application field for the segmentation of reconstructed surface models is in the area of cultural heritage. Here, the automatic recognition of patterns poses a non-trivial problem that can be tackled with image-based methods operating on a 2D parametrisation of the mesh (Lengauer et al., 2021).

Considering the characteristics of textures from reconstructed surface models (see section 2.3.5), although texture segmentation methods are powerful tools, they are not well suited for the segmentation of mixed prototypes. Nevertheless, other tools from the area of computer vision still present potential solutions.

As an alternative to detecting segment boundaries, contour detection methods can be used, a process that identifies the edges of an image. Unlike texture segmentation methods, these offer no guarantee that they will produce closed contours and hence do not necessarily provide a partition of the image into regions (Arbeláez et al., 2011). Nevertheless, due to the surface characteristics of mixed prototypes (See section 2.1.4), they still could be helpful to detect relevant features in places where there are changes in materials or other features that change the texture colours. For an overview of edge detection methods, please refer to the work made by Ghosh et al. (2020)

Finally, another alternative for the segmentation of mixed prototypes are thresholding methods (Sezgin & Sankur, 2004). Although not explicitly developed for this purpose, they could work similarly to a contour detection method if the background is homogeneous and there is high contrast with the contours to be extracted. These methods define a threshold value, and if a pixel value is smaller than the threshold, it is set to 0 or black; otherwise, it is set to a maximum value or white. If the threshold value is the same for all the pixels in the image or is recalculated for smaller image regions, thresholding methods can be categorised as global or local/adaptive, respectively.

In principle, most of the texture segmentation, contour detection and threshold methods could be implemented to get helpful information that can be then transferred to the 3D model to be segmented. Nevertheless, the object's specific characteristics must be considered, as the texture features can widely change between object categories.

3.4 Critical analysis and research questions

3.4.1 Content authoring methods for mixed prototyping

There has been an evolution in the methods applied to support content authoring for MR experiences. Several approaches have tackled the problem of technological adoption from different perspectives based on their objectives. In the beginning, these methods were primarily focused on aiding the execution of low-level tasks, such as programming and evaluating the impact of new interaction modalities, such as the use of TUI and immersive authoring. However, now the focus has shifted to developing content authoring methods for specific applications that, although more limited in their application, could better address the needs of those applications. Some of these methods were applied in the product development process and supported the creation of prototypes through MR experiences.

In this context, due to the ability to have spatial cues while designing and allowing the user to see their ideas placed in a physical environment, several methods were tailored to the creation of 3D models. Nevertheless, the complexity of such models has been limited to the use of primitive geometries and the creation of other simple shapes. This limitation is acceptable within the idea generation process, where low-fidelity models still can provide helpful information to take design decisions (Liker & Pereira, 2018), or in applications where the 3D models do not need to match a physical counterpart perfectly. However, these models are not appropriate to support the creation of mixed prototypes, where the virtual and physical prototypes must be geometrically equal.

An approach found within the literature to support the creation of 3D models that match the geometry of physical objects was the use of surface reconstruction methods. However, this technology has been primarily used to allow realistic interactions with the environment when applied within MR experiences. Furthermore, in the cases where it is applied to replicate the geometry of physical objects, its use has been limited to the visualisation of digital replicas. While this is a step forward to support the authoring of mixed prototypes, the post-processing of such models to allow interactions, such as changes in the colour of different segments or placement of graphical elements on the surface, has not been addressed. Moreover, surface reconstruction technologies have reached a high level of maturity, and off-the-shelf solutions can quickly obtain high-quality 3D models of physical objects (Gebler et al., 2021). Therefore, although multiple developments supported the authoring of MR experiences, there is still a need for methods to aid the preparation of the 3D models to be used in mixed prototyping. In particular, focusing on the segmentation and UV mapping of such models, as these processes work as the base to integrate the capability to edit the digital content on the surface of mixed prototypes in real-time, a key feature to support design activities.

As shown in the second part of the literature review focused on mesh segmentation methods, several solutions exist with the capability to execute this task automatically. Nevertheless, due to the specific characteristics needed for mixed prototyping (i.e., the resulting segments must correspond to the parts that will be edited independently during design sessions), these methods are unsuitable to support the content authoring process. Considering the lack of suitable mesh segmentation methods and the availability of new texture information from reconstructed surface models that can be exploited as valuable input for segmentation processing, tools derived from computer vision and image segmentation are suitable solutions.

Additionally, it is also relevant to define the type of surface features to use as input for the segmentation process, as they can directly affect the quality of the results. While surface features specifically added for segmentation could produce more controlled results, they would require the user to modify the physical prototype previously. On the contrary, if existing features were to be used, the user could directly focus on the segmentation of the model, however, with less control over the resulting segmentation and potentially worst results. Regardless, both alternatives are viable options that can reduce the required knowledge in 3D modelling to carry out the segmentation.

In consequence, to evaluate the technical suitability of this approach to support the mixed prototyping content authoring process, the first research question is:

RQ1: Can a computer vision supported process generate a segmented 3D model suitable for mixed prototyping?

> *RQ1.1: using added graphical mark on the surface of the physical prototype.*
>
> *RQ1.2: using existing features on the surface of the physical prototype.*

3.4.2 Mesh segmentation

Several methods exist with the capability to segment 3D models. Nevertheless, it was found that they were not suitable to support the mixed prototyping authoring process because the results would not match the segmentation of the prototype parts needed by the designers. To solve this limitation, a computer-vision-supported approach is being proposed to automate this process. The first research question addresses whether or not it will accomplish its objective. However, as in any other automated process, it is essential to understand that the obtained ease of use has an associated cost related to reduced control over the results. Hence, the implications of this change and how its results compare to current alternatives should be assessed.

Since previous methods for mesh segmentation have different objectives than the one proposed, a comparison against their results would provide limited information.

Nevertheless, the current method for mesh segmentation of reconstructed surface models used in mixed prototyping can still provide helpful reference information for comparison. Due to the lack of existing tools to support this process, it is currently done by the designers or developers using the standard tools provided by existing 3D modelling software. In consequence, the second research question is:

RQ2: How do the results of a texture-based mesh segmentation compare in terms of quality to a manual segmentation?

3.4.3 Technological adoption

As seen through the analysis of existing MR content authoring methods, multiple efforts have been made to support its technological adoption. Moreover, although it has not reached a generalised adoption, it is evident that mixed reality applications have reached an acceptable level of maturity in some contexts, such as assembly and maintenance (de Souza Cardoso et al., 2020). Nevertheless, product design has not been the case, and some barriers remain (Bhattacharya & Winer, 2019; Giunta et al., 2019; Masood & Egger, 2020). Many of the discussed authoring methods focus on reducing the competencies required to take advantage of the benefits of MR platforms. However, these competencies were highly dependent on the type and objective of the supported MR experience.

In mixed prototyping, one of the primary competence requirements, other than the specifics of each MR platform, is creating 3D models to be used during design sessions. Although a method is being proposed to support this process, it is unclear how it will impact the technological adoption of mixed prototyping. Furthermore, the process of technology adoption and the related innovation resistance is a multifactorial problem that can be affected by innovation characteristics, individual characteristics and external factors (D. Huang et al., 2021), hindering the evaluation of the impact of the solution in this regard. Nevertheless, some factors can still be evaluated due to their correlation to technological adoption. Some of these factors include the previously mentioned requirement of competence 3D modelling, the time required to conduct the segmentation process, and the ease of use of the proposed solution compared to the current method. Following this logic, the third research question is:

RQ3: Does a content authoring process supported by computer vision improve user acceptance of mixed prototyping preparation?

Chapter 4: Automated mixed prototyping content authoring process

This section presents an integrated workflow of content authoring for mixed prototyping spanning from the availability of a physical prototype to the exporting of a 3D model suitable for mixed prototyping that is appropriately segmented and UV mapped to be used in mixed prototyping platforms. Additionally, two implementations of the algorithm are presented, the first being an add-on for an existing 3D modelling software and the second a standalone application.

4.1 Content authoring workflow

As mentioned in the previous section, the segmentation and UV mapping of the 3D model is a required step for virtually any system with 3D rendering capabilities. However, since each platform for mixed prototyping still can have its workflow, some steps of the procedure described here, not strictly related to segmentation and UV mapping, could vary. To have a realistic view of the implementation, some workflow steps have been adapted to the SPARK platform requirements (See section 2.1).

Figure 4.1 shows a diagram of the proposed content authoring workflow with the main steps in the process. First, the physical prototype preparation ensures that the object has texture information that can aid the segmentation. Next, surface reconstruction creates a digital replica of the object with its texture information. The model is prepared for the projection system by applying the required transformations (e.g., rotation, location, or scale). Then, a feature detection algorithm is used to extract the information related to the boundaries of the segments from the original texture and uses it to select the corresponding vertices in the 3D mesh. Finally, the input 3D model is segmented using that information, and new UV maps are generated for each segment.

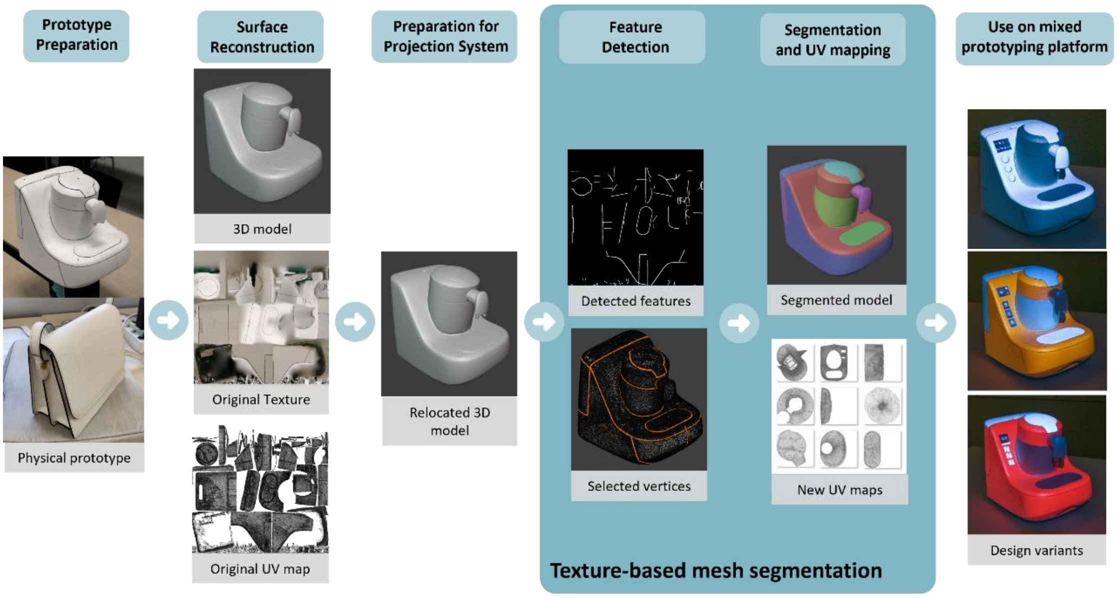


Figure 4.1 - Proposed content authoring workflow

In the following subsections, each workflow step is explained in more detail.

4.1.1 Physical prototype preparation

Considering that a physical prototype is already available, there are two potential paths: modify it to comply with the requirements of the tool or use features naturally available in the prototype as input to generate the segmentation (See figure 4.2).

Figure 4.2 - Physical prototypes with added features (left) and natural features (right)

In the first case, the surface should be prepared to reproduce the projected virtual information as intended in terms of colour accuracy. To achieve this, a homogeneous paint coat of a light neutral colour such as white or grey should be applied. It is worth mentioning that some solutions to calibrate the colour reproduction independently of the surface colour have been proposed (Morosi & Caruso, 2020). Nevertheless, proper surface preparation reduces the need for these corrective measures.

After the surface has been prepared, extra information to aid the segmentation and UV mapping can be added to the surface manually. More specifically, black lines are manually drawn over the physical prototype, marking each independent part's boundaries. Particular attention should be put into ensuring the generation of closed boundaries to avoid contiguous parts being detected as one.

In the second case, a physical prototype should be selected taking in consideration analogous requirements, that is (1) a neutral base colour such as white or grey, and (2) the existence of high contrast boundary features that generated closed segments to be edited individually in the mixed prototyping platform. This approach is especially useful in cases where the prototype is already available but cannot be modified due to user requirements.

Moreover, depending on the mixed prototyping platform, other information could be added at this stage. For example, in the case of the SPARK platform, retroreflective infrared markers needed for tracking must be applied to get a reference of their position in the 3D model and ease calibration.

4.1.2 Surface reconstruction

Once the physical prototype has been prepared, a 3D model of it must be generated to have a digital replica that can then be modified and projected over its surface to change its appearance. Several techniques and off-the-shelf 3D scanners are suitable for this process. However, the main requirement is that the selected method can generate a corresponding texture for the 3D model. This information will be needed to correlate the added surface features to the vertices and polygons in the 3D model. Photogrammetry and Structured light scanning methods are suitable for this task as several commercial software are available, allowing the creation of highly accurate 3D models of prototypes in restricted timeframes (Gebler et al., 2021).

The specific characteristics of the texture and relative UV map will depend on the input data used to reconstruct the geometry and the software used, which could affect the quality of the output mesh segmentation. Nevertheless, this variability is addressed during the following steps.

4.1.3 Preparation for projection system

Using the results from the surface reconstruction, the rotation, scale, and location of the 3D model must be adjusted to match the physical prototype dimensions and initial position. Moreover, depending on the tracking solution to be used in the mixed prototyping platform, the origin (i.e., the point in the 3D space that defines the location of the 3D model) could also require additional adjustments. These changes can be done either during the surface reconstruction or later in a 3D modelling software using basic functions.

This step can change the position of all the vertices and polygons on the input 3D model. Nevertheless, as long as the modifications are applied equally to the complete 3D model, they will not affect the results of the texture-based mesh segmentation.

For the specific case of the SPARK platform, the physical prototype with the infrared retroreflective markers must also be calibrated using tracking software and a set of infrared cameras.

4.1.4 Extraction of texture features

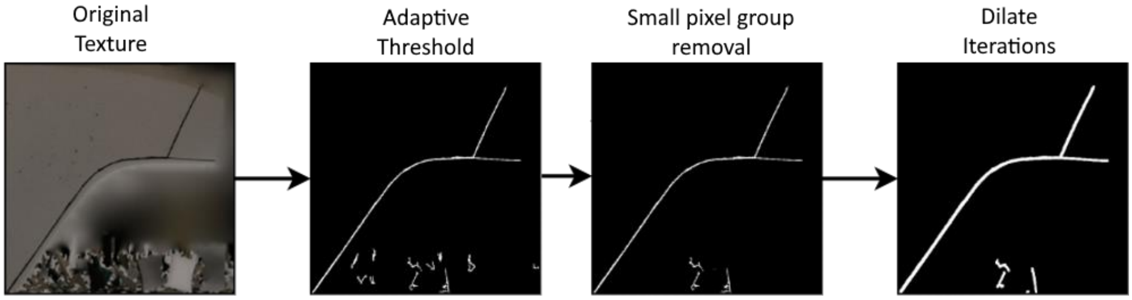


Figure 4.3 - Processing steps to extract boundary features from original texture

This step comprises three successive sub-steps applied over the original texture (Figure 4.3). First, all the high-contrast features on the original texture are extracted using an adaptive threshold function presented in section 3.3.2. The rationale for the choice of this method over the other image segmentation techniques can be summarised with the following points:

1. Independent to repeating textures: By using the boundaries of the parts to be segmented rather than their texture, if several segments have the same texture, they can still be distinguished.
2. Independent to fragmentation: Even if the detected boundaries of a segment are discontinuous in the 2D texture due to fragmentation, they will remain connected in the 3D model. This information can be retrieved using the UV map parametrization.
3. Independent to inpainting artifacts: If multiple features are incorrectly connected to different parts during the digital inpainting process, these errors will be discarded when transferring the information to the 3D model, as detected features in the texture that are outside the parametrization will not correspond to any face of the 3D model.
4. Independent to illumination changes: Some surface reconstruction methods could include in the texture the effect of the ambient illumination, making some parts of the texture darker than others. Since adaptive thresholding analyses small groups of pixels at a time, gradual changes in colour are not falsely detected as boundaries.

After applying the adaptive threshold, a filter removes any group of connected pixels below a fixed number of pixels to reduce the texture elements not corresponding to boundary features. Finally, the selection is expanded to ensure the connection of boundary features that were not wholly recognised during surface reconstruction or incomplete in the physical object. This can be implemented through a dilate function, which considers any existing white pixel and converts all the surrounding pixels to white, allowing to repeat this operation multiple times. The kernel size (i.e., the size of the pixel area used to define the results of the adaptive threshold) and the number of dilating iterations are provided as parameters that can be modified to improve the feature detection (See Figure 4.4).

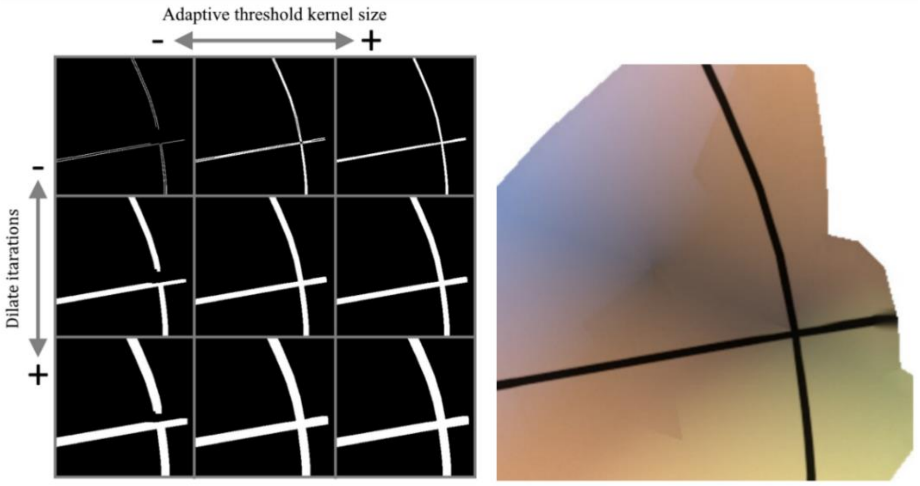


Figure 4.4 - Results of feature detection using different parameter combinations

Once the algorithm extracts the features from the original texture, the UV map is used to correlate the position of each element in the 2D texture to 3D model vertices.

Below, the pseudocode for the extraction of texture features is presented:

Data: Texture from surface reconstructed model (*T)* and its corresponding UV map (*UV*).
Input parameters: Size of adaptive threshold kernel (*at)*, number of dilate iterations (*di)* and number of pixels for small object removal (*p)*.
Results: List of polygons inside of the detected texture features.

Procedure:

```
# Feature extraction
get T
convert T to grayscale
apply adaptive threshold to T using at
get list of connected pixels and quantity of pixels on each of them
for each group of connected pixels
    if group of connected pixels < p then
        delete group of connected pixels
dilate the remaining groups of connected pixels di times

# Selection of faces in extracted features
get UV
for each face in UV
    for each vertex in face
        convert vertex coordinates to pixel coordinates
        if all vertex in face are inside of extracted features then
            save the index of the face
```

4.1.5 Segmentation of 3D model and UV mapping

The algorithm segments all the polygons of the 3D model that are placed inside the detected texture features. This process generates several 3D meshes, which, as shown in Figure 4.5, will be separated into two categories for processing purposes. If the 3D mesh is composed of multiple polygons, it is assumed that it corresponds to the surface of a part of the prototype and will be called primary mesh. Therefore, the number of primary meshes equals the number of segmented parts in the result. On the other hand, if the 3D mesh is composed of an individual polygon, it is assumed that it was positioned inside a line and will be called secondary mesh.

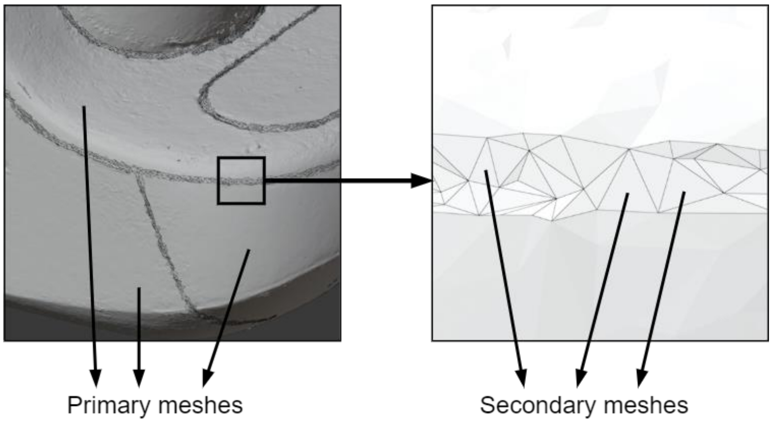


Figure 4.5 - Categorization of segmented meshes

To avoid that small primary meshes generated by imperfections in the selection are considered independent parts of the prototype, the algorithm allows users to control the maximum number of the primary meshes. This action can be done by selecting the number of final parts beforehand and keeping the primary meshes with the highest number of polygons or by setting the minimum number of polygons that a primary mesh should have to be maintained. Any primary mesh excluded through this process is then separated into individual polygons; each is considered a secondary mesh.

After the classification, each secondary mesh is assigned and joined to the closest primary mesh based on the distance between their boundaries. As a result of this process, the 3D model will be separated into multiple parts with reduced geometric complexity compared to the initial 3D model. Finally, automated UV mapping solutions can be applied for each primary mesh, and the complete model can be exported to a mixed prototyping platform.

Below, the pseudocode for the mesh segmentation and UV mapping is presented:

Data: 3D model (*M)* and a list of face indexes (*F*) selected during the extraction of texture features
Input parameters: Number of segments (*s*)
Results: Segmented 3D model with new UV map for each segment

Procedure:

```
# Definition of mesh type
primary_meshes = faces in $M$ not included in $F$
secondary_meshes = faces in $M$ included in $F$
get list of segments in primary_meshes
get the number of faces of each segment
order segments by their number of faces
for each segment in primary_meshes:
    if segment within the first $s$ segments with most faces, then
        keep in the list of primary_meshes
    else
        move segment to secondary_meshes

# Assignment of faces in secondary_meshes to segments in primary_meshes
b_pm = list of coordinates of vertices in the boundaries of the primary_meshes
f_sm = list of coordinates of vertices in each face of the secondary_meshes
calculate the distance between vertices in b_pm and f_sm
for each face in secondary_meshes
    join to the closest segment of the primary_meshes

#UV mapping
for each segment in primary_meshes
    calculate new UV map

#Export
export 3D model with all segments of primary meshes and UV maps
```

4.1.6 Use in mixed prototyping platform

Once the 3D model is segmented, and the UV map for each segment has been created, it can be imported into a mixed prototyping platform. For the case of the SPARK platform, this process is carried out through an Information System that manages all the digital assets, including the 3D models, textures, and other graphical elements to be projected over the mixed prototype.

First, a design session is created, and all the digital assets to be used are assigned to that session. Then, a virtual prototype is created, and the session is started. After this is done, a SAR system accesses the data in the Information System, and the projection is calibrated. Then, a graphical user interface is presented where each segment can be selected (See Figure 4.6). Finally, their surface characteristics can be changed, with each modification simultaneously applied to the mixed prototype. Figure 4.7 shows multiple design variants achieved using this workflow.

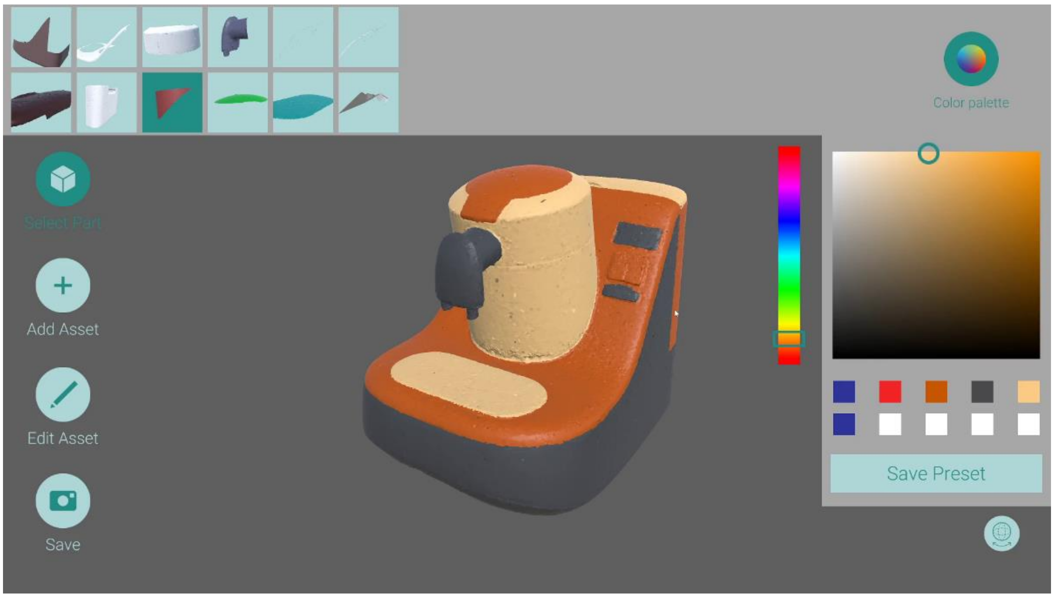


Figure 4.6 - GUI of SPARK platform during Mixed Prototype editing

Figure 4.7 - Design variants generated using the same Mixed Prototype

4.2 Developed tools

This section presents the developed tools to support the execution of the proposed workflow for mixed prototyping content authoring.

Although the proposed workflow considers tasks that span from the availability of a physical prototype to the exporting of a 3D model suitable for mixed prototyping, some of those tasks must be done manually or can be executed by readily available solutions. Because of these reasons, the developed tools only consider the steps from preparing the 3D model for the projection system to exporting to mixed prototyping platforms.

In the following sections, two implementations of the supporting tool are presented with instructions on how they work, their main benefits and their limitations. The first implementation was integrated within an existing 3D modelling software, making all its additional functions available to the user. Conversely, the second implementation was developed as a standalone, more specialised and optimised application, but also limited in terms of functions.

It is worth noting that although a demonstration is already shown in this chapter for explanation purposes, chapter 5, which focuses on technical suitability, will have a more in-depth analysis of the obtained results.

4.2.1 Blender add-on

The first tool was developed assuming the user knows at least how to import a 3D model into a 3D modelling software and navigate the viewport. More specifically, an add-on was developed for Blender, a free and open-source 3D modelling software, being programmed mainly using an internal implementation of Python as well as functions of OpenCV (Bradski, 2000), an external library for computer vision. This 3D modelling software was selected for development due to multiple reasons. First, it offers a perfect platform for fast prototyping, providing several add-on templates. At the same time, parts of the process can be manually executed by an experienced user while the underlying programming functions are recorded, easing the writing of more complex scripts. Furthermore, the author already had extensive experience using it, allowing him to focus on development rather than learning the software.

4.2.1.1 Utilization

As shown in figure 4.6, the add-on was integrated as part of Blender user interface. It was composed of 6 operations designed to be applied sequentially to a selected textured 3D model (outlined in orange). These functions were separated into two categories: the alignment tools, which focused on preparing the 3D model for the tracking system, and the segmentation workflow, which focused on all the operations needed to segment the input model using the provided texture information.

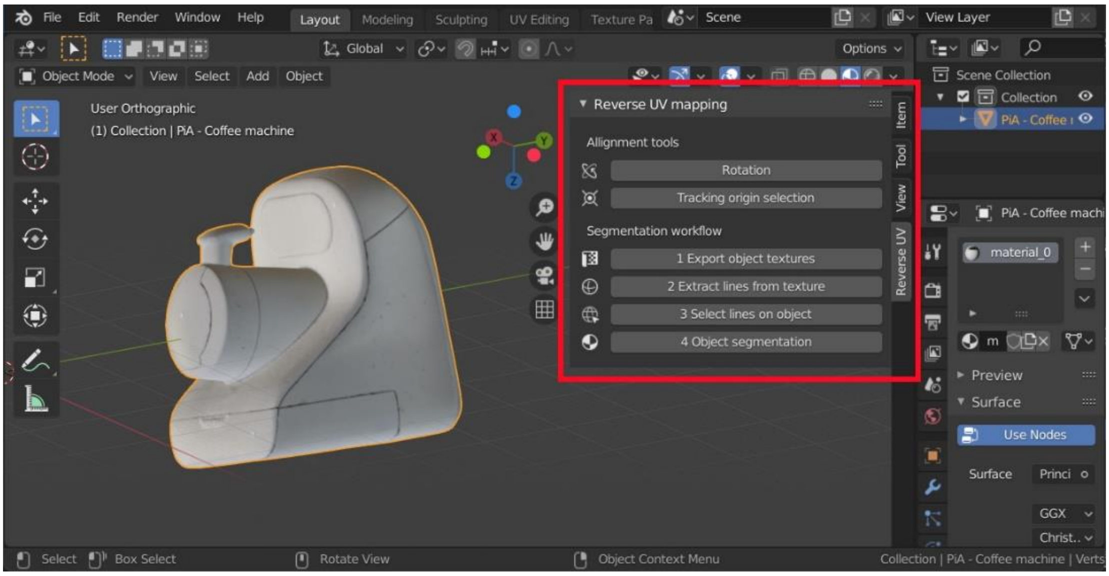
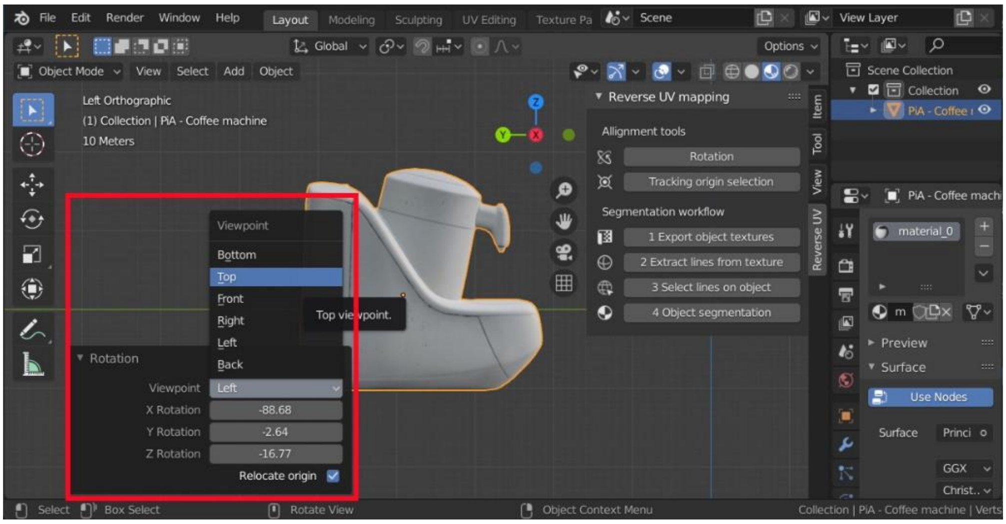


Figure 4.8 - User interface of Blender with the developed add-on highligthed in red

The first step before using the add-on is importing the 3D model with its corresponding texture using the default functions of Blender. Additionally, the project must be saved, as the add-on uses its directory path to output the results, log files and intermediate processing files.

As shown in figure 4.7, the rotation function automatically changes the 3D viewport to an orthographic perspective with a list of viewpoints to select and allows the user to rotate the 3D model in the X, Y and Z axis. In addition, this function automatically places the 3D model in the centre of the screen and relocates the origin of the 3D model (i.e., a point in the 3D space that defines the location of the 3D model). This allows the users to align the object's base with the corresponding plane, obtaining a 3D model with the same orientation as the physical prototype over a surface. Although each of the actions involved in this function is simple to execute with the default functions of Blender, they were included together to aid users with no experience in 3D modelling.


Rotation

Viewpoint: Left

X Rotation: -88.68

Y Rotation: -2.64

Z Rotation: -16.77

Relocate origin ☑

Figure 4.9 - Blender add-on with rotation function highligthed in red

It is worth noting that although the scale adjustment is relevant for preparing the 3D model for the projection system, surface reconstruction software already considers this operation. Hence, it was decided not to include it as part of the add-on.

The tracking origin selection function was added explicitly due to the requirements of the SPARK project, where the origin should be equal to one of the previously added retroreflective infrared markers. This function takes advantage of the markers' location embedded in the texture to accomplish this objective. Once activated, it requests the user to click on the desired marker. As shown in figure 4.8, this creates a reference object (highlighted in orange) that will provide this information in the last step of processing to relocate the origin of the segmentation results automatically.

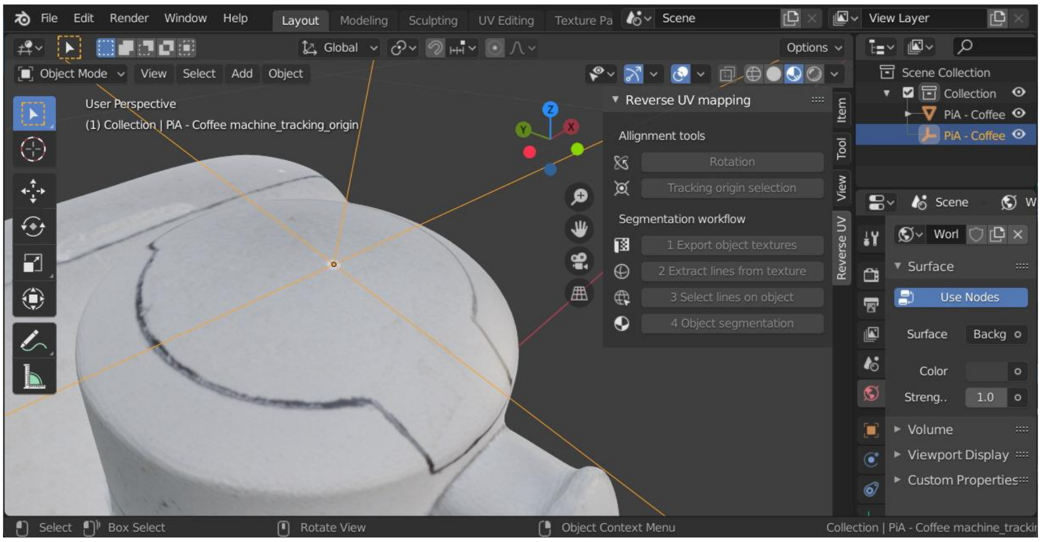
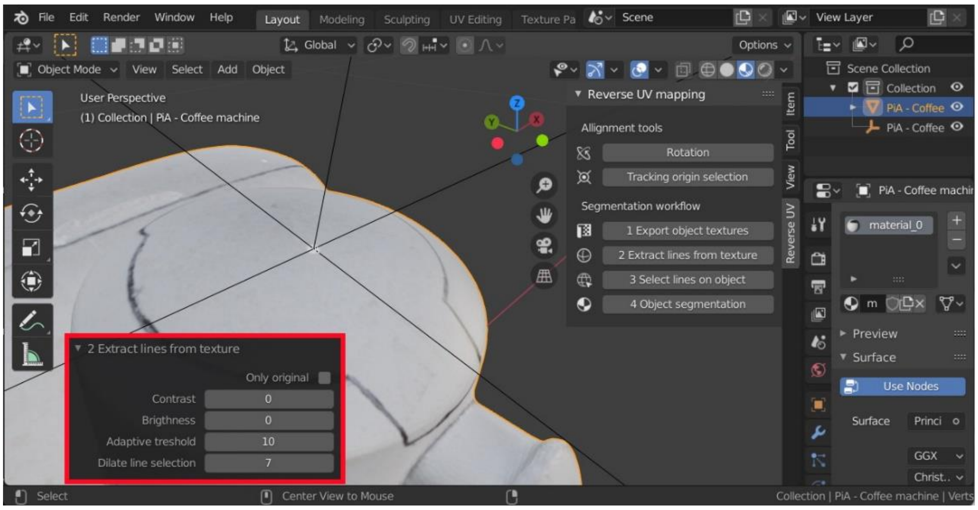


Figure 4.10 - Selection of tracking origin in Blender add-on

Then, starting with the segmentation workflow, the first operation exports the texture applied to the object to be segmented and generate an image of the UV map to use during processing. This step only requires the user to click and wait until it is done. Although this operation could have been executed automatically after the extraction of lines from the texture, it was decided to leave them as independent functions to avoid reloading the texture and regenerating the image of the UV map each time new parameters were used.

The second operation of the segmentation workflow extracts the lines (or high-contrast features) from the 3D model texture. As shown in figure 4.9, a new panel will appear once executed, allowing the user to change four parameters and one checkbox to change the results visualisation mode. The contrast and brightness apply some adjustments to the texture before extracting the features. As previously explained in section 4.1.4, the adaptive threshold and dilate line selection affect which features will be selected.



2 Extract lines from texture

Only original ☐

Contrast 0

Brigthness 0

Adaptive treshold 10

Dilate line selection 7

Figure 4.11 - Blender add-on with the extract lines from texture function highligthed in red

Once the function to extract lines from texture is activated, an additional window shows a low-resolution preview of the results superimposed in the original texture (see figure 4.10). Since the user does not know the ideal parameters beforehand, to ensure the selection of all the boundary features, the preview is automatically updated each time one of the parameters is changed. Moreover, since visualising which features are missing could be difficult due to the extra information, the users can disable the preview and check the original texture without added information.

Figure 4.12 - preview of feature detection results (left) and orginal texture (right) generated by Blender add-on

After the texture features have been extracted, the user can activate the function *to select lines on object*. Like the function that exports the textures, the user must only wait until the process is done without further output. It was considered an independent function to avoid recalculating all the selected faces each time the parameters were changed. As shown in figure 4.11, once the function has finished, using existing functions of Blender, the user can change the visualisation of the 3D model to see the selected faces and evaluate if the results are appropriate to execute the segmentation. In particular, the user must verify that the selected faces generate closed boundaries around the segments to be separated.

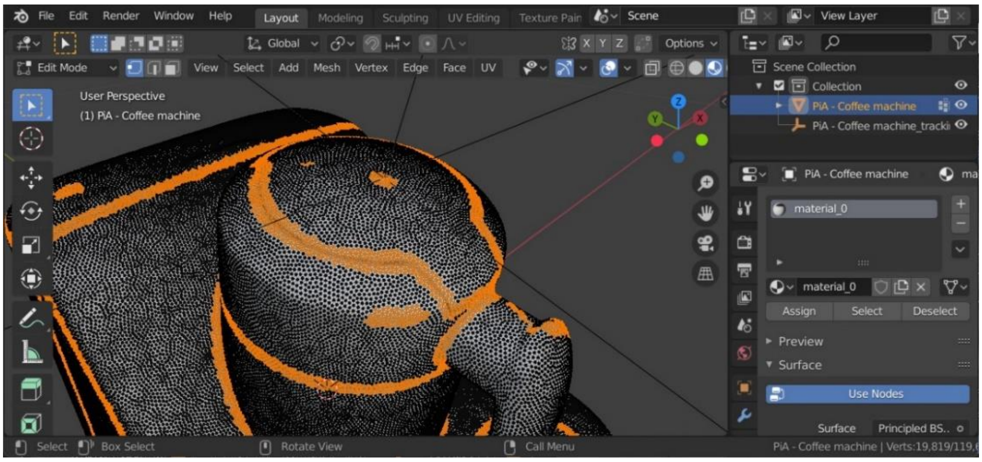
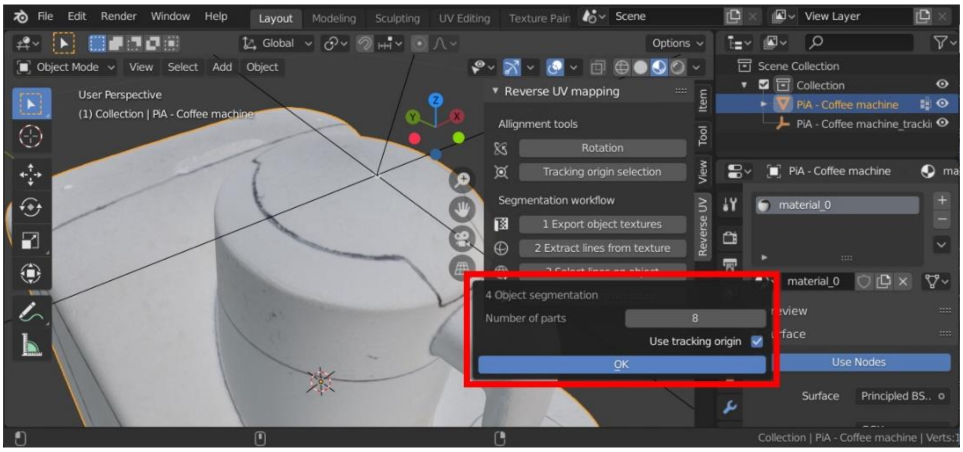


Figure 4.13 - Faces automatically selected by the Blender add-on

It is worth noting that smaller groups of selected faces that do not correspond to a boundary are not ideal. However, they do not affect the segmentation result as they will be assigned to the closest segments, which will be the ones that surround them.

If the selected faces comply with the requirements, the user can proceed with the last step. Otherwise, the features should be extracted again using different parameters, and the selection of the faces will need to be recalculated. As shown in Figure 4.12, after executing the object segmentation function, the user will be prompted with a panel asking the number of parts of the output segmentation. Moreover, as part of the SPARK project requirements for tracking, the user can confirm if he wants to relocate the 3D model origin by using the reference object previously created or leave it unchanged.



4 Object segmentation

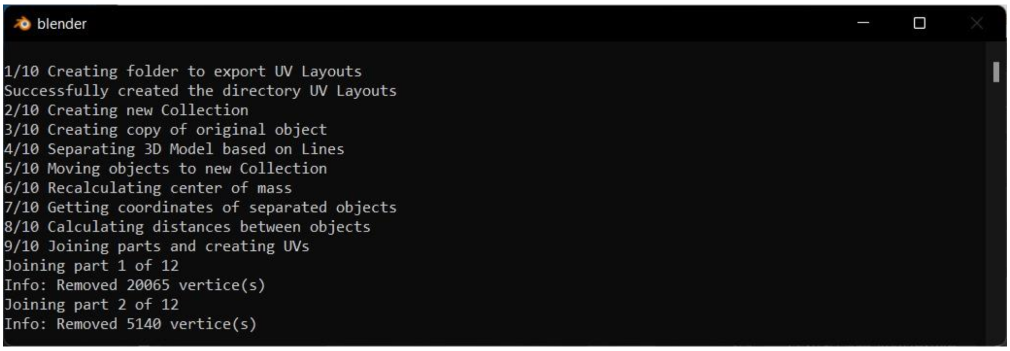
Number of parts 8

Use tracking origin ☑

OK

Figure 4.14 - Blender add-on with object segmentation options panel highligthed in red

As shown in figure 4.13, after choosing the parameters and starting the function, the add-on will execute a series of functions which will be prompted to the user in a new window. This function was added to keep the user informed about the process until it is finished, as it could take several minutes.



```
1/10 Creating folder to export UV Layouts
Successfully created the directory UV Layouts
2/10 Creating new Collection
3/10 Creating copy of original object
4/10 Separating 3D Model based on Lines
5/10 Moving objects to new Collection
6/10 Recalculating center of mass
7/10 Getting coordinates of separated objects
8/10 Calculating distances between objects
9/10 Joining parts and creating UVs
Joining part 1 of 12
Info: Removed 20065 vertice(s)
Joining part 2 of 12
Info: Removed 5140 vertice(s)
```

Figure 4.15 - Information window created during the object segmentation of the Blender add-on

After finishing the processing, the information window is automatically closed and the results are shown in the 3D viewport, where each segment has a random unique colour (See figure 4.14). Moreover, the add-on automatically generates a UV map for each new segment and exports them as an OBJ file for use on a mixed prototyping platform.

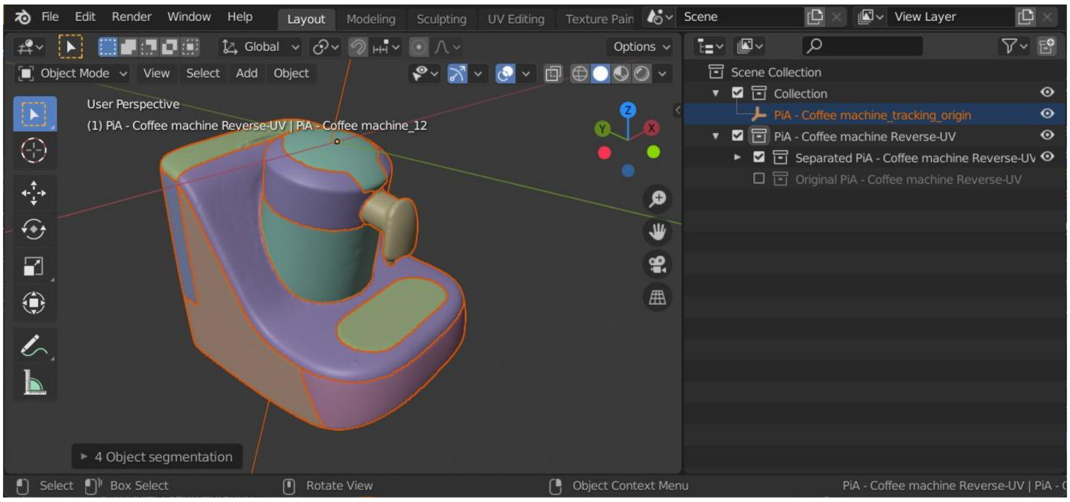


Figure 4.16 - Segmentation results of Blender add-on

As mentioned, all the intermediate files used for processing, the resulting segmentation, and log files can be found in the same directory path as the Blender project.

4.2.1.2 Benefits and limitations

The main benefits of this implementation are:

- Higher flexibility on the process for the user by having at his disposal all the already existing functions of a 3D modelling software.
- Reducing the need of switching software when operations not related to the proposed approach must be applied over the 3D model.

On the other hand, some of the limitations of this approach are:

- Overwhelming user interface due to the wide range of operations that can be done inside Blender.
- Slower processing times due to the use of internal functions of blender that must handle other data such as change history and scene organization, adding an “unnecessary” processing overhead to the add-on functions.

- Complex installation process due to the need of OpenCV, a library that is not included by default in the python environment used by Blender, and therefore require the use of the command prompt to add it.

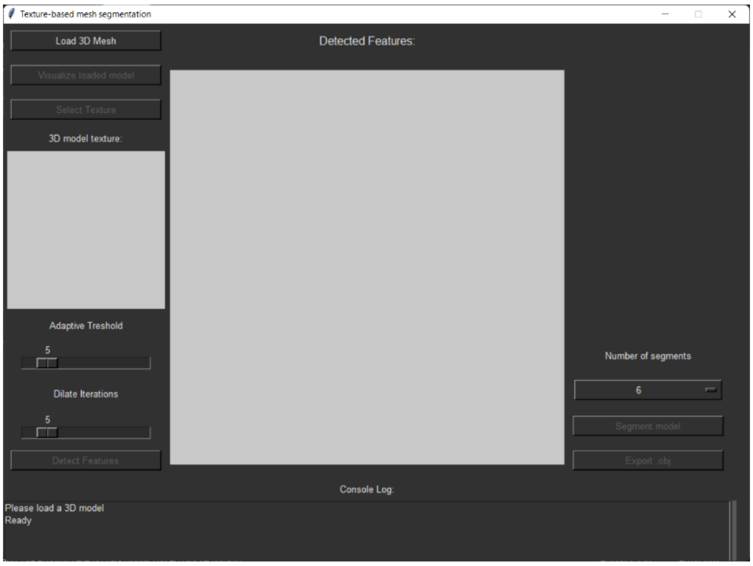
Some of the limitations could be due to a lack of competence in programming. Nevertheless, they were not critical for evaluating the proposed approach and could be solved in the future.

4.2.2 Standalone application

A standalone application was developed directly on Python using libigl (Jacobson et al., 2018), a geometry processing library, as well as functions of OpenCV (Bradski, 2000), and Tkinter to create the user interface. Regardless of these changes, the internal logic for processing remained almost identical to the Blender add-on. The primary motivation for creating this implementation was to overcome the inherent limitations of Blender regarding the processing performance, which was impacted by the need to work with the provided functions and data formats inside the application. Moreover, this was an opportunity to further simplify the process by giving the user only access to the specific functions needed for the texture-based segmentation, reducing the room for potential user errors.

4.2.2.1 Utilization

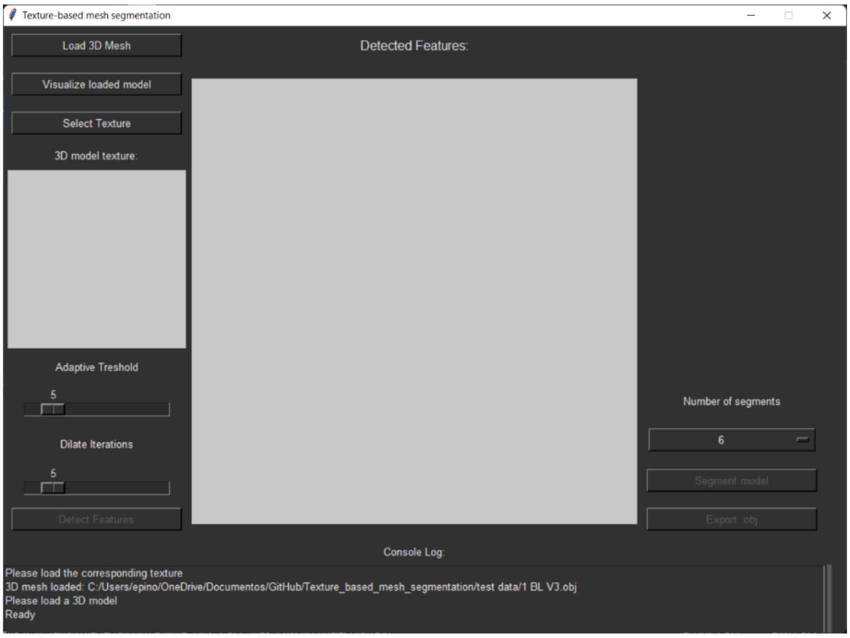
As shown in figure 4.15, the standalone application offers the same functions as the add-on but in a unique window. Initially, this window is shown with most of the functions disabled. This was done to make the user follow the expected order of the functions, making the other functions available only when the previous step was executed correctly. Moreover, a console log was added at the bottom of the window. This section allows the user to review a history of which functions have been executed and extra information, such as file paths and subprocesses executed during the segmentation.



```
Please load a 3D model
Ready
```

Figure 4.17 – User intertace of standalone application

Once the application is open, the first step is to load the 3D model to be segmented. This function is done through the "Load 3D Mesh" button, which opens a new window to search for the desired file upon activation. As shown in figure 4.16, after the 3D model has been loaded, the operation is recorded in the console log, and two new functions are enabled.



```
Please load the corresponding texture
3D mesh loaded: C:/Users/epino/OneDrive/Documentos/GitHub/Texture_based_mesh_segmentation/test data/1 BL V3.obj
Please load a 3D model
Ready
```

Figure 4.18 - User interface of standalone application after loading a 3D model

The “Visualize loaded model” opens a new browser window where the loaded 3D model is presented to the user (See figure 4.17). This window is interactive and allow the user to review if the model they are going to segment is the correct one.

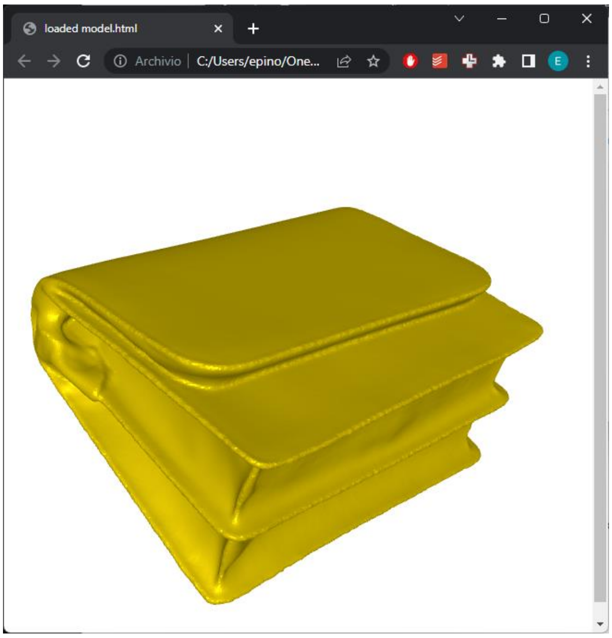
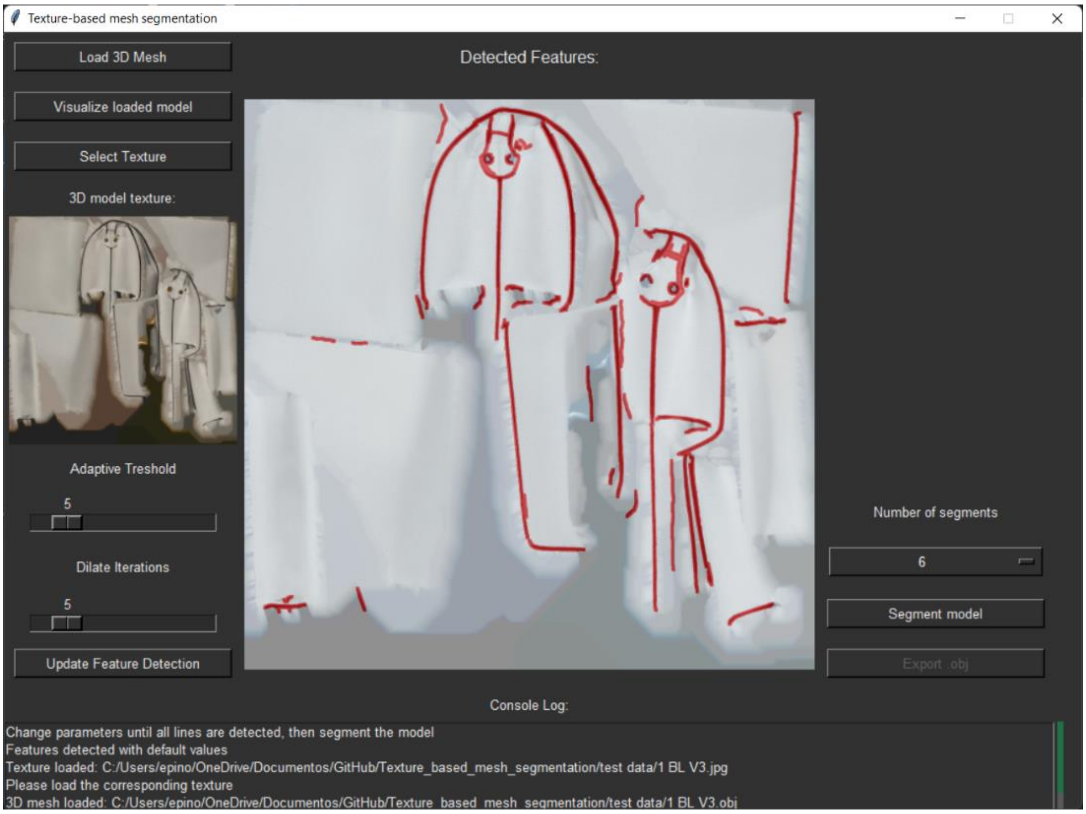


Figure 4.19 - Visualization of input 3D model with standalone application

Then, like the function to the load 3D model, when the user press “select texture”, a new window is open to search for the corresponding texture file.



Adaptive Treshold: 5

Dilate Iterations: 5

Number of segments: 6

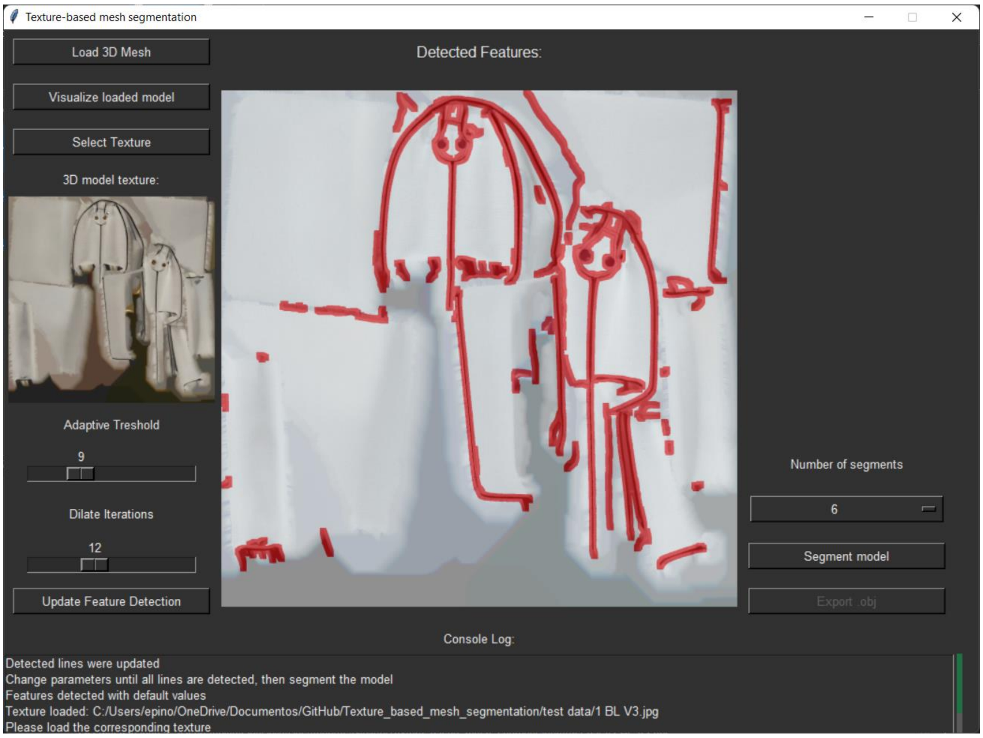
```
Change parameters until all lines are detected, then segment the model
Features detected with default values
Texture loaded: C:/Users/epino/OneDrive/Documentos/GitHub/Texture_based_mesh_segmentation/test data/1 BL V3.jpg
Please load the corresponding texture
3D mesh loaded: C:/Users/epino/OneDrive/Documentos/GitHub/Texture_based_mesh_segmentation/test data/1 BL V3.obj
```

Figure 4.20 - User interface of standalone application after loading texture

Once this is done, as shown in figure 4.18, the two grey squares initially placed on the user interface are replaced. At the left is a small image of the original texture for the user to keep as a reference when choosing the correct parameters. While at the right, a bigger image with the feature detection results is shown. In this second image, the original texture is converted to grayscale, and the detected features are highlighted in red to help the user recognise them. This image uses the default parameters and is meant to be a starting point for the user as he searches for the parameters that better select the texture features for segmentation. Moreover, as soon as the texture is loaded, the "segment model" function is enabled, as it has all the required information to do this process, even if the default feature detection parameters are not ideal.

To change the detected features, the user can move two sliders that change the input parameters for the adaptive threshold and the dilate iterations. After the user has changed the values of the sliders, it must press "Update Feature Detection" to visualise the new results and be able to use them during the segmentation process (see figure 4.19). The updating process takes less than a second and is intended to be repeated several times until the user is satisfied with the results.

]



Adaptive Treshold 9

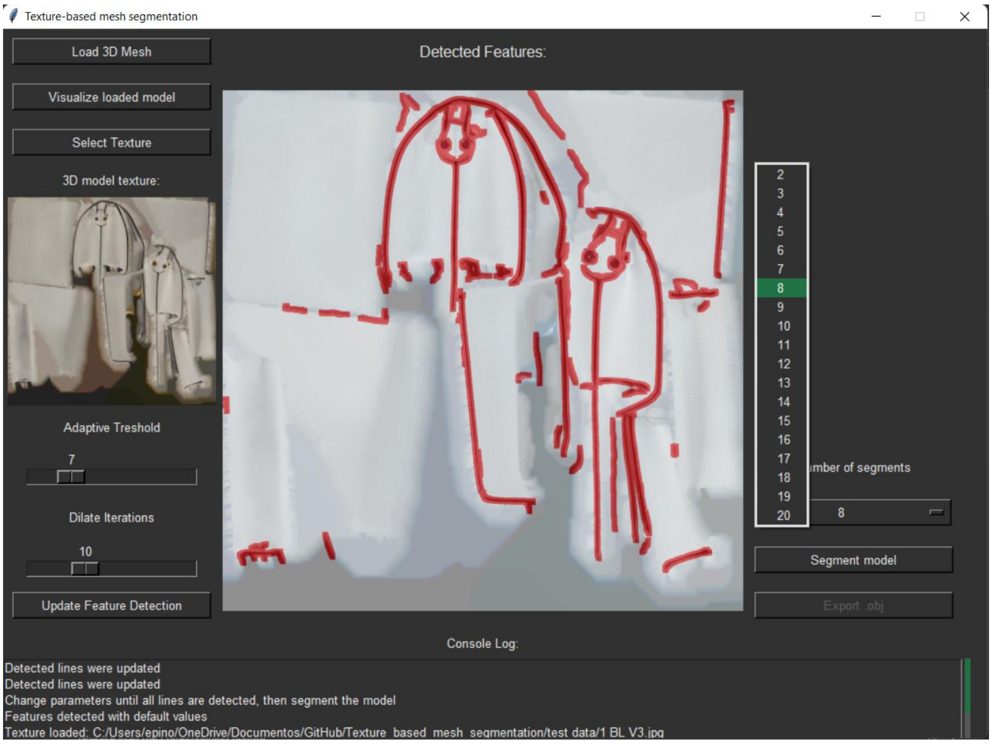
Dilate Iterations 12

Number of segments 6

```
Detected lines were updated
Change parameters until all lines are detected, then segment the model
Features detected with default values
Texture loaded: C:/Users/epino/OneDrive/Documentos/GitHub/Texture_based_mesh_segmentation/test data/1 BL V3.jpg
Please load the corresponding texture
```

Figure 4.21 - User interface of standalone application after updating feature detection parameters

Once the feature detection parameters have been chosen, the user has to select the number of segments that he wants in the output model. As shown in figure 4.20, this parameter is chosen using a button that presents values ranging from 2 to 20 segments upon activation. It is worth noting that the developed algorithm can output more segments. However, this could heavily impact processing times, so it was decided to add a limit.



Adaptive Treshold 7

Dilate Iterations 10

Number of segments 8

```
Detected lines were updated
Detected lines were updated
Change parameters until all lines are detected, then segment the model
Features detected with default values
Texture loaded: C:/Users/epino/OneDrive/Documentos/GitHub/Texture_based_mesh_segmentation/test data/1 BL V3.jpg
```

Figure 4.22 - User interface of standalone application when selecting the number of segments in the ouput model

After deciding the number of output segments, the user can segment the model. This process can take seconds depending on the 3D model's resolution, the texture's resolution, and the parameters used. Once the segmentation has finished, a new window is opened to visualise the results (See figure 4.21). The results visualisation is interactive and allows the user to review the obtained segmentation to decide whether to use that result or go back to try different parameters. Each segment is assigned a random colour to aid this process.

If the result of the segmentation is satisfactory, the user can proceed with the last operation of the application, exporting the 3D model to be used in other platforms. When this function is activated, the segmented model in OBJ format is generated in the same folder as the original model.

Additionally, this implementation generated a log file for evaluation purposes where all the user actions were recorded, including the parameters used. Moreover, aside from the exported model in OBJ, the segmentation was also exported as SEG files, a format specifically used to describe segmentations of a model and compare them against some reference segmentation.

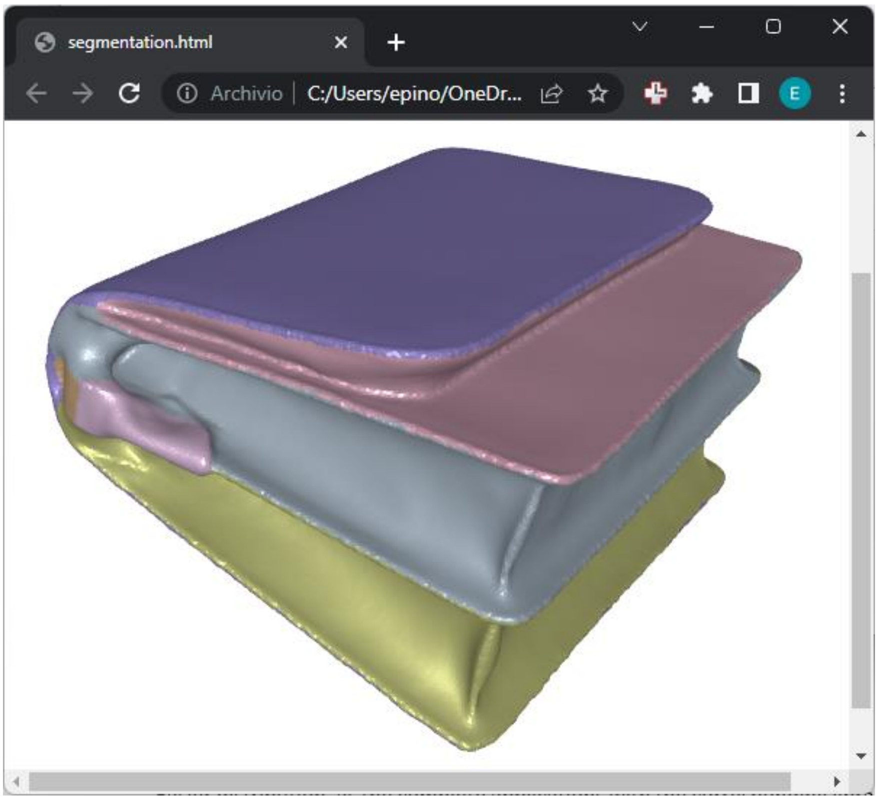


Figure 4.23 - Visualization of results from standalone application

4.2.2.2 Benefits and limitations

The main benefits of this implementation are:

- Easier distribution, as the complete applications with the corresponding libraries and dependencies can be compressed in a self-extracting installation file.
- Easier process of testing by generating specific 3D model formats needed for benchmarking.

- Improved processing times by working at a lower level (i.e., modifying and accessing vertex and polygon data directly, without the blender processing overhead)
- Simplified process by only offering the needed functions for the proposed approach.

On the other hand, some of the limitations of this approach are:

- The operations that the user can execute are only those related to the proposed approach, hence if other operations must be applied to the 3D model a second software would be needed.
- Not capable of showing the 3D models inside the user interface.
- This implementation was not capable of generating the UV maps of the segmentation when exporting it. For this reason, a simple add-on for Blender was developed. The add-on automatically generated UV maps for all the selected segments.

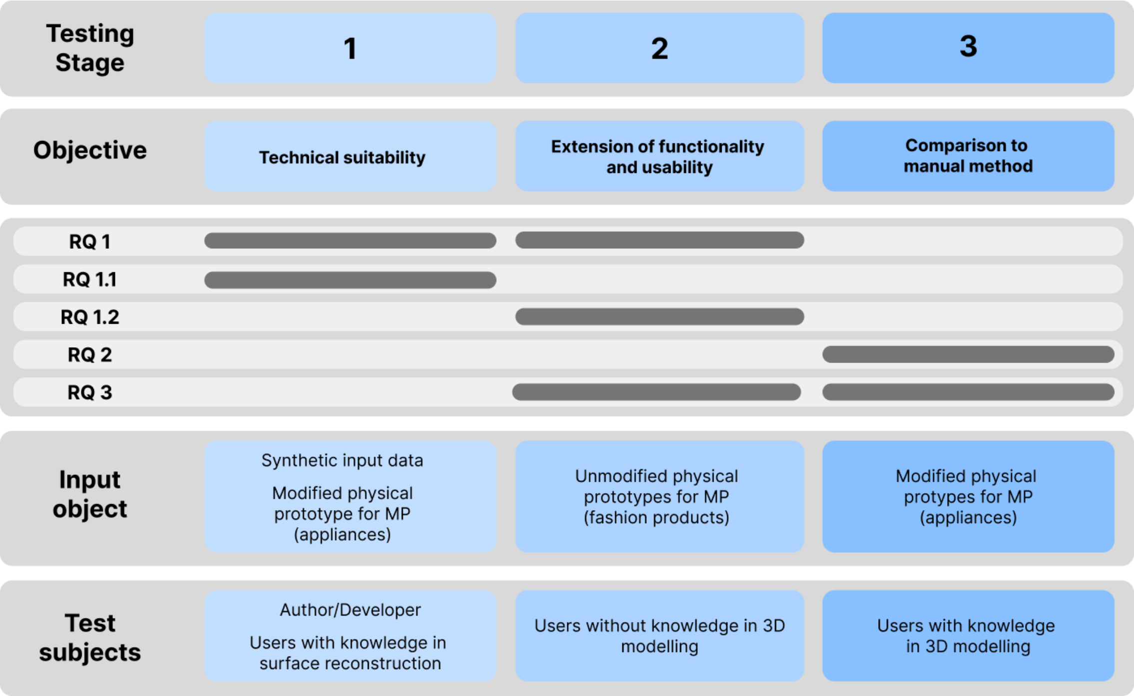
Like the add-on implementation, some of the limitations could be due to a lack of competence in programming. Nevertheless, they were not critical for evaluating the proposed approach and could be solved in the future.

Chapter 5: Validation plan

Considering the development of a novel texture-based mesh segmentation method to support the authoring of mixed prototypes, this chapter presents the validation plan to answer the previously defined research questions. In particular, an overview of the validation plan, the motivations, and the methods for the activities are presented. The execution and results of these activities will then be presented in chapters 6 to 8.

5.1 Overview

As shown in Figure 5.1, the validation plan comprises three testing stages, each with different objectives and generating supporting data to answer the research questions. Within these testing stages, the main variables were the parts of the content authoring workflow tested, the type of input objects used and the levels of technical knowledge of the test subjects.



Testing Stage	1	2	3
Objective	Technical suitability	Extension of functionality and usability	Comparison to manual method
RQ 1	▬	▬	
RQ 1.1	▬		
RQ 1.2		▬	
RQ 2			▬
RQ 3		▬	▬
Input object	Synthetic input data Modified physical prototype for MP (appliances)	Unmodified physical prototypes for MP (fashion products)	Modified physical protypes for MP (appliances)
Test subjects	Author/Developer Users with knowledge in surface reconstruction	Users without knowledge in 3D modelling	Users with knowledge in 3D modelling

Figure 5.1 - Validation plan overview

5.2 Validation activities

5.2.1 Technical suitability

This activity addresses the following research questions:

RQ1: Can a computer vision supported process generate a segmented 3D model suitable for mixed prototyping?

RQ1.1: using added graphical mark on the surface of the physical prototype

This was the first research activity of the developed method's validation plan. As such, the included tests focus on verifying if it is feasible in a simplified scenario. To achieve this, a testing process was executed iteratively, adding new variables, and increasing the complexity of the input at each step. Moreover, this process was carried out in parallel to the initial implementation of the proposed approach using the results of those tests as direct feedback for its improvement.

5.2.1.1 Test 1: Synthetic input data

First, the core of the proposed mesh segmentation approach was tested, that is, the feature detection and segmentation algorithms. The main objectives of this test were:

1. Verify if the proposed approach can segment a 3D model using its texture.
2. Implement and debug the feature detection and segmentation algorithms.
3. Evaluate the correlation between the input parameters and resulting segmentation.

To accomplish this, the test used synthetic input data instead of 3D models generated through surface reconstruction. This input data considered simple geometries and digitally created texture, reducing the variables and complexity of the input model when compared to a reconstructed model. Hence, easing the detection and fixing of potential errors during the implementation.

Due to the early stage of development in which this test was executed, no user interaction was considered.

5.2.1.2 Test 2: Real physical prototype

After the proposed approach was tested using synthetic input data and the implementation was refined, the next step was to test the complete workflow in a more realistic scenario. The main objectives of this test were:

1. Verify if the positive results obtained with synthetic data could be replicated using a texture input generated through surface reconstruction methods.
2. Evaluate the robustness of the tool to handle complex geometries and correctly segment it in the multiple parts of the prototype.
3. Verify the suitability of the approach to support two common tasks of the product development process: material selection and design of user interfaces.
4. Debug functionalities of the implementation related to the specific mixed prototyping platform to be used (i.e., SPARK).

To achieve this, an actual prototype was used. The proposed approach was executed from preparing the physical prototype to using the results within a mixed prototyping platform.

Due to the intention to simulate a realistic scenario, a prototype with an adequate level of complexity was selected, considering the number of segments and geometrical shape. On one side, a prototype with a low number of segments would fail to stress the approach's capabilities, limiting the test's ability to help find early problems in the results. While on the other hand, a prototype with a high number of segments would difficult the analysis of the results to enhance the implementation. As a result, a prototype ranging from 5 to 15 segments was considered to fit this activity. Moreover, regarding the geometrical shape of the prototype and its segments, the selected prototype was expected to provide a representative result that could be extrapolated to a wide range of physical prototypes. Hence, it included segments of different sizes and geometrical features (e.g., planes, curves, and hard edges).

The evaluation of the complete workflow provided valuable data to understand the requirements and impact of each stage and the transitions between them. Among these, particular interest was the effect of the characteristics of the reconstructed surface model in the resulting segmentation. Such characteristics include mesh resolution, remeshing operations, texture resolution, and texture post-processing, among others. Moreover, by considering the use of the resulting model on a mixed prototyping platform, this test also helped to define additional requirements in the developed tool to support the authoring process.

Like the previous test, due to the early stage of development in which this test was executed, no user interaction was considered, and the author executed the complete activity with the feedback of the research team.

5.2.1.3 Test 3: Multiple objects with users

The previous test helped verify the technical suitability of the proposed approach in a more realistic setup. However, it considered only one test object, and the author executed the process, which certainly limited the findings obtained from it. Considering this, an additional test was planned with the following objectives:

1. Evaluate the behaviour of the mesh segmentation algorithm on a wider variety of object to understand better the effect of variables such as the geometry of the object and the preparation process of the physical prototype.
2. Evaluate if mesh segmentation approach could output geometrical features as a single segment when there are no surface features indicating that they are separate entities and at the same time, segment as separate elements 2 or more portions of a single geometrical feature when there are surface features defining their boundaries.
3. Detect usability problems related with the software implementation.

To achieve these objectives, new objects were used. Moreover, the workflow was executed by subjects not related to the development, from preparing the physical prototype to segmenting the 3D models.

Regarding the new object, due to the lack of existing prototypes available to provide to the subjects, several small appliances and products were provided instead. These objects had similar requirements in terms of complexity as in the previous test. However, more emphasis was placed on having products from different categories and geometries rather than the number of segments.

Regarding the selection of the subjects, the main requirement in terms of knowledge background was to have experience in surface reconstruction methods. This base allowed them to execute the proposed workflow and generate the input 3D models for the segmentation algorithm. Moreover, since the quality of models generated through surface reconstruction methods depends in part on the skill of the users, this test was also an opportunity to avoid a bias towards models reconstructed only by the author. Nevertheless, some output requirements were still provided to the subjects based on the results of the previous tests, namely, the resolution and format of the mesh and texture. Furthermore, due to the participation of users external to the development, this activity was also an opportunity to refine the testing methodology for following evaluations focused on user interaction.

Finally, the workflow evaluation was focused on the preparation of the physical prototype until the segmentation of the 3D models. The results evaluation on a mixed prototyping platform was excluded during this testing activity because a qualitative analysis of the results was already done with the previous test. Moreover, further evaluation of the results could be done by analysing the segmentation results without importing them on the mixed prototyping platform.

5.2.2 Extension of functionality and usability

This activity addressed the following research question:

RQ1: Can a computer vision supported process generate a segmented 3D model suitable for mixed prototyping?

RQ1.2: using existing features on the surface of the physical prototype.

RQ3: Does a content authoring process supported by computer vision improve user acceptance of mixed prototyping preparation?

This activity focused on evaluating the suitability of the proposed method, although at a more advanced level of development, and considering the use of existing features in the physical prototype rather than added features. Moreover, it also focused on the solution’s impact on user acceptance, a variable that was not addressed in the previous test. More specifically, this test had the following objectives:

1. Evaluate the capacity of the proposed approach to segment textured 3D models using existing features.
2. Evaluate the capacity of subjects with low knowledge in 3D modelling to use the proposed approach and reach a satisfactory segmentation.
3. Evaluate the behaviour of the subjects while using the proposed approach.
4. Evaluate the usability of the developed solution.

To achieve this, a group of subjects with low competence in 3D modelling executed the segmentation process using a provided 3D model of an unmodified physical prototype. During this process, data from the different stages was gathered. At the end of the activity, a survey was carried out to obtain additional information from the participants.

For the selection of the physical prototype, some requirements were considered to allow automatic segmentation without modifying it. Specifically, it must have a neutral base colour such as white or grey and high contrast boundary features that generate closed segments to be edited individually in the mixed prototyping platform (See section 4.1.1 for more details).

Regarding the selection of the subjects, the main requirement was that they had low competence in 3D modelling to simulate the worst-case scenario when considering the future users of mixed prototyping. Moreover, since there is an interest in understanding the potential impact on technology adoption, those subjects also had technical knowledge in the current prototyping process. This allowed them to have a reference point for comparison and increased engagement during the activity.

Due to the broad scope of this testing activity, several evaluation metrics were considered. Therefore, a brief description is presented below, with a more extensive explanation in section 7.4.

To analyse the capacity of the algorithm to detect the boundaries of the segments in the input texture and the subject's capacity to select the ideal parameters, the features extracted by the subjects were compared against a ground truth feature detection. This comparison was done using a confusion matrix and calculating the precision and recall, two metrics commonly used to evaluate categorisation problems in computer vision and machine learning.

To analyse the capacity of the algorithm to segment the textured 3D model using existing features and the capacity of the subjects to select the ideal parameters, the features extracted by the subjects were compared against a ground truth segmentation. This comparison was made using the segmentation benchmark tool developed by Chen et al. (2009), which evaluates the results of automatic segmentation algorithms against manual segmentations. The metrics included in this benchmark are: Cut discrepancy, Hamming distance, Rand index and Consistency error.

To analyse the impact of the proposed approach in the adoption of the technology, the TAM (Technology Acceptance Model) survey was used. This survey comprises two sections, one focused on the PU (Perceived Usefulness) and the other on the PEoU (Perceived Ease of Use).

Finally, the developed segmentation tool integrated a logging functionality to analyse user behaviour. This allowed to gather the time and order in which each function was executed, which was then used to detect patterns in the subject's results.

5.2.3 Comparison to manual method

This activity addressed the following research questions:

RQ2: How do the results of a texture-based mesh segmentation compare in terms of quality to a manual segmentation?

RQ3: Does a content authoring process supported by computer vision improve user acceptance of mixed prototyping?

The previous test integrated a comparison between the current method for mesh segmentation used in MP (i.e., using the standard tools of 3D modelling software) and the automated mesh segmentation approach proposed in this thesis. However, this comparison was made through a unique ground truth segmentation created by the author. Therefore, although it was helpful to confirm if the proposed approach could properly segment the models, it did not address both methods from the user's perspective. Moreover, multiple persons could generate different manual segmentations. Therefore, this variation should also be considered when evaluating the quality of the segmentations obtained by the proposed approach. Hence, this new testing activity compared the manual and automated segmentation with both processes conducted by the users. Furthermore, although the previous test addressed the solution's impact on the user acceptance of the technology, it considered only the proposed solution. Therefore, it missed a reference point to evaluate changes in the process. Considering this, the testing activity had the following objectives:

1. Evaluate the similarity between a model segmented using the proposed approach and the standard tools of 3D modelling software.
2. Evaluate the perceived ease of use of a segmentation process done using the proposed approach and using standard tools of 3D modelling software.
3. Evaluate the correlation between quality of the segmentation and the time required to execute this process.

To achieve these objectives, a group of subjects with various levels of knowledge in 3D modelling were asked to segment textured models generated through surface reconstruction using the proposed approach and the standard tools in 3D modelling software. In addition, data on the users' actions and the segmentation results were gathered during this process. Finally, a survey was conducted to evaluate the perceived ease of use of both methods.

Regarding the selection of the subjects for this test, the main requirement was that they had competencies in 3D modelling, allowing a more realistic comparison of both methods. Moreover, having subjects with a wide range of experience allowed the evaluation of the method's usefulness in correlation to their experience. Finally, it is worth noting that since mesh segmentation is a specific competence inside 3D modelling and is not a given that experienced users will know how to do it, a tutorial was provided to all the subjects.

Chapter 6: Technical suitability

This chapter presents the first activity of the validation plan, which evaluates the technical suitability of a texture-based mesh segmentation approach within the mixed prototyping content authoring workflow. First, part of the proposed approach was evaluated with synthetic input data to check the core functionality of the algorithm and better understand its capacity to segment a model using texture features. Then, the complete workflow was evaluated with a real physical prototype to evaluate the tool's robustness to handle complex geometries and input textures generated through surface reconstruction methods. Finally, a test, including users and multiple objects, was conducted to evaluate its behaviour in various geometries and gather early usability feedback.

6.1 Test 1: Synthetic input data

6.1.1 Context

This test was executed within the first stages of implementation of the proposed approach. Therefore, the code was changed several times during its execution. Nevertheless, the core algorithm remained virtually untouched. Most of the changes were related to the integration through the Blender python API, accessing the available data, the user interface, and the performance of the algorithms.

6.1.2 Input selection and preparation

At this initial test, the focus was the core of the proposed approach, which is the functionality of the algorithm and its capacity to segment a model using texture features. To evaluate this with the least number of variables, it was decided to use a simple geometry with a digitally created texture, skipping the preparation of a physical prototype and surface reconstruction steps.

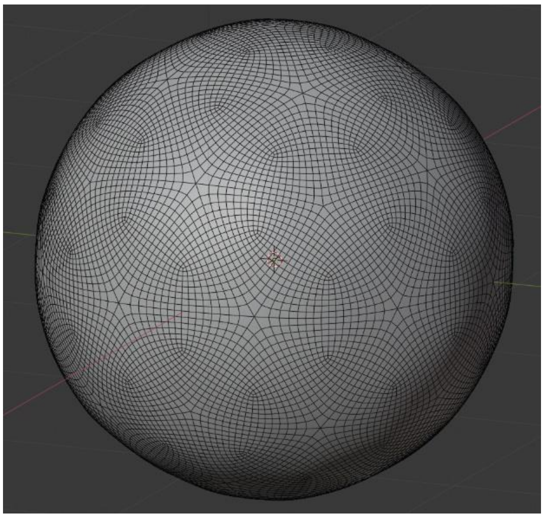


Figure 6.1 – Wireframe view of subdivided icosphere

The selected geometry was an icosphere, mainly because this primitive has a homogeneous distribution of polygons. Hence, having a higher probability that a vertex will be inside a texture feature, independently of where it is placed on the surface. Moreover, the curved shape of the mesh and the possibility to subdivide it allowed replicating to some degree the characteristics of a surface reconstructed model (See figure 6.1).

Regarding the texture, it was digitally created within Blender in 3 steps, resulting in the textured model shown in figure 6.2. First, a white base colour was applied to simulate the surface of a mixed prototype. Then, black lines were added to define the boundaries of the segments to be separated by the algorithm. Finally, several light sources were added to the 3D scene, and its effect was saved to the texture through a process known as baking. This last step simulated the effect of shadows and reflections, typically captured by surface reconstruction methods such as photogrammetry. The added lines clearly defined eight symmetric segments. However, they did not necessarily follow the natural path of the edges in the mesh.

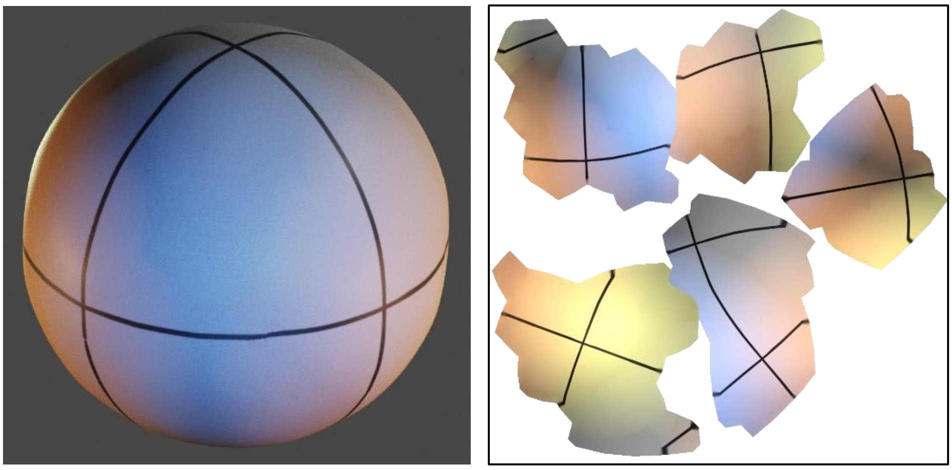


Figure 6.2 - Textured icosphere (left) and its correspoding 2D texture (right)

Moreover, although most of the tests were carried out with a mesh of 15,360 faces and a texture of 1024 pixels by 1024 pixels, several variations on the mesh's resolution and the texture's resolution were created to better understand its impact on the performance of the algorithm.

6.1.3 Execution of the proposed approach

The execution of the first implementation of the proposed approach was carried out by the author and did not include a physical counterpart. Additionally, it considered the preparation for the projection system; the extraction of texture features; the segmentation of the 3D model and UV mapping; and the exporting to the mixed prototyping platform.

6.1.4 Results

Although several insights were taken from this test, only the results related to the extraction of texture features and the segmentation of the 3D model and UV mapping are reported. This is mainly because the other steps were too simple, did not have much value from a research standpoint, or were closely related to the SPARK platform, hence being not generalizable to other situations.

Regarding the extraction of texture features, several input values were used for the adaptive threshold and dilate iteration parameters, being, in most cases, able to detect the black lines correctly. Nevertheless, as seen in the closeup shown in figure 6.3, when the input parameter that controls the adaptive threshold was too low, some parts of the line with lower contrast against the background were not detected. Hence, the boundaries did not form a closed boundary for that segment. Moreover, it is worth noting that while the dilate operation could close small gaps during the feature detection, more significant gaps remained open, even after a higher number of iterations.

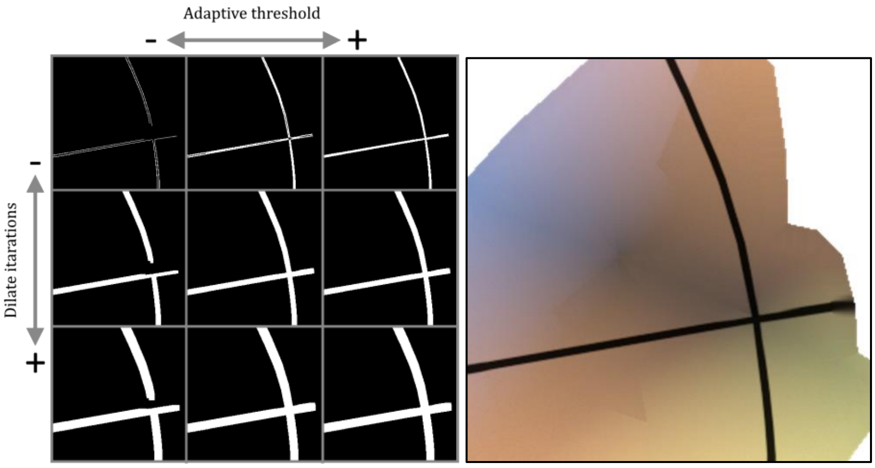


Figure 6.3 - Closeup of feature extraction using different parameters

Understanding this, by selecting the appropriate parameters, the proposed approach could correctly extract the black lines from the texture without being affected by changes in the colour or luminance around them and avoiding any discontinuities in the boundaries (see figure 6.4).

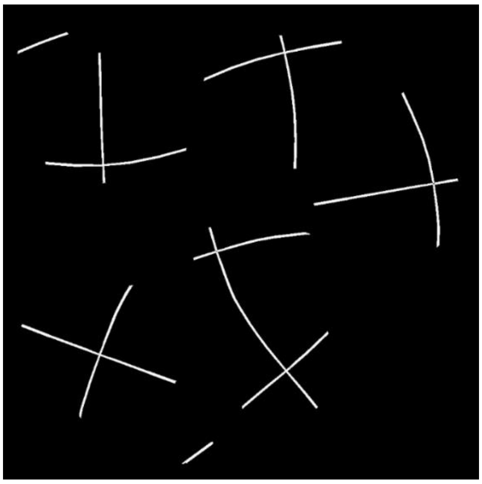


Figure 6.4 - Extracted features from icosphere texture

Regarding the segmentation of the 3D mesh, as shown in figure 6.5, the algorithm used the extracted features as an input to select the corresponding vertices in the UV map and then transfer that selection to the mesh. It is worth noting that, like the extraction of texture features step, the dilate iteration parameter significantly influences the continuity of the selection in the mesh. Although potentially more accurate, a narrow selection could leave vertices out of the selection and prevent the formation of closed boundaries.

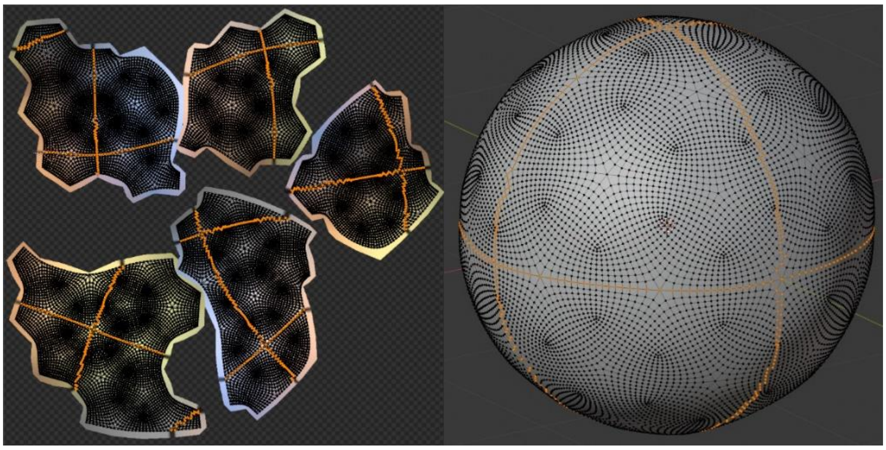


Figure 6.5 - selected features on UV map (left) and mesh (right)

As shown in figure 6.6, with an input of 8 expected segments, the final output was properly segmented following the black lines defined by the texture, although with slight deviations in some parts of the boundaries. Nevertheless, this was an expected behaviour, as the algorithm does not change the original mesh but finds a path within the existing possibilities.

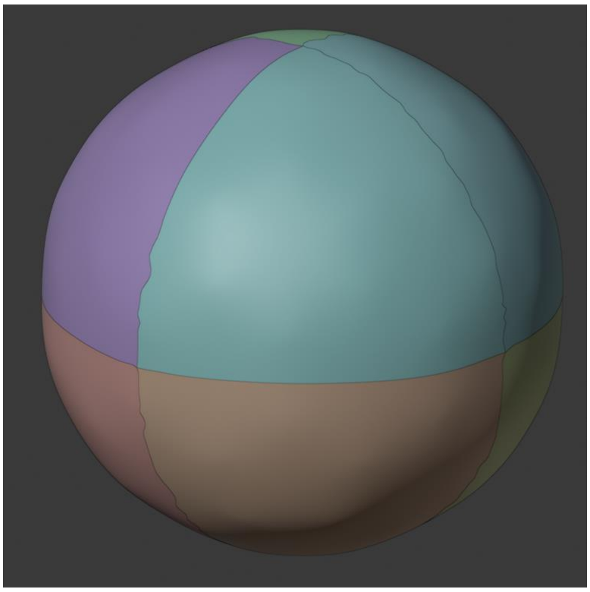


Figure 6.6 - Segmented icosphere using proposed approach

Finally, regarding the algorithm's performance, it was found that the most processing-intensive step of the approach was assigning each of the polygons within the boundaries to the corresponding segments. Moreover, it was also noted that the parameters that most influenced the required processing time were the resolution of the mesh, the number of polygons selected using the extracted features as an input, and the number of output segments. This behaviour was due to their direct impact on the number of calculations needed to define to which segment each of the selected polygons belongs.

6.2 Test 2: Real physical prototype

6.2.1 Object selection and preparation

This test was conducted with the physical prototype of the coffee machine shown in figure 6.7. This prototype was selected due to the following reasons:

1. It had a complex geometry with several planar, concave, and convex sections.
2. Although it was composed of one solid piece, the surface had clearly defined segments separated by crevices around them.
3. Some segments were placed inside other segments, a case that was not tested before.
4. The surface was white and porous, and the size was approximately 22 cm X 29 cm X 28 cm, being ideal for surface reconstruction methods as well as projection of digital elements over it.

Figure 6.7 - Coffe machine prototype before preparation

The prototype was previously used for display-based augmented reality, and because of this reason, it had several markers over its surface. To prepare it, all the markers and other added labels were removed, leaving a surface of a light neutral colour. Then, two additional changes were applied. First, using a marker, black lines were manually added following the existing crevices on the prototype's surface to indicate the boundaries of the different segments to separate. Furthermore, for the specific tracking requirements of the SPARK platform, a series of infrared retro-reflective markers were added to the surface (See figure 6.8).

Figure 6.8 – Prepared coffee machine physical prototype (left) with close-up of added features (right)

It is worth noting that although the added lines were placed following the geometrical features of the prototype, the proposed approach at no point uses this data, as it relies only on the texture information.

6.2.2 Execution of proposed approach

Figure 6.9 shows the results of the surface reconstruction, where it can be seen that the added features have been appropriately transferred to the texture of the 3D model. The 3D model was composed of 149,959 vertices and 299,914 polygons.

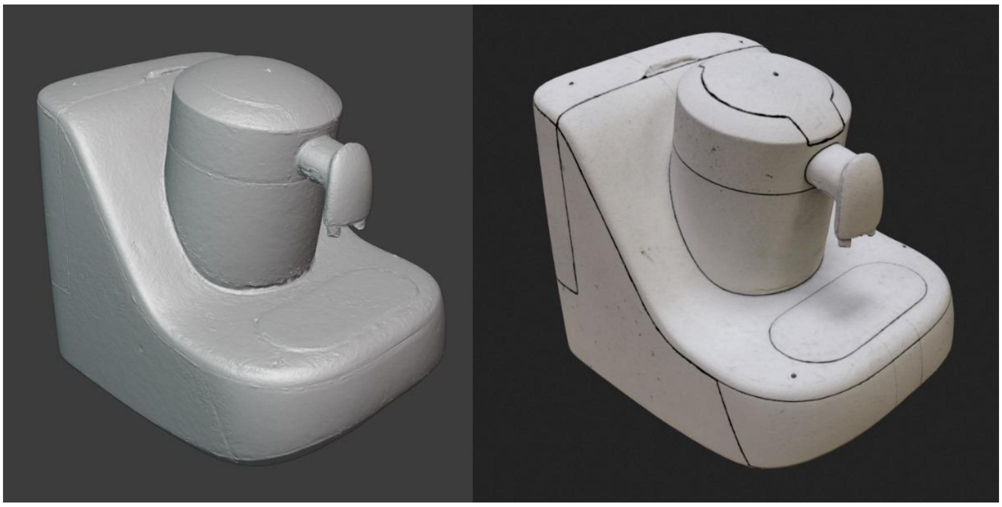


Figure 6.9 – Coffee machine prototype reconstructed mesh generated through photogrammetry (left) and textured 3D model (right)

Once the textured 3D model was imported into Blender, its rotation and scale were adjusted, and the origin of the 3D model was relocated using the developed algorithm. Figure 6.10 shows the results of the first three steps of the preparation of the 3D model for real-time editing: the extraction of the original texture (a) and initial UV map (b) generated by the surface reconstruction software, the resulting extracted lines from the original texture (c) and the selection of vertices in the 3D model based on the extracted lines (d).

Figure 6.10 - Original texture (a), initial UV map (b), extracted lines (c) and selected vertices (d)

Finally, the process of segmentation was executed with an output parameter of 12 segments.

6.2.3 Results

Figure 6.11 shows the results of the segmentation of the 3D model and UV mapping of the individual parts. As expected, the segmentation process generated an output mesh of 12 segments. During processing, a total of 33,142 polygons were categorized as secondary meshes. This step's total processing time was 24 minutes and 46 seconds using a computer with an Intel Core i7-9750H processor.

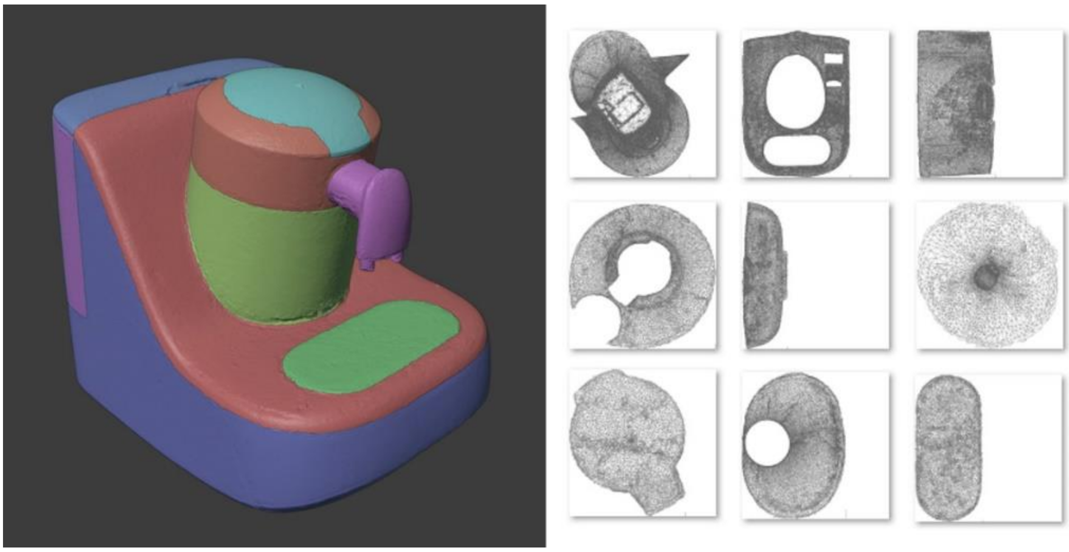


Figure 6.11 – Segmented model (left) with new UV maps (right)

The algorithm was able to separate a 3D model with a complex geometry on the relevant parts of the prototype and generate the corresponding UV maps for each segment with low distortion. Because of this, the model was suitable for real-time interactions and loaded in the SPARK platform. Furthermore, figure 6.12 shows several design variants created within the platform, where it was possible to modify the materials of each segment independently and add user interface elements such as buttons and displays without noticeable distortion. Hence, this test demonstrated the technical suitability of a computer-vision-based approach for mesh segmentation in mixed prototyping.

Figure 6.12 - Coffee machine design variants tested in SPARK platform

6.3 Test 3: Multiple objects with users

6.3.1 Context

This testing activity was done in collaboration with the course "Computer vision and Reverse engineering", supervised by professors Gabriele Guidi and Laura Micoli at the mechanical engineering department of Politecnico di Milano. Initially, this course had a final project where the student selected an object in conjunction with the professors to generate a textured 3D model using surface reconstruction methods. Since the proposed approach requires textured 3D models, this was an excellent opportunity to expand the number of tested objects and have initial feedback from new users. Moreover, it is worth noting that due to the global pandemic that affected the normal execution of courses at the time of this test, some students attended the course remotely.

6.3.2 Object selection and preparation

Two options were proposed depending on the student's location to align the course plan with the testing activities.

For the students present in Milano during the course development, a set of objects was prepared for them. Since the main research area corresponds to the product development process, the object category used was home appliances. To simulate this kind of prototype, six home appliances and objects were bought and prepared (See figure 6.13). Particular attention was put on selecting objects with a wide variety of geometrical features (e.g., curves, planes, symmetries, asymmetries, holes, concavities, and convexities) and segmentation requirements (e.g., output various geometric features as a single segment, and separate single geometric features in multiple segments).

Two layers of spray paint were applied to create a base suitable for surface reconstruction and accurate reproduction of digital elements. First, a layer of white mate primer to cover the original colour of the objects and then a second using a white and grey granite effect spray to add non-repetitive patterns to ease the surface reconstruction process. Finally, black lines were added using a marker to indicate the boundaries of the different segments to separate.

Figure 6.13 - CV&RE course: objects prepared by the author

The students that followed the course remotely were allowed to choose any object, as in the original course plan, but had to prepare them for the texture-based segmentation. Following the painting steps explained before, a tutorial to prepare the objects was provided to the students. However, economic and local availability limitations prevented the students from reaching the same results in this case. Considering this, while five students agreed to participate with objects prepared by themselves, only one was considered a suitable input for the proposed approach. Figure 6.14 shows a picture of the prepared object extracted from the student's datasets for surface reconstruction.

Figure 6.14 - CV&RE course: object prepared by a student

It is worth noting that while the professors promoted this activity, ultimately, it was an optional path for their final project. They could choose any object they wanted without the need to participate in testing the proposed approach.

6.3.3 Execution of the proposed approach

Considering that the students already had the physical prototypes available, the next step of the proposed workflow was to obtain a textured 3D model through surface reconstruction methods. As part of the regular plan of the course, they did it using photogrammetry and the cameras they had at their disposal, i.e., from professional to smartphone cameras. The main requirements for the resulting 3D model were:

1. a total polycount between 200,000 and 500,000.
2. the inclusion of the corresponding texture with a resolution of 2048 by 2048 pixels.
3. the use of OBJ format for compatibility.

In parallel and integrated within the course lectures, it was explained to the students how to use the developed tool utilizing the coffee machine 3D model as an example. Once the students learned how to use the segmentation tool, they used it in their textured 3D models as an assignment and sent the results via mail. Additionally, for those students attending the course in person, an extra interview was done to gather comments about the tool or the process itself.

6.3.4 Results

All provided objects were surface reconstructed using photogrammetry using input images acquired from various devices ranging from smartphones to professional cameras. Later, each of the models was segmented by the subjects using the texture-based mesh segmentation.

Figure 6.15 - Input textures (top), detected features (middle), and segmentation results (bottom)

As can be seen in figure 6.15, this activity provided input images with different levels of fragmentation, lightning conditions, and visible features that did not correspond to boundary features. Nevertheless, the algorithm could detect and extract the boundary features, transfer them to the 3D mesh and generate a part-based segmentation. Moreover, as shown in figure 6.16, in some exemplary segments of the processed models, the texture-based was able to output various geometrical features as a single segment when there were no surface features indicating that they were separate entities, and at the same time, separate in multiple segments single geometrical features when surface features defined their boundaries.

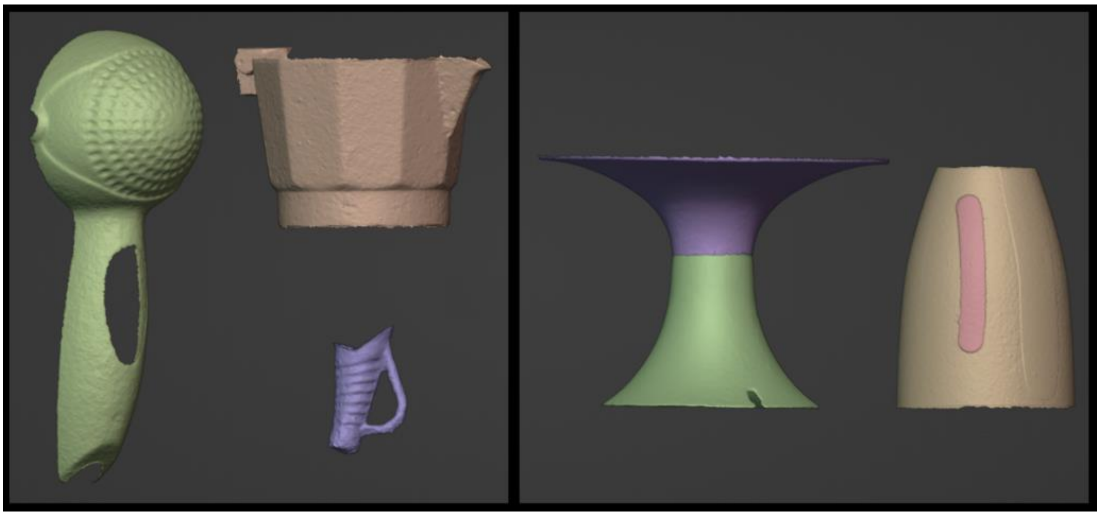


Figure 6.16 - Exemplary segments of processed models. Single segments composed of diverse geometrical features (left), and single geometrical features segmented in multiple parts (right).

A particular case was the student that worked with a mixer, which due to its shape, could not gather photos from a side of the object with added lines. Therefore, the resulting 3D model missed that texture information for automatic segmentation (See figure 6.17). This limitation was seen as an opportunity to test an alternative process to add texture features for segmentation: digitally adding the lines in the texture of the 3D model. This option yielded similar results for segmentation but required extra knowledge of 3D texturing, contradicting the main objective of the proposed approach of reducing competence requirements. Hence it was not further developed.

Figure 6.17 - CV&RE: Mixer with missing texture (left) and with digitally added lines (right)

Under closer inspection of the segmented models, it was noticed that small features with several close boundaries such as the handle in the coffee pot had problems to generate a clear segmentation of the parts (see figure 6.18). This is explained due to the higher uncertainty on to which segment assign each of the polygons classified as boundaries.

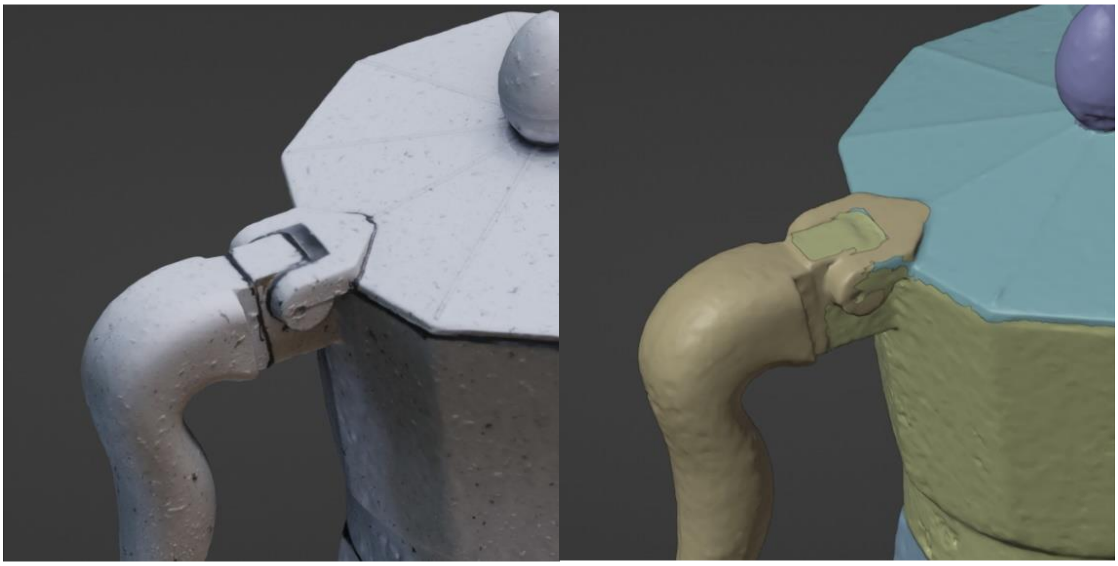


Figure 6.18 - closeup to coffee pot model with original texture (left) and segmented (right)

Conversely, the student that applied the proposed approach to the object prepared himself did not manage to segment its 3D model, as the algorithm could not detect the corresponding segments with the input parameter he used. This behaviour can be attributed partly to the low fidelity of the surface reconstruction and the irregularity in the definition of the added features in the physical object, most of which did not generate closed boundaries (see figure 6.19).

Figure 6.19 - CV&RE: 3D model (left) and texture (right) of object prepared by student

To verify these results, the author tested another set of parameters much higher than those used on the other objects, and the result partially followed the intended segmentation (see figure 6.20).

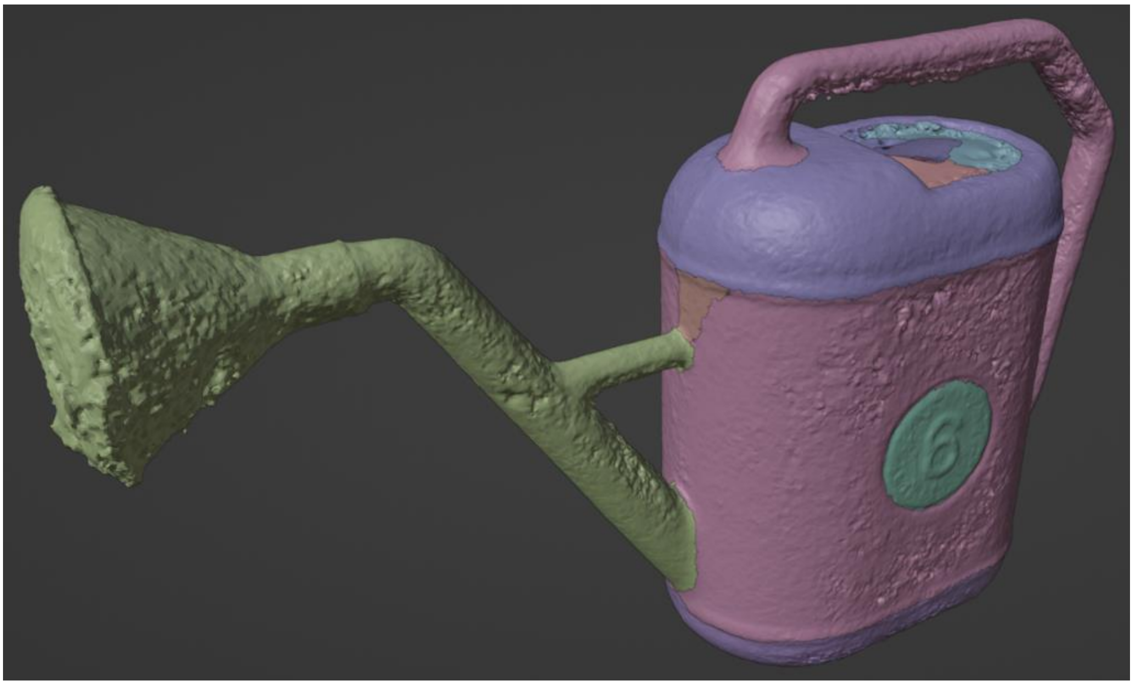


Figure 6.20 - Segmented model using sub-optimal input textured 3D model

Additional to the results obtained from the segmentation process and thanks to the interview with the students, several bugs and potential improvements in the code and process were detected. Some of those issues included:

- Lack of software guidance during the segmentation process.
- Unexpected behaviours while placing reference points for the tracking system.
- Lack of pre-processing steps on the original texture to enhance the feature detection results.
- Low responsiveness of the feature detection process when high resolution input textures are used.
- Lack of clarity in the meaning and impact of the adaptive threshold parameter.

Moreover, some of the students informed that for the rotation of the 3D model, which was a function integrated into the developed add-on, they used the built-in functions of the 3D modelling software instead. His reason was that it was a simple operation, so they did not see the need to use the add-on.

6.4 Discussion

The tests presented in this section iteratively increased the number of variables and the level of fidelity to an actual use case, and each provided valuable insights.

The first test used synthetic input data, allowing to development of the core functionality of the proposed approach. First, the algorithm extracted the texture boundary features without being affected by changes in the colour or luminance around them. Then, the information was used as input to segment the 3D model. Moreover, this activity provided valuable data to understand the impact of different parameters in the feature detection step, the processing time and segmentation results. Although the joint use of adaptive thresholding and dilate iterations demonstrated its capability to properly detect the boundary features in the texture, not all parameter combinations generated successful results. Hence, it requires an evaluation of the capacity of users to select such parameters.

Additionally, it was noted that some parameters significantly influenced processing time due to their relation to the amount of calculation needed to assign the polygons to the corresponding segments. These parameters include the resolution of the mesh, the number of polygons selected using the extracted features as an input, and the number of output segments. Since reconstructed surface models are typically high resolution, special attention must be given to the optimization of the algorithm to handle the previously mentioned calculations.

The second test considered the evaluation of the proposed mesh segmentation on an actual physical prototype. This activity demonstrated that previously obtained results with synthetic data could be replicated using a model and texture generated through surface reconstruction. The extraction of texture features worked as expected, recognizing the added boundary features on the surface of the reconstructed model, and transferring it to the mesh using the original UV map. The algorithm properly segmented the model into twelve segments with different geometrical characteristics, generating new UV maps for each segment. Moreover, the results of the segmentation process were exported and used in a prototyping platform where its functionality was demonstrated in two common tasks of the product development process: material selection and design of user interfaces. Hence, the suitability of the complete proposed workflow to support the mixed prototyping content authoring process was validated.

The third test included a wider variety of physical objects, both prepared by the author and the students, to evaluate the behaviour of the approach on inputs of different geometrical complexity and detect early usability problems. As is expected from most processes, a low-quality input will produce a low-quality output, and the proposed approach is no exception. Although only one of the students prepared the physical object by himself, the results show that no combination of parameters from the proposed approach will be able to properly segment a model that lacks a clear definition of the boundary features in the physical object.

Nevertheless, this problem did not extend to the surface reconstruction process. All the textured 3D models were created by the students learning surface reconstruction methods. Therefore, there was high variability in the quality of the input mesh for the automatic segmentation. Regardless, all the students with a properly prepared physical object achieved satisfactory results. These results align with previous research that shows that photogrammetry is suitable for supporting design tasks (Gebler et al., 2021). Moreover, the results show that the proposed approach can use the surface boundary features as an input to define the segmentation output regardless of the geometrical complexity of such segments. Hence, it allows the generation of unique segments composed of multiple geometrical features and separate unique geometrical features in multiple segments.

Regarding the errors generated on small features with several close boundaries, such as in the handle of the coffee pot, a better segmentation result could have been potentially reached if the input 3D model and texture had a higher resolution. Nevertheless, this solution would also mean a considerable increase in processing times and cost if better equipment and software are used for the surface reconstruction process; hence this is considered a limitation of the proposed approach.

Finally, based on the comments of the students that preferred using the built-in functions for the more manageable steps of the preparation process, it seems that there is a point of diminishing returns when trying to simplify parts of the process of mixed prototyping content authoring that are already simple. Hence focusing the efforts on aiding the users only on the more complex steps rather than the complete process could lead to a more efficient and impactful development process. Indeed, these results are the main reason the second version of the segmentation implementation (presented in section 4.2.2) does not include basic operations such as rotating the 3D model or changing the location of the 3D model origin.

Chapter 7: Extension of functionality and usability

This chapter presents the second research activity of the validation plan, which extends the evaluation of the functionality of the texture-based mesh segmentation approach to unmodified physical prototypes, i.e., using already existing texture features. Moreover, it also focuses on the proposed approach's usability and the ease of use of the developed implementations to validate its impact on reducing the required competence in 3D modelling to prepare a segmented model for mixed prototyping. To achieve this, the proposed approach was tested in fashion design with users with low competence in 3D modelling.

7.1 Selection of the case study

The previous research activities mainly focused on appliances and other home products. Conversely, this research activity focused on fashion products to expand the variety of tested objects, not only in the geometrical aspect but also in the application area. During the design process of these products, selecting surface attributes is necessary, such as colour, texture, embroidery, and carving style (Luh et al., 2013). Due to this, the products in this area, like appliances, can take advantage of the capability of mixed prototypes to easily create design variants by changing the surface characteristics. Indeed, augmented reality technologies have allowed designers and potential buyers to customise products such as footwear (Jimeno-Morenilla et al., 2013) and clothing (Saakes et al., 2016). Moreover, the use of projector-based augmented reality demonstrated to have the added capacity to influence the perception of material stiffness and softness (Punpongsanon et al., 2015, 2018), as well as the capacity to adapt to deformable surfaces (Fujimoto et al., 2015), making it a valuable technology to support the design process of fashion products.

As part of the expansion of the SPARK platform, new users interested in learning about mixed prototyping content authoring were at disposal as test subjects. Thanks to this, testing was done in collaboration with MiTA[1], a technical institute focused on the fashion sector located in the region of Tuscany, Italy.

Additionally, it is worth noting that the process of creating fashion product prototypes is mostly a manual task and due to the materials used they are subject to deformation after handling. Due to this, they are a perfect example of an application area where is likely that CAD models will not be available, and even if they were available, it is likely that the geometries of the 3D model and the physical object would not match perfectly. Hence surface reconstruction methods would be the ideal path to obtain a 3D model with the same geometry as the physical object, fitting perfectly with the workflow of the proposed approach.

7.2 Testing methodology

To evaluate the user interactions and the capacity of the proposed approach to use existing surface features, the experimental setup had the following set of requirements:

---

[1] https://mitacademy.it/

1. The availability of physical prototypes that complied with the requirements described in section 4.1.
2. Test subjects with knowledge in design of the selected object, so they are engaged in the process of mixed prototyping.
3. Test subjects that at the same time have low knowledge in 3D modelling, so they represent the worst-case scenario as final users of the technology.

For the selection of the physical prototype, while MiTA had a wide variety of existing prototypes available, most of them had textured designs, non-neutral colours or reflective surfaces. Hence, they were not suitable for projecting digital information over them. Nevertheless, two leather handbag prototypes matched the surface requirements of mixed prototyping (See figure 7.1). They were constructed using white textured leather and had the borders of each leather patch sealed with a process known as burnishing, which with the additional use of dyes, made them smooth and black. This feature generated high contrast closed boundaries around the segments of the handbag, making them an ideal candidate to test the proposed segmentation approach without modifying the prototypes. Moreover, due to their materials and use outside the mixed reality setup during teaching sessions, they could not be modified. Hence, being a real example of situations where using existing features would be the only option to aid the automated segmentation process. Using these objects, two textured 3D models were reconstructed using the Artec Leo, a structured light scanning device with a maximum resolution of 0.2mm. The resulting models had 297,496 polygons and 223,338 polygons, respectively, for the first and second prototypes, and both had a corresponding texture of 2048 pixels by 2048 pixels.

Figure 7.1 - Physical prototypes (a & c) and their corresponding reconstructed textured 3D model (b & d)

The second and third requirements were addressed through the selection of the sample, composed of professionals attending a course on the use of mixed prototyping applied to the fashion sector (More details of the sample are provided in Section 7.4).

Regarding the testing protocol, in the training stage corresponding to the course lectures, theoretical and practical knowledge on mixed prototyping and mesh segmentation was provided to the subjects using the first prototype. Later, as part of the evaluation process of the course, the test was conducted:

1. A textured 3D model of the second prototype was provided to the students.
2. Students segmented the model using the provided add-on, having the possibility to change two parameters for the feature detection (adaptive threshold and dilate iterations) and the number of segments in the final output.
3. Segmented models used parameters and log files of their interactions were collected.

4. A technology Acceptance Model (Davis, 1989) survey was executed.

It is worth mentioning that the segmentations used for comparison were recreated using the input parameters used by the subjects rather than their output result. This was because the output 3D models were rotated, moved, and scaled by the subjects. Moreover, due to the internal functions of Blender and how it handles some of the segmentation and joining operations, the relative position of the vertices changed slightly from the original unsegmented model. Therefore, it was impossible to use those results for the mesh segmentation evaluation directly.

7.3 Evaluation metrics

7.3.1 Feature detection

As explained in section 4.4, the feature detection step of the proposed approach uses the original texture of the 3D model as input. And the results depend on two parameters to generate a mask that determines which polygons in the mesh will be selected in the segmentation step. The aim is to evaluate the capacity of the approach to use surface features naturally present on a physical prototype, as well as the ability of the users to select the feature detection parameters properly. Hence, their results were compared to a ground truth in which the pixels corresponding to black borders in the prototype were manually selected (see figure 7.2).

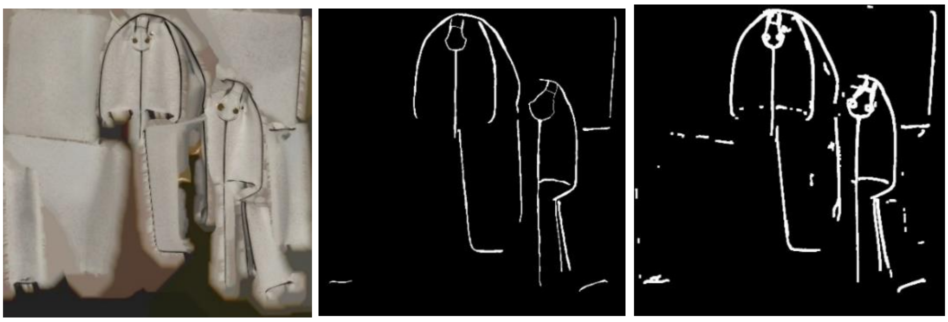
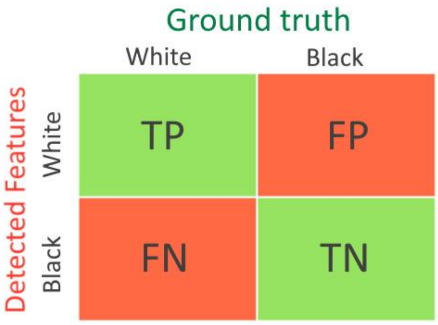


Figure 7.2 - Original texture (left), ground truth (center), and example of automatic feature detection (right)

This comparison can be made pixel by pixel using a confusion matrix (Sammut & Webb, 2017), a tool widely used to evaluate classification problems in computer vision and machine learning. As shown in figure 7.3, depending on the classification of each pixel for the ground truth and the automatic feature detection, each pixel is classified as true positive, true negative, false positive or false negative.



		Ground truth	
		White	Black
Detected Features	White	TP	FP
	Black	FN	TN

Figure 7.3 - Confusion matrix

Using the obtained results from the confusion matrix, for each of the obtained automatic feature detections, two metrics can be calculated:

$$Precision = \frac{True\ positive}{True\ positive + False\ positive} \qquad Recall = \frac{True\ positive}{True\ positive + False\ negative}$$

Precision corresponds to the percentage of features detected by the algorithm correctly classified. Due to the presence of the dilate iteration parameter, it is expected to have a low value, as this function explicitly increases the number of detected features to ensure the creation of closed boundaries in the 3D mesh to allow a proper segmentation. Nevertheless, it is worth mentioning that an abuse of this parameter could also generate unwanted closed boundaries and increase processing times by increasing the number of secondary meshes assigned. Hence there is a trade-off to be made by the user, and any extreme is not ideal.

Recall, on the other hand, corresponds to the percentage of detected features in the ground truth correctly classified by the algorithm. This second metric is expected to have a high value because the adaptive threshold operation should detect the high contrast between the base texture and the black borders. Moreover, the dilate iterations should add any nearby feature left undetected by the previous step. Contrary to the precision, where there was not an ideal value, the recall must be near 1 to ensure that all the existing features in the physical prototype are then used as input to segment the reconstructed model.

7.3.2 Mesh segmentation

The mesh segmentation quality was assessed using the mesh segmentation benchmark tool developed by Chen et al. (2009), which includes well-established metrics to evaluate classification methods adapted to work on 3D meshes. A short description of them can be found in Table 7.1, while a detailed description of the calculation methods is presented in Appendix A.

Table 7.1 - Metrics for mesh segmentation comparison

Metric	Description
Cut discrepancy (Q. Huang & Dom, 1995)	Calculates the overall boundary-based difference between two segmentation results.
Hamming distance (Q. Huang & Dom, 1995):	Calculates the overall region-based difference between two segmentation results.
Rand index (Rand, 1971)	Calculates the likelihood that a pair of faces are either in the same segment in two segmentations, or in different segments in both segmentations.
Consistency error (Martin et al., 2001)	Calculates the overall Region-based difference between two segmentations while eliminating the penalization in differences related to hierarchical granularity.

The benchmark tool was developed to compare the performance of automatic segmentation methods on a dataset of low-resolution untextured objects. Conversely, the model to be evaluated in this test was high-resolution, textured, and not part of the provided dataset. Nevertheless, the existence of a texture didn't affect the comparison process because it uses only the mesh information, also supporting high-resolution 3D models.

To evaluate the segmentation quality of the approach, the subject's results were compared against a ground truth segmentation which was manually done following the existing texture features in the 3D model and resulted in eight independent segments. While the quality of the segmentation results mainly depends on the correct selection of parameters during the feature detection step, the subjects still were allowed to choose the number of segments to obtain as an output. Because of this, it is expected that users that selected the same number of segments as the ground truth will have the lowest errors. Nevertheless, the error added by the difference in the number of segments should only occur in a small section of the overall mesh due to the hierarchical order in which the algorithm defines the resulting segments.

7.3.3 User acceptance

User acceptance was assessed using the Technology Acceptance Model (TAM) proposed by Davis (1989), a tool designed to understand better how companies and individual users adopt new technologies. It has been widely used in research due to its ability to predict the future use of some of the evaluated technologies (Q. Chen et al., 2007), including mixed reality (T.-L. Huang & Liao, 2015; Jang et al., 2021; Sagnier et al., 2020) and 3D modelling tools (Jou & Wang, 2013; G. Wang et al., 2020). Hence, being appropriate to evaluate the user acceptance of the proposed texture-based mesh segmentation in the context of mixed prototyping.

The TAM survey comprises two parts, the first focused on Perceived Usefulness (PU), and the second focused on Perceived Ease of Use (PEoU), each including six statements to be evaluated using a Likert scale of agreements/disagreement. While the PU demonstrated to be more closely related to the future usage of the technology than the PEoU (Davis, 1989), both are important. On the one hand, a high PU could push users to try new technology, but if the PEoU is low, they can be discouraged from using it long-term. While on the other hand, even a system perceived as easy to use will not be adopted by potential users if it does not perform a useful function.

Both parts of this survey are usually applied focusing on the same technological development to better understand its user acceptance. However, this approach was inappropriate for evaluating only a part of the workflow. On the one hand, if only the mesh segmentation software were to be assessed, the subjects that did not need to segment a 3D model in their typical job could not appropriately evaluate the PU of the tool. While on the other hand, if the complete mixed prototyping content authoring process were to be assessed, a better assessment of the PU could be done, but the evaluation of the PEoU would provide limited information on the mesh segmentation. Due to this, it was decided to evaluate the PU, focusing on mixed prototyping in general and the PEoU, focusing on the specific tool for texture-based mesh segmentation. Hence, allowing to verify the PU of mixed prototyping and simultaneously gather information on the particular impact of the proposed approach within the content authoring workflow. The specific statements of the TAM used to evaluate user acceptance can be found in Appendix B.

Based on the benefits of mixed prototyping demonstrated by research found in the literature, it is expected that the PU of this technology would be high, with most of the participants agreeing or strongly agreeing with the statements of the survey. Hence, having the initial motivation to conduct the segmentation process to use this technology.

Additionally, it is also expected that users with low competence in 3D modelling would perceive the proposed method as easier to use than the standard tools in 3D modelling software currently used to segment 3D models. However, due to time constraints and the course's scope, the subjects did not perform the manual segmentation for comparison, preventing this part of the evaluation. Nevertheless, this difference will be discussed at the end of chapter 8 after the PEoU of the manual segmentation has been assessed by subjects with previous experience in 3D modelling.

7.3.4 Log files

The use of data collected directly from software applications while it is being used has proved to be extremely useful for understanding and monitoring user's behaviour (Krieter & Breiter, 2018). This type of data allows a fast evaluation of the results, containing objective and reliable data (Becattini et al., 2019). Moreover, depending on the size and the contents of log files, they can be used as input for advanced processing algorithms, potentially allowing the identification of behaviours that would remain hidden otherwise (Y. Lee, 2019).

Consequently, a log function was implemented in the proposed approach to understand the users' behaviour and gather other useful information. During the interactions, the software reported each of the operations the users did with a corresponding timestamp allowing the calculation of the number of times the users attempted the selection of the surface features, the segmentation of the model, and the total working time. Additionally, each time the model was segmented, a log file was generated reporting the object characteristics, the user input for the number of segments to be segmented, and a detailed list of the processing steps with the amount of time required to execute each of them.

7.4 Sample

The testing activities were implemented within three courses whose main objective was to teach the students how to prepare and use surface reconstruction and mixed prototyping technologies using the SPARK platform. Based on their career path within the institute, the subjects that participated in these courses already had previous experience making leather handbags and using specialised software for this purpose. However, this was their first approach to content authoring using 3D modelling software.

The total sample was separated into two groups based on the algorithm implementation they used during testing. This separation was not planned initially, but due to the timing of each course, there was enough time to develop the standalone implementation.

Hence, it was an opportunity to gather additional information considering this variable while evaluating the mesh segmentation quality and user acceptance.

7.4.1 Course 1

The first course carried out the testing activity using the add-on implementation of the proposed approach and originally included 30 enrolled subjects. Within them, 24 attended the entire course and participated in the testing activity, progressing from a textured 3D model of the physical prototype to a segmented 3D model ready for mixed prototyping. Unfortunately, one of the subjects was excluded from the segmentation evaluation because some of the acquired data was missing, preventing a complete analysis.

7.4.2 Courses 2 & 3

The second and third courses carried out the testing activity using the standalone implementation of the proposed approach and originally included 55 subjects. Among them, 46 attended the entire course and participated in the testing activity, progressing from a textured 3D model of the physical prototype to a segmented 3D model ready for mixed prototyping.

7.5 Results

7.5.1 Feature detection

Figure 7.4 shows a histogram of the precision and recall obtained by the subjects when comparing their detected features against the ground truth detection. Since selecting the feature detection parameters was similar in both implementations of the proposed approach, the evaluation includes the results of the complete sample (70 subjects).

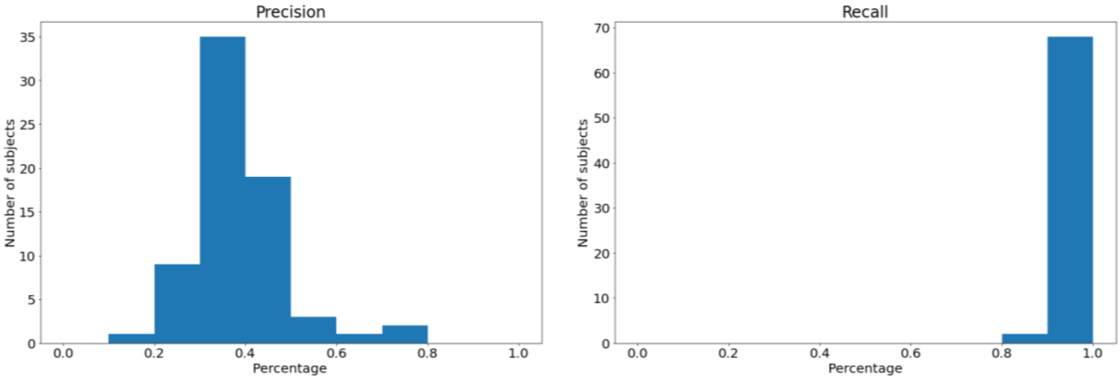


Figure 7.4 - Histograms of Precision and Recall of feature detection using subject parameters

Regarding the *precision*, the mean was 0.400, the 50th percentile was 0.393, and the standard deviation was 0.100. Meaning that from all the detected features using the parameters provided by the students, about 40% were correctly classified, or in other words, most students used parameters that selected more than double the number of pixels than the ground truth selection.

The dilate iteration function explicitly increases the number of detected features to ensure the creation of closed boundaries in the 3D mesh to allow a proper segmentation. Hence, the low precision of the algorithm was expected. In turn, this low precision could also generate unwanted closed boundaries and increase processing times by increasing the number of secondary meshes assigned. Nevertheless, as shown in the following section, these problems were not found in the resulting segmentations, confirming the capacity of the users to select appropriate parameters.

Regarding the *recall*, the mean was 0.980, the 50th percentile was 0.981, and the standard deviation was 0.024. Meaning that the students selected parameters that successfully detected most the features in the texture. Based on the processing steps, the adaptive threshold operation should detect the high contrast between the base texture and the black borders and the dilate iterations should add any nearby feature left undetected by the previous step. Hence, selecting the majority of the features of interest in the texture. This expectation was matched by the results.

Figure 7.5 shows the correlation between the two metrics used for the analysis of the feature detection. As can be seen, the two metrics have a negative correlation, with the *recall* decreasing as the precision increase. Moreover, most of the subjects selected parameter values that generated a relatively low *precision* (0.3 to 0.5) but a high *recall* (over 0.96). Three subjects (two of them with the same values) had higher precision than the others, and because of this, their recall was also the lowest. Under closer inspection of the input parameters, it was noticed that these students also used the lowest values, both in the adaptive threshold and in the dilate iteration parameters. Additionally, and contrary to these cases, one subject obtained the highest *recall* and lowest *precision*, and under the inspection of the input parameters he was the subject that used highest values, both in the adaptive threshold and in the dilate iteration parameters.

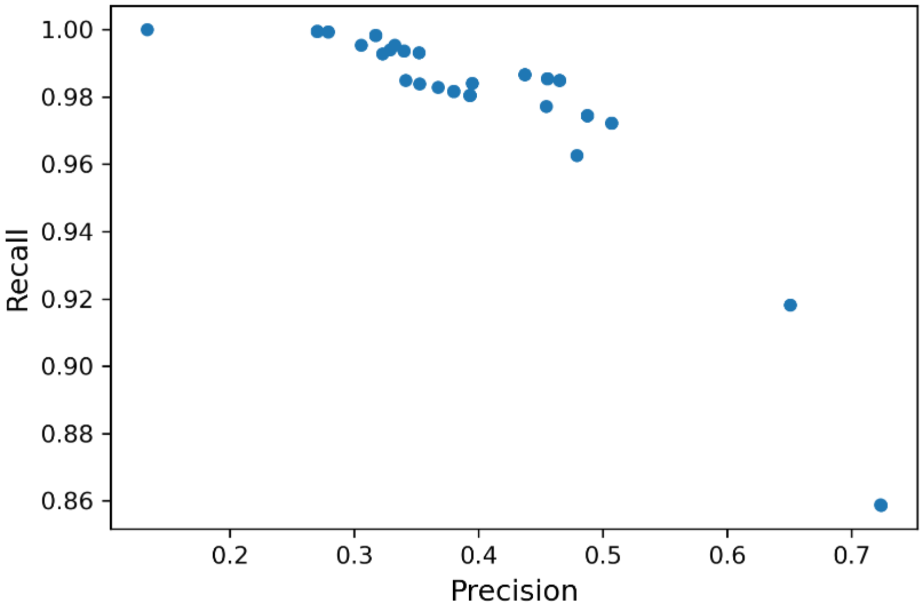


Figure 7.5 - Correlation between precision and recall

7.5.2 Mesh segmentation

Considering the total sample, figure 7.6 shows the number of output segments selected by the subjects. After the mesh segmentation, the subjects were asked to explain the motivation for their choices to understand the discrepancy between their values and the number of segments in the ground truth (8). The rationale behind selecting higher numbers of segments was that some independent boundary features were close to each other. Because of this, they were interpreted by the subjects as a continuation of the same features, generating the perception of a higher number of segments with closed boundaries. Conversely, the selection of a lower number of segments was related to some groups of segments being considered as one because they were small and close to each other.

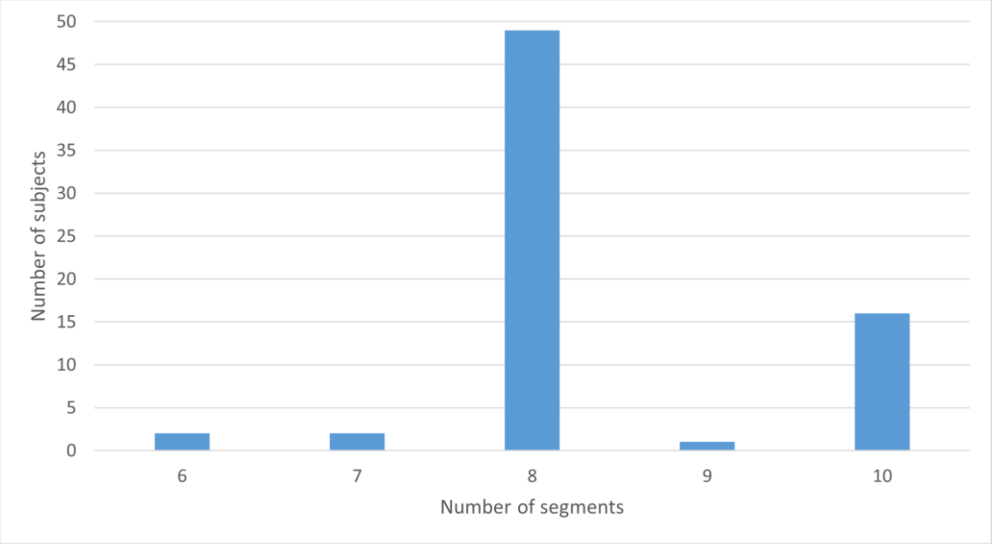


Figure 7.6 - Histogram of output number of segments selected by the subjects

Using the input parameters of the subjects, all segmentations were recreated and compared against the ground truth. The results of the subjects using the Blender implementation are reported in table 7.2, while the results of the subjects using the standalone implementation are reported in table 7.3.

Metric	Min. error	Mean error	Max. error	SD
Cut discrepancy	5.5×10^{-3}	7.1×10^{-3}	9.5×10^{-3}	9.2×10^{-4}
Hamming distance	8.9×10^{-3}	1.0×10^{-2}	1.7×10^{-2}	1.8×10^{-3}
Rand index	6.9×10^{-3}	8.0×10^{-3}	1.4×10^{-2}	1.6×10^{-3}
Local consistency error	1.1×10^{-2}	1.5×10^{-2}	2.2×10^{-2}	2.0×10^{-3}

Table 7.2 - Results of mesh segmentation comparison between subjects and ground truth (course 1)

Metric	Min. error	Mean error	Max. error	SD
Cut discrepancy	5.8×10^{-3}	6.4×10^{-3}	7.7×10^{-3}	7.5×10^{-4}
Hamming distance	9.3×10^{-3}	1.0×10^{-2}	1.3×10^{-2}	1.2×10^{-3}
Rand index	7.4×10^{-3}	8.3×10^{-3}	1.1×10^{-2}	1.1×10^{-3}
Local consistency error	1.3×10^{-2}	1.6×10^{-2}	1.8×10^{-2}	1.3×10^{-3}

Table 7.3 - Results of mesh segmentation comparison between subjects and ground truth (course 2 & 3)

As can be seen, the mean error, maximum error, and standard deviation were lower for all the metrics when using the standalone implementation, but the minimum error was also higher. However, these differences were low in absolute and relative terms, making the distinction between both methods unnecessary for the following evaluations. Considering this, table 7.4 present the results considering the total sample of subjects with both implementations.

Metric	**Min. error**	**Mean error**	**Max. error**	**SD**
Cut discrepancy	5.5×10^{-3}	6.6×10^{-3}	9.5×10^{-3}	8.7×10^{-4}
Hamming distance	8.9×10^{-3}	1.0×10^{-2}	1.7×10^{-2}	1.4×10^{-3}
Rand index	6.9×10^{-3}	8.2×10^{-3}	1.4×10^{-2}	1.3×10^{-3}
Local consistency error	1.1×10^{-2}	1.5×10^{-2}	2.2×10^{-2}	1.7×10^{-3}

Table 7.4 - Results of mesh segmentation comparison between subjects and ground truth (total sample)

Cut discrepancy calculates the overall boundary-based difference between two segmentation results. Therefore, if we consider the dimensions of the physical prototype (214 mm X 151 mm X 166 mm, as reported by the 3D modelling software), a maximum error of 9.5×10^{-3} means that for the worst subject result, on average, the distance between each vertex in the boundaries of the automatic segmentation and the closest vertex in the boundaries of the ground truth, and vice-versa, is 0.834 mm. Considering a projector-based mixed prototyping platform with a projection width of one meter (depending on the placement of the prototype) and a full HD resolution, each projected pixel has a width of 0.52 mm. Hence the average error is less than two pixels, which is a satisfactory result.

Regarding the Hamming distance and Consistency error results, which provide a region-based comparison against the ground truth, the data indicate a high percentage of overlap between segments produced with the proposed approach and the reference segmentation. The Hamming distance, for example, indicates that 98.3% of polygons of the model correctly overlap between both segmentations when comparing the closest corresponding segments. While the Local consistency error, the metric that shows the maximum error overall, indicates that nearly 97.8% of polygons overlap when comparing the segments related to each of the polygons of the model.

In line with the other metrics, the Rand index indicated a high likelihood (98.6% for the worst subject result) that a random pair of faces from the evaluated model was part of the same segment on the generated result and the reference segmentation, or in different segments in both segmentations.

To better visualise these results, figure 7.7 compares the ground truth segmentation and the automatic segmentations with the highest and lowest overall errors. As can be seen and interpreted from the error results, even considering the discrepancy in the number of output segments, the subject's output segmentation still closely followed the ground truth segmentation. Additionally, it is worth noting that the segmentation with the highest overall error was from a subject that selected an output of 6 segments. Consequently, the polygons that conformed the smaller segments present in the ground truth and lowest error segmentations were assigned to the closest remaining segments.

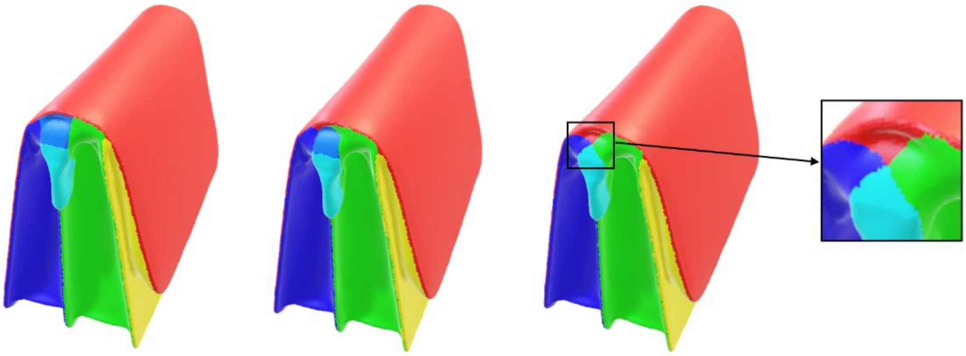


Figure 7.7 - Segmentation results: ground truth (left), lowest overall error (center), and highest overall error (right) with close-up of its boundary

7.5.3 User acceptance

Figure 7.8 shows the results for the first part of the TAM survey, focused on the Perceived Usefulness of mixed prototyping to aid the design process of fashion products. Since the mixed prototyping platform did not change during the testing activity, the subjects' responses in all three courses were analysed.

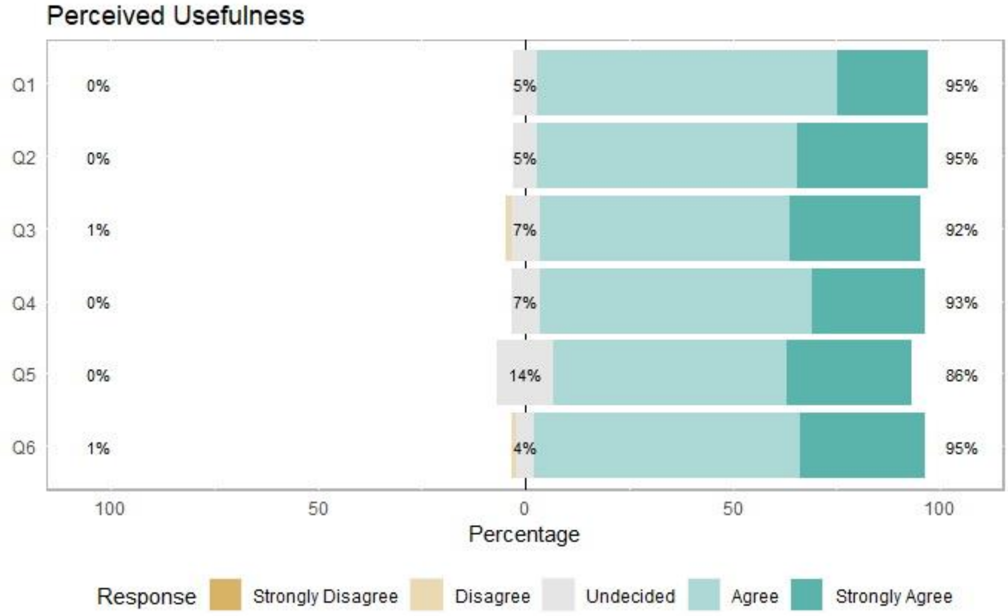


Figure 7.8 - Perceived Usefulness of SPARK (total sample)

As can be seen, for each statement, at least 86% of the subjects agreed, with most obtaining over 90% of agreement. These results suggest that the subjects will have at least the initial motivation to try the technology and that there is a high likelihood that they will use the technology in the future if no other problems arise. An interesting result includes a relatively high number of neutral responses (14%) in statement 5, which indicate that the mixed prototyping platform could simplify the work done by the subjects. Although the survey does not provide more information about the reasoning behind the subject's selection, this result could be an early indication of the adoption barriers affecting this technology. In particular, the contrast between the complex authoring process of mixed prototypes and the primarily manual process currently applied for the design of fashion products could generate some doubts in the user about the usefulness of the system.

Regarding the second part of the TAM survey, focusing on the Perceived Ease of Use of the developed implementations, figures 7.9 and 7.10 show the results using Blender and the standalone application, respectively.

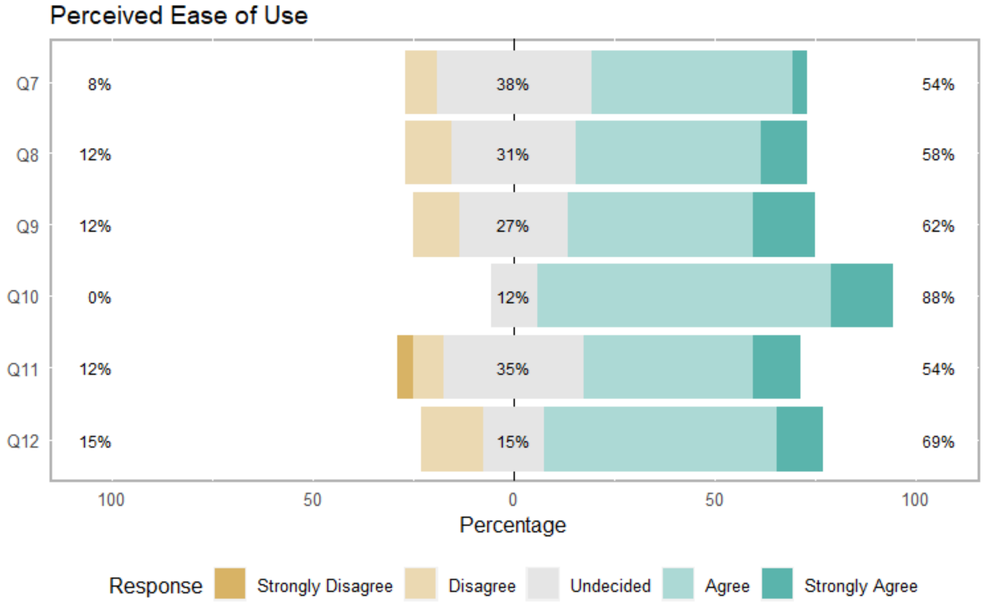


Figure 7.9 - Perceived Ease of Use of the blender implementation (Course 1)

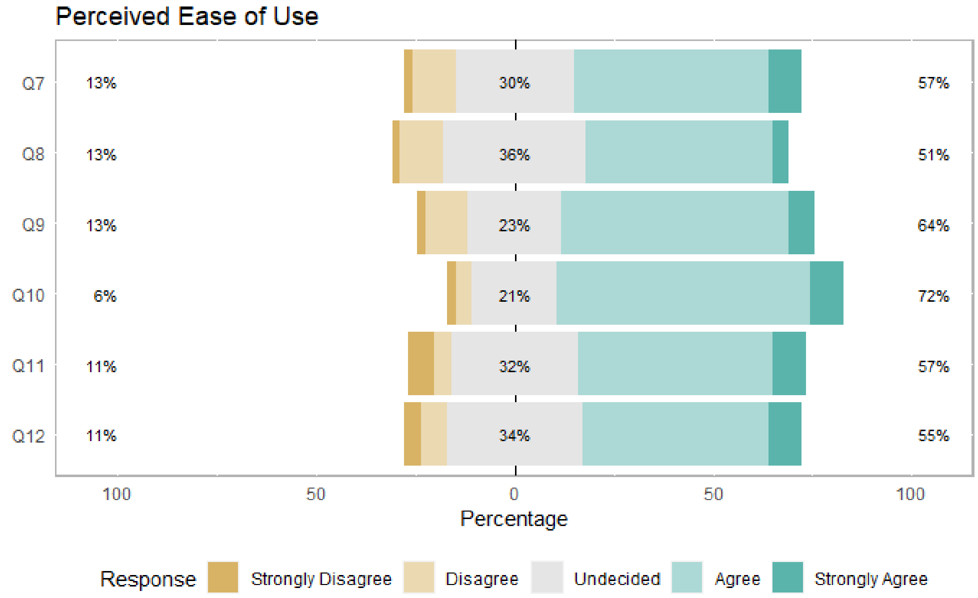


Figure 7.10 - Perceived Ease of Use of the standalone implementation (Course 2 & 3)

Most of the subjects agreed with the statements indicating the Ease of Use of the implementations. However, contrary to the Perceived Usefulness, there was a high percentage of subjects that either provided neutral responses or disagree with the presented statements (ranging from 12% to 49% depending on the statement and method used). Some interesting results include the 11th statement, that indicated that "it would be easy to become skilful at using the tool/add-on", which obtained the highest percentage of strong disagreement, regardless of the implementation. Similarly, the 10th statement, that indicated that "the tool/add-on used was flexible to interact", obtained the highest percentage of agreement (88% with the Blender implementation and 72% with the standalone implementation), regardless of the implementation.

Even though the proposed approach was developed having the ease of use and reduction of knowledge requirement as objectives, and most of the subjects agreed with the statements, the results seem to indicate that there is still an adoption barrier in this regard. Nevertheless, as mentioned in the section 7.3.3, to get the full context of these results, they should also be evaluated against a segmentation process carried out using the standard tools of 3D software, which is the current method used to prepare mixed prototypes. This comparison will be done at the end of the chapter 8, after data from users with experience in 3D modelling has been gathered.

Additionally, the differences in the Perceived Ease of Use depending on the implementation used were assessed through statistical analysis. Due to the non-parametric data gathered through a Likert scale survey and the independent samples, the Wilcoxon rank sum test was used. The results show that the impact on the perceived ease of use when changing between the Blender add-on and the standalone implementation was not statistically significant ($p = 0.399$).

7.5.4 Log files

Regarding the processing speed of both implementations, a comparison was made using the parameters that produced the best segmentation results. The Blender implementation required 400.48 seconds to segment the handbag 3D model, while the standalone implementation required only 3.53 seconds. This test was conducted on a computer with an Intel i7 9750H processor and 32 Gb of RAM. Due to the significant difference in processing times, the following evaluations of the log files only consider the results of the subjects using the standalone implementation.

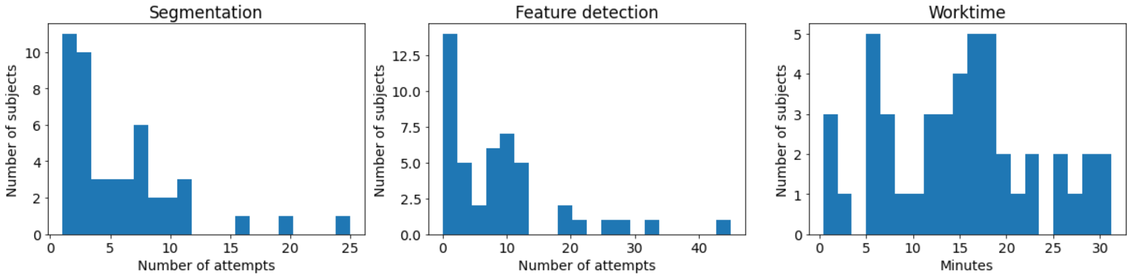


Figure 7.11 - Results of log data evaluation

Figure 7.11 shows various histograms of data generated from the log files. As can be seen from the histograms of the number of attempts, 86.95% of the subjects finished the segmentations within 1 to 10 attempts, each requiring about 4 seconds. Similarly, 59.56% of the subjects completed the feature detection within 1 to 10 attempts, each requiring less than a second. Regarding the total work time to execute the segmentation of the models, there was no clear distribution of the results, ranging from 24 seconds to 31 minutes and 12 seconds.

7.6 Discussion

The test presented in this chapter extends the evaluation of the functionality of the texture-based mesh segmentation approach to unmodified physical prototypes, i.e., using already existing texture features. Moreover, it also focuses on the proposed approach's usability and the ease of use of the developed implementations to validate its impact on reducing the required 3D modelling knowledge to prepare a segmented model for mixed prototyping.

Concerning the suitability of texture-based mesh segmentation, using already existing texture features for automatic segmentation poses different challenges than detecting features added explicitly for this purpose (Piñones et al., 2021). Nevertheless, the results suggest that using computer vision techniques such as adaptive thresholding and dilate iterations is also suitable for feature detection in this context. Hence, if such high-contrast features coincide with the boundaries of the segments to be separated, a computer-vision-based approach will be capable of generating a part-based segmentation that replicates those boundaries. This capability is demonstrated by the resulting segmentations of a handbag shown in figure 7.7, and comparable results are expected on other types of products of similar surface characteristics.

Regarding the quality of the mesh segmentation, all four metrics used to evaluate the dissimilarities to the ground truth segmentation reported low errors, with 3.0×10^{-2} being the highest value among them for the subject with the worst segmentation. Hence, reaching values close to zero meant that the automatic segmentations were almost identical to the ground truth segmentation. Moreover, the proposed approach was demonstrated to be robust to variability in the input parameters, as it could generate segmentations with a low error compared to a ground truth segmentation for any of the parameter combinations used by the participants. Additionally, in the context of projector-based mixed prototyping, the obtained errors would be barely noticeable, with a mean value equivalent to less than 2 pixels in the position of the boundaries when compared to manual segmentation. Hence, these results indicate that the proposed approach can produce segmentation results that closely replicate a manual segmentation and that the errors in the boundary selection are acceptable in the context of mixed prototyping.

Regarding the usability of the approach, the results showed that the subjects, who had low expertise in 3D modelling, chose parameters that detected most of the features to be used to segment the 3D model. Consequently, the subjects generated a properly segmented 3D model for mixed prototyping without manually intervening in the mesh. In other words, the proposed texture-based approach demonstrated its capability to reduce the 3D modelling competence requirements for mixed prototyping content authoring. Therefore, this change is expected to positively influence user acceptance and technology adoption (de Souza Cardoso et al., 2020). However, one element not considered at the beginning of testing was the variability in interpreting the amount and shape of segments that the subjects detected and used as input. Because of this, while the segmentations closely replicated the ground truth segmentation, they did not necessarily match the subjects' expectations when selecting the parameters. This situation was overlooked because the boundaries, and by extension, the segments, were clearly defined, and it was expected that the subjects had a similar interpretation.

Additionally, regarding the processing speed, there was an improvement of two orders of magnitude when changing from the Blender implementation to the standalone implementation. Although this improvement is related to the optimisation of the code and did not influence the quality of the segmentation results, a lower work time could positively impact the user acceptance of the technology.

Finally, regarding the limitations of the testing methodology, although the selected physical prototypes were an ideal case to apply this approach due to their surface characteristics and geometry, the final evaluation was done on only one object. Hence, the result could include a bias towards similar prototypes and not entirely represent the expected results on other objects. Moreover, only one ground truth segmentation was used to evaluate the segmentation quality, which could have provided limited insight into the results, as two observers do not necessarily share the same opinion on the segmentation of a model (Benhabiles et al., 2009). Nevertheless, these limitations are addressed in the following chapter.

Chapter 8: Comparison to manual method

This chapter presents the third research activity of the validations plan, which evaluates the differences between a segmentation done using the texture-based mesh segmentation approach, and the standard tools in a 3D modelling software.

Section 3.4.2 provides the reasoning behind the comparison being made. It is suggested that existing automated methods for mesh segmentation have different objectives from the proposed method, thus comparing them would provide limited information. Hence a comparison against their results would provide limited information. Nevertheless, the current method for mesh segmentation of reconstructed surface models used in mixed prototyping can still provide helpful reference information for comparison. In particular, due to the lack of existing tools to support this process, the mesh segmentation is currently done by the designers or developers using the standard tools provided by existing 3D modelling software.

The evaluation is done from multiple perspectives, including the geometrical quality of the segmentations, the execution time of the segmentation process, and the perceived ease of use when using each of the methods.

8.1 Selection of the case study

Due to the focus on comparing the segmentation quality between the process conducted manually and using the proposed approach, the main requirement was the participation of subjects with competence in 3D modelling. To comply with this requirement, this test was made in collaboration with the department of design engineering of the Technical University Federico Santa María in Chile. In particular, the test subjects were students in their 2nd year of product design engineering enrolled in the “Virtual representation of products” course. These students previously approved a course focused on 3D modelling using CAD software, and at the time of testing within the course, they also had experience working on freeform models.

8.2 Testing methodology

For this activity, two products were prepared following the requirements presented in section 4.1.2, a Kettle, and a Hairdryer (see figure 8.1).

Figure 8.1 - Models used for comparison of segmentation methods

While both had previously defined boundary features in their surface and a similar level of geometrical complexity, the later had a higher number of surface features not related to the segmentation boundaries (e.g., dust, and shadows generated by crevices). Due to this difference, it is expected that the participants processing the hairdryer would require more time or produce less accurate results when using the proposed approach. The provided models had 461,191 polygons and 191,250 polygons, respectively for the kettle and hairdryer, and both had a corresponding texture of 2048 pixels by 2048 pixels.

In preparation for the test, the participants went through a training stage where the process of segmenting textured meshes was explained using the manual and the texture-based methods. This was done regardless of the subjects' experience level, as mesh segmentation corresponds to a specific process within 3D modelling. During this process, primitive geometries and a reconstructed surface model of a handbag were used for demonstration. Once the training stage was finished, the test was conducted:

1. Textured 3D models were provided to the subjects depending on their experience with wireframe modelling. For subjects with less than one year of experience, one 3D model was provided, while for the advanced users, two models were provided. This was done because it was expected that subjects with less experience would take more time to execute the activity. Moreover, it allowed to increase the number of reference segmentation considering the lower availability of subjects with advanced knowledge in 3D modelling. The models were distributed, aiming to have a similar number of results from each of them.
2. Each subject was informed of the order they had to conduct the segmentations. The order of execution was assigned randomly to the participants, aiming to have half of the subjects start with manual segmentation and the other half start with texture-based segmentation.

3. The subjects were asked to segment their corresponding models based on the texture features (i.e., following the black lines on the surface of the models) using the manual and texture-based methods following the assigned order.
4. The segmented models, the time required to execute each process, and log files of their interactions were collected.
5. The subjects were asked to answer the Perceived Ease of Use survey two times, one for each method.

The mesh segmentation quality was assessed using the mesh segmentation benchmark tool developed by Chen et al. (2009) presented in section 7.3.2. While the perceived Ease of Use was evaluated using a part of the Technology Acceptance Model survey (Davis, 1989) presented in section 7.3.3.

Due to the timing of this test, at the end of the PhD program, the participants were requested to use the standalone application. Hence, allowing the comparison of the ease of use of the manual process against the latest implementation of the approach.

8.3 Sample

From the course “Virtual representation of products” of the Technical University Federico Santa María, 37 students separated into two groups (18 and 19) participated in this activity. Victor Urrutia, the course professor, supported executing the testing activity during their regular lecture schedule.

Additionally, as part of the elective course “Authoring of Digital Applications” conducted at the Politecnico di Milano by the author, where the students were taught wireframe modelling and 3D animation, two additional students participated in the testing activity.

Moreover, to use as a reference for the comparison of the segmentation quality, five professionals with more than one year of experience in freeform modelling, were also requested to participate in the test.

In total 44 subjects were part of this research activity and almost all of them executed the test completely in remote, with the two students of the Politecnico di Milano being the only exception. Considering the proposed methodology and the number of participants, a total of 98 resulting segmentations were expected, 20 generated by the advanced users and 78 generated by the students (See figure 8.2).

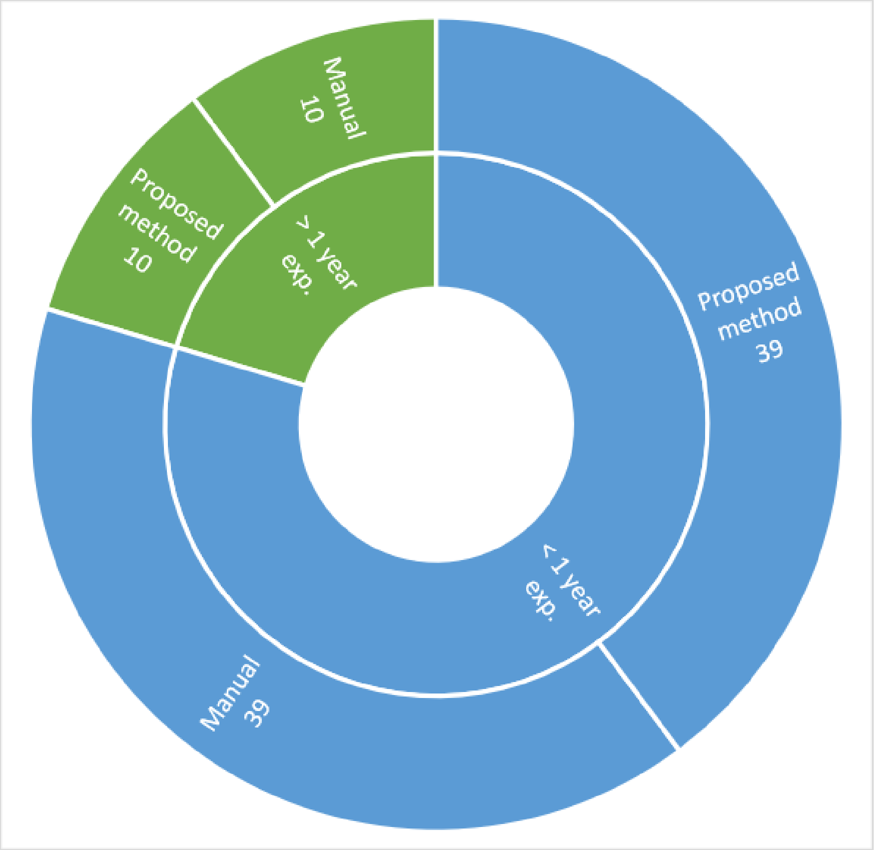


Figure 8.2 - Expected segmentation sample distribution

8.4 Results

As mentioned previously, most of the subjects (42 out of 44) participated in the testing activity remotely. This situation affected the quality of the communication with the participants, generating some difficulties during the training stage and on the gathering of the data, especially with the students of the course “Virtual representation of products”. Even though a professor was physically present with them, some factors could not be fully controlled, such as participants missing parts of the explanations or pausing during some of the activities.

These problems directly affected the results of the test, making necessary the exclusion of some of them. Nevertheless, the remaining results still provided sufficient data to be evaluated and the statistical analyses shown significant differences between both segmentations methods.

Within the gathered results, those that did not comply with the segmentation requirements (i.e., followed the black lines on the surface of the models), were directly considered as failed and excluded from the error calculations. Examples of failed segmentations include 3D models that were over segmented or missed the segmentation of significant parts of the model. Consequently, the results show that the success rate for the proposed methods was 82.60%, while the success rate for the manual segmentation was 71.11%.

Concerning the results using the manual segmentation, 17 of the 39 expected segmentations were excluded from the error calculations, 13 due to failed segmentations and four due to missing files. Moreover, regarding to the results using the proposed methods, 11 of the 49 expected segmentations were excluded from the error calculations, eight due to failed segmentations, and three due to missing data. The final distribution of the sample for the error calculations is presented in figure 8.3.

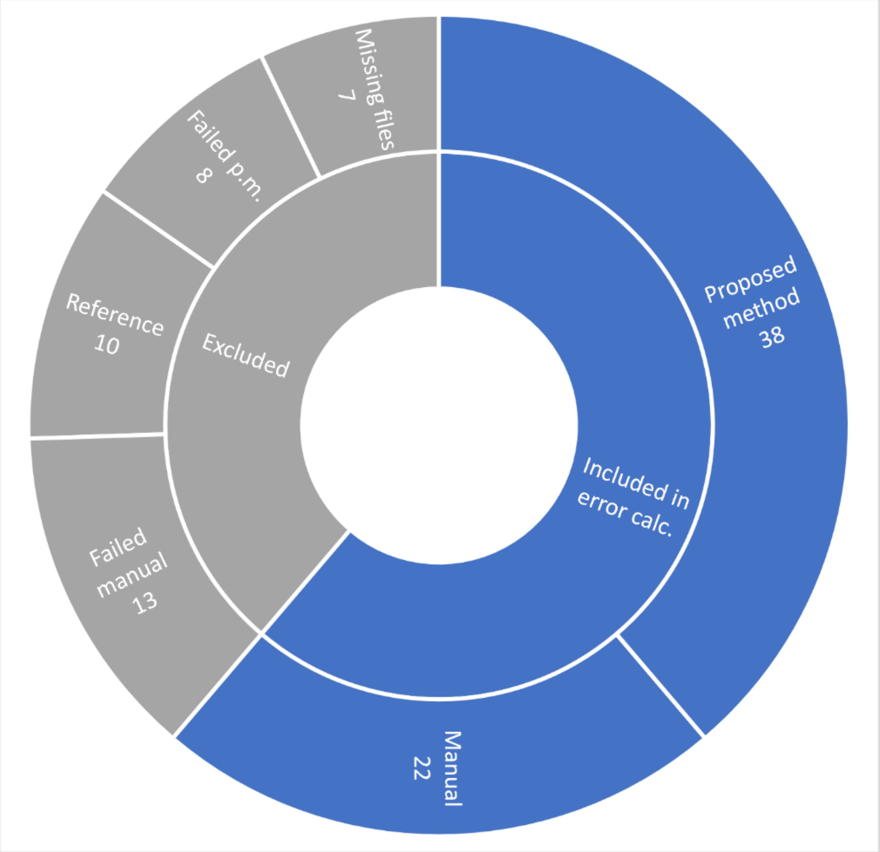


Figure 8.3 - Distribution of segmentation results for error calculation

Additionally, even though some students properly conducted the segmentation using the proposed method, five log files were totally or partially missing, and two subjects registered long periods of inactivity, preventing an accurate calculation of the processing time.

8.4.1 Segmentation quality and execution time

The first analysis compares the results obtained by the subjects using the proposed approach and the existing tools on 3D modelling software. Figure 8.4 presents the overall distribution of these results considering their error against the reference segmentations and the time required to execute the segmentation process.

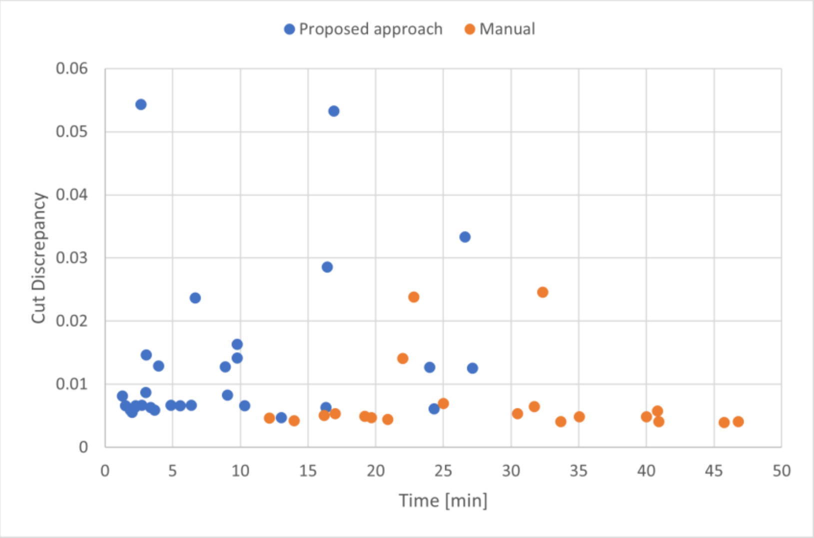


Figure 8.4 - Error vs execution time by method

The data distribution shows that the proposed method's segmentations tend to require less execution time than the manual segmentation, although having a higher difference against the reference segmentations. Moreover, when focusing only on the manually segmented models, it can also be appreciated that the difference against the reference segmentations remains low regardless of the execution time, with only a small number of subjects with a higher error.

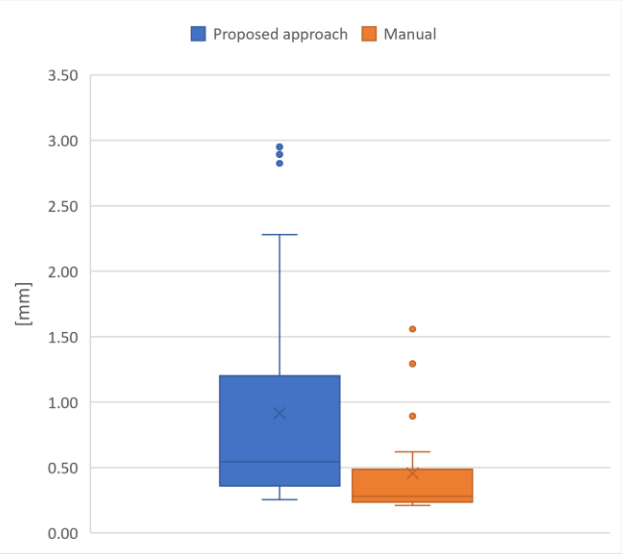


Figure 8.5 - Average distance between the boundaries of the segmentation results and the boundaries of the reference segmentations

Focusing on the difference against the reference models, figure 8.5 presents the conversion of the Cut Discrepancy results to the average distance between the boundaries of the segmentation results and the boundaries of the reference segmentations. While the proposed method does have a bigger difference against the reference segmentations when compared to the manual segmentations, the average distance between the boundaries that was still low in absolute terms, with most of the results being less than 1 mm.

Like the Cut Discrepancy, all the other metrics presented similar distributions, with the manual segmentation performing better in terms of segmentation quality (See Figure 8.6).

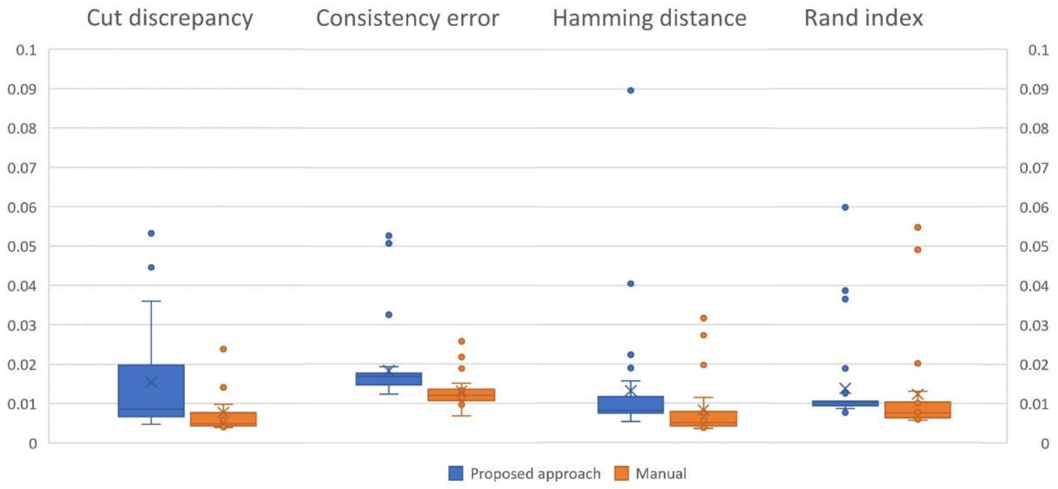


Figure 8.6 - Error by method and metric

Under a closer inspection of the results with higher error and failed segmentations, it can be noticed that the discrepancies between the segmentations come from two distinct types of resulting segments (see figure 8.7). First, small segments added due to the selection of a high number of output segments compared to the reference segmentations. And second, segments created due to texture features that generate unwanted closed segments with a more extensive surface area than the expected closed segments, hence, being erroneously added to the results. The latter has a more significant impact on the results due to their size relative to other segments. It is worth noting that both types of errors can be considerably reduced or eliminated on the test objects when the proper feature detection parameters and number of output segments are selected.

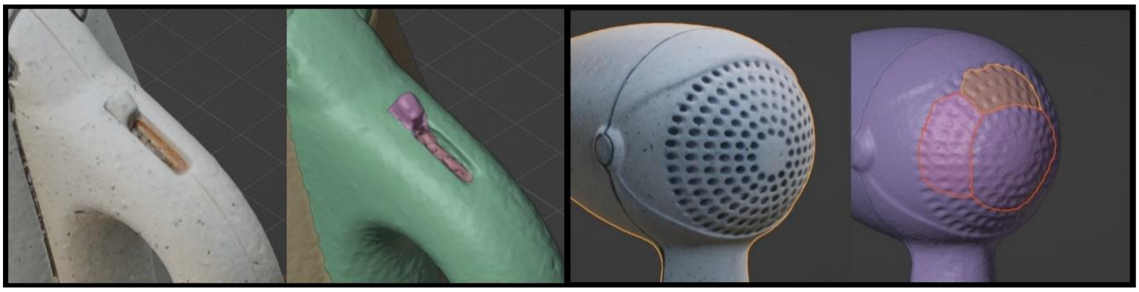


Figure 8.7 - Segmentation errors due to high number of output segments (left) and texture features creating unwanted closed segments (right)

Focusing only on the execution time, figure 8.8 shows the distribution of the results, while table 8.1 presents the corresponding summary statistics. The data shows that the mean time required to segment the models using the proposed approach was nearly 1/3 of the mean time required to segment the models manually. Indeed, the minimum execution time of using the proposed approach was more than 8 minutes faster than the minimum execution time for the manual segmentation. Similarly, the maximum execution time of the proposed approach was almost 20 minutes faster than the maximum execution time for manual segmentation. Nevertheless, using the proposed approach, some subjects executed the manual segmentation faster than the slowest segmentation. Regardless, when considering all the results, the data indicate a clear benefit in reducing execution time when using the proposed approach. This result was statistically validated through a paired t-test (See table 8.2).

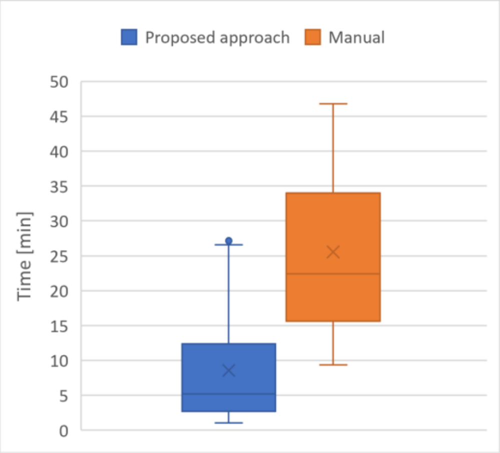


Figure 8.8 - Execution time by method of segmentation

	Proposed approach	Manual
Minimum	1.05	9.35
1st Quartile	2.71	16.09
Median	5.22	22.40
Mean	8.56	25.53
3rd Quartile	10.99	33.63
Maximum	27.16	46.78
Standard deviation	7.95	11.42

Table 8.1 - Summary statistics of execution time [min] by method of segmentation

t-value	Degrees of freedom	p-value
-7.8562	24	<0.001

Table 8.2 - Results of paired t-test for execution time by method

As stated previously, it was expected that the hairdryer model would require more time due to the inclusion of additional surface features unrelated to the model's segment boundaries. For this reason, the difference in execution time using the proposed approach for the provided models was also evaluated. As shown in figure 8.9 and the corresponding summary statistics in table 8.3, the distribution of the execution time for the hairdryer model was higher than that of the kettle models. However, a two-sample t-test (See table 8.4) shows that the difference is relatively small and that the p-value is considerably higher than 0.05. Hence, not being statistically significant.

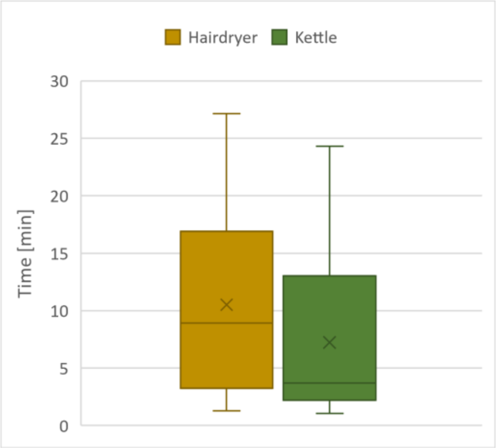


Figure 8.9 - Execution time of proposed approach by model

	Hairdryer	Kettle
Minimum	1.30	1.05
1st Quartile	3.43	2.21
Median	8.90	3.68
Mean	10.51	7.23
3rd Quartile	9.76	11.67
Maximum	27.16	24.31
Standard deviation	9.25	6.87

Table 8.3 - Summary statistics of proposed method execution time [min] by model

t-value	Degrees of freedom	p-value
1.0868	20.789	0.2896

Table 8.4 - Results of two-sample t-test for execution time by method

8.4.2 Perceived Ease of Use

Figures 8.10 present the results of the Perceived Ease of Use survey for manual segmentation, while figure 8.11 present the results related to the proposed approach. As can be appreciated from the percentage of responses agreeing with the statements, the proposed approach performs better on every survey statement. Hence, the results indicate that the proposed approach is perceived as easier to use than manual segmentation.

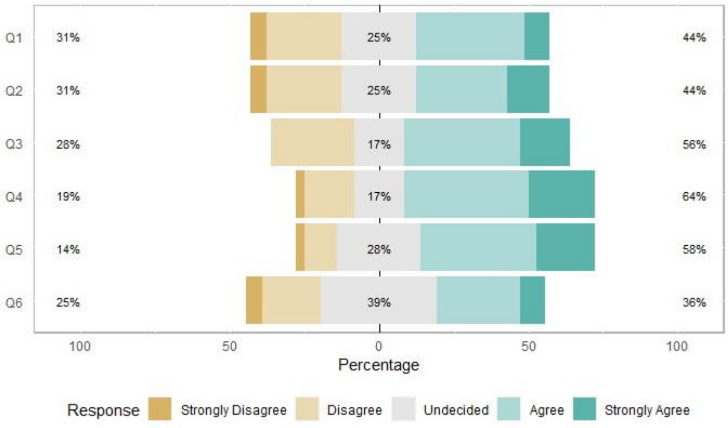


Figure 8.10 - Perceived Ease of Use - Manual segmentation

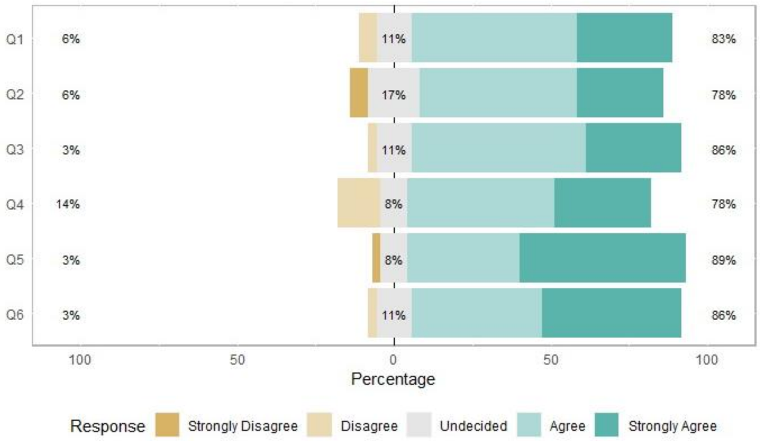


Figure 8.11 - Perceived Ease of Use - Texture-based mesh segmentation

One interesting result was the distribution of the responses in Q4, which was at the same time statement with the best result of the manual segmentation and one of the worst results of the proposed approach. This statement refers to the flexibility of the tool used to execute the segmentation and hence, the result was to be expected. While the manual method offers the possibility to use all the typical operations of a 3D modelling software, the more automated proposed method offers the user only a small number of parameters to be modified.

Figure 8.12 shows a histogram of the difference in the Perceived Ease of Use among both methods for each of the participants. As can be seen by the distribution of the data, most of the participants (82.0%) had difference higher than zero, meaning that perceived the proposed methods as easier to use than the manual methods. Nevertheless, a small number of the participants (12.8%) perceived the manual method as slightly easier to use than the proposed methods.

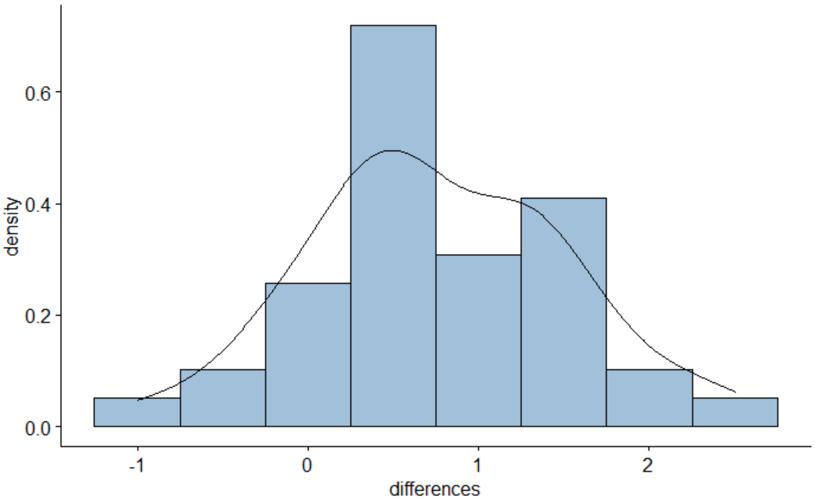


Figure 8.12 - Histogram of difference in Perceived Ease of Use (Manual to Texture-based mesh segmentation)

Due to the non-parametric nature of data gathered through a Likert scale survey, a statistical analysis was carried out using the Wilcoxon signed rank test (see table 8.5). The results shown that the impact on the perceived ease of use when changing between the manual and the proposed method was large, and statistically significant.

Sample	Effect size (r)	p-value
39	0.748 (Large)	< 0.001

Table 8.5 - Results of Wilcoxon signed rank test on Perceived Ease of Use by method

Finally, the effect of the subject's experience in 3D modelling in the perceived ease of use of the proposed approach was assessed. Figure 8.13 shows the average Perceived Ease of Use results obtained during this test (i.e., executed by subjects with previous experience in 3D modelling) against the results presented in chapter 7 (i.e., executed by subjects with low competence in 3D modelling). Due to the non-parametric nature of data and the independent samples, the Wilcoxon rank sum test was used (see table 8.6). The results show that the subjects with previous experience in 3D modelling perceived the proposed method as easier to use than the users with no previous experience. Moreover, this difference was considered moderate and statistically significant.

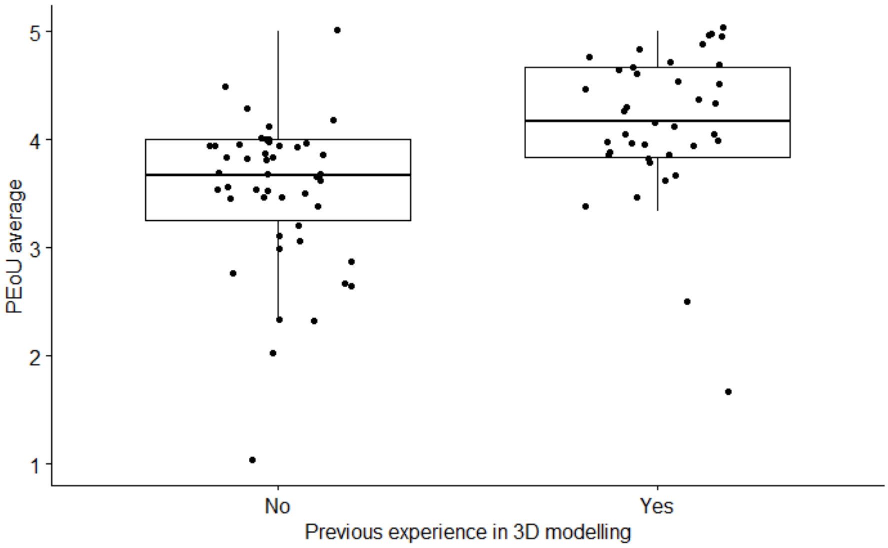


Figure 8.13 - Boxplot of average Perceived Ease of Use by experience in 3D modelling

Sample by exp. in 3D modelling		Effect size (r)	p-value
No	Yes		
47	39	0.492 (Moderate)	< 0.001

Table 8.6 - Results of Wilcoxon rank sum test on Perceived Ease of Use by experience in 3D modelling

8.5 Discussion

The objective of this testing activity was to compare the results of a mesh segmentation using the texture-based mesh segmentation method and the standard tools available in a 3D modelling software. The comparison was focused on the quality of results relative to reference segmentations made by advanced users, the time required to execute the segmentation process, and the perceived ease of use of each method.

During this activity a high number of subjects failed to execute the segmentation of the selected models, obtaining a success rate of 82.60% when using the proposed method and 71.11% when the segmentation was done manually. This result was not expected, as in the previous research activity, where the subjects had low knowledge in 3D modelling, all the subjects managed to get a proper segmentation. One potential reason for this could be the difficulties in the communication generated due to the remote testing, as the previous test was completely carried out in person. Nevertheless, these difficulties should have affected both parts of the testing similarly, and a relative comparison could still be valuable. In this perspective, the 11.49% difference between the success rate of both methods could indicate that the proposed approach was easier to learn and use than the manual mesh segmentation.

Regarding the quality of the successful segmentations, the results show that the users tend to generate less accurate results in terms of segmentation quality when using the proposed approach than when using the tools of 3D modelling software. Moreover, the texture-based segmentation also presented a higher dispersion than the manual method. Nevertheless, the errors obtained using the proposed method were low in absolute terms, meaning that even though the results are not as accurate as a manual segmentation, they could still be suitable for applications where that error level is acceptable.

Regarding the differences in the processing time of each method, the mean time of execution using the proposed approach was nearly 1/3 of the mean time required to segment the models manually. Moreover, the fastest execution time using the proposed approach was 1 minute and 3 seconds, considerably lower than the 9 minutes and 21 seconds obtained by the fastest manual segmentation. Hence, demonstrating significant benefits in terms of time reduction when using the proposed approach against manual segmentation. However, it is worth noting that some subjects executed the manual segmentation faster than the slowest segmentation using the proposed approach. A potential explanation for these results was the differences in the previous 3D modelling experience of the subjects, expecting that the advanced users would finish faster than subjects with less experience. Nevertheless, after closer inspection of the data, it was noted that all the advanced users executed the segmentation faster using the proposed approach, hence discarding this explanation.

Finally, regarding the Perceived Ease of Use, the results shown that the subjects perceived the proposed method as easier to use than the standard modelling tools to execute the mesh segmentation. Nevertheless, a small number of the participants perceived the manual method as slightly easier to use than the proposed method. Initially, this was thought to be correlated to the experience of the subjects with the software for manual segmentations, making them perceive it as easier to use than a new software. However, under a closer inspection of the data, four of the five participance with most experience in 3D modelling indicated that they perceived the proposed method as easier to use. Hence, discarding this potential explanation. Moreover, no other traits of these subjects were found that could support other explanation.

Additionally, the comparison against the results presented in the chapter 7 show that the subjects with previous experience in 3D modelling perceived the proposed method as easier to use than the users with no previous experience. One potential explanation for this difference could be that the previous experience in 3D modelling software served as a reference point for comparison, favouring the assessment of the proposed method due to its lower complexity. Moreover, the difference in the type of object used during testing could also have affected the results, as the handbag had less surface features not related to the segment boundaries than the kettle and hair dryer. Hence, potentially easing the finding of the correct feature detection parameters.

Chapter 9: Discussion

This chapter presents a general discussion of the research outcomes in response to the research questions, as well as the identified limitations related to the research methodology and the proposed approach.

9.1 Answer to research questions

RQ1: Can a computer vision supported process generate a segmented 3D model suitable for mixed prototyping?

RQ1.1: Using added graphical mark on the surface of the physical prototype.

RQ1.2: Using existing features on the surface of the physical prototype.

To address this research question, a texture-based mesh segmentation approach that used the texture features of reconstructed surface prototypes to define the boundaries of the output segments was developed. Moreover, multiple tests were conducted using input models with various characteristics (i.e., surface features, mesh resolution, texture resolution, segmentation requirements and surface reconstruction methods).

As an initial step toward validating this approach, the solution was tested using prototypes with added graphical marks defining the boundaries of the segments. As demonstrated in chapter 6, through a combination of two image processing operations (i.e., adaptive thresholding and dilate functions), the proposed approach could identify such features and use them to segment the input model. Moreover, due to the use of texture information rather than geometrical information during the segmentation process, the output segments were not dependent on their geometrical complexity. Hence, it allows the generation of unique segments composed of multiple geometrical features and separate unique geometrical features in multiple segments.

Similarly, when using existing features on the surface of the physical prototype to define the boundaries of the segments, the proposed approach demonstrated to be capable of identifying such features and using them to segment the input model. While using already existing texture features for automatic segmentation offers reduced control over the results, the same combination of adaptive threshold and dilate functions remains suitable for feature detection in this context. Hence, if such features coincide with the boundaries of the segments to be separated, a computer-vision-based approach will be capable of generating a part-based segmentation that replicates those boundaries. This was demonstrated in chapter 7 by segmenting the 3D model of a handbag using only the marks created by its manufacturing process.

Additional to the feature detection capabilities, the results of the segmentation process were exported and used in a prototyping platform where its functionality was demonstrated in two common tasks of the product development process: material selection and design of user interfaces. Hence, the suitability of the complete proposed workflow to support the mixed prototyping content authoring process was validated.

RQ2: How do the results of a texture-based mesh segmentation compare in terms of quality to a manual segmentation?

A test including subjects with diverse levels of competence in 3D modelling was conducted to address this research question. The resulting segmentation using the proposed approach and the standard tools of 3D modelling software were evaluated using the manual segmentations of experienced users as a reference. This process was supported by multiple metrics that evaluate differences between segmentations of the same 3D model.

The results show that the subjects tend to generate less accurate segmentations when using the proposed approach than when using the tools of 3D modelling software. Moreover, the segmentations using the texture-based method also presented a higher dispersion of the results when compared to the manual method, while most of the subjects obtained consistently lower errors. Nevertheless, the errors obtained using the proposed method were still low in absolute terms. Considering the *cut discrepancy*, a metric that calculates the average distance between the boundaries of segmentation and a reference, most of the results obtained using the proposed approach had a difference of less than 1 mm. Based on these results, even though the proposed approach does not produce segmentations as accurate as manual segmentation, it could still be suitable for applications where that error level is acceptable, such as projector-based mixed prototyping platforms.

RQ3: Does a content authoring process supported by computer vision improve user acceptance of mixed prototyping?

The evaluation of the overall impact of the proposed approach on the user acceptance of mixed prototyping can be challenging to address due to the multiple variables that must be considered. Nevertheless, by focusing on the preparation process, the influence over some independent parameters could partially support this assessment. Considering this, three different parameters were evaluated during this thesis: The time required to prepare a 3D model to be used in a mixed prototyping platform, the competencies in 3D modelling needed to execute this process, and the perceived ease of use of the tool used.

Regarding the time required to execute the segmentation of a surface reconstructed model to be used in mixed prototyping, the results show that the mean time required to segment the models using the proposed approach was nearly 1/3 of the mean time required to segment the models manually. Indeed, the minimum execution time of using the proposed approach was more than 8 minutes faster than the minimum execution time for the manual segmentation. Furthermore, the maximum execution time using the proposed approach was almost 20 minutes faster than the maximum execution time for the manual segmentation. Nevertheless, using the proposed approach, some subjects executed the manual segmentation faster than the slowest segmentation. Regardless, when considering all the results, the data indicate a clear benefit in reducing execution time when using the proposed approach.

To reduce the competencies in 3D modelling needed to create mixed prototypes, the proposed approach transferred the decision of how to separate the 3D model from a task in a 3D modelling software to a task in the preparation of the physical prototype. The results show that subjects with low expertise in 3D modelling could properly segment 3D models for mixed prototyping without manually intervening in the mesh. Hence, demonstrating the capability of the proposed approach to reduce the 3D modelling competence needed for mixed prototyping content authoring.

Finally, regarding the impact on the perceived ease of use, the results showed that the subjects perceived the proposed method as easier to use than the standard modelling tools to execute the mesh segmentation. Moreover, it was found that subjects with previous experience in 3D modelling perceived the proposed method as easier to use than the users with no previous experience. One potential explanation for this difference could be that the previous experience in 3D modelling software served as a reference point for comparison, favouring the assessment of the proposed method due to its lower complexity.

All these improvements combined indicate that a content authoring process supported by computer vision has the potential to positively influence user acceptance and technological adoption of mixed prototyping (de Souza Cardoso et al., 2020). Although other factors could affect mixed prototyping adoption within the product development process, the proposed approach is a step forward to lower the current adoption barriers.

9.2 Limitations of the research methodology

The main limitation of the research methodology was related to the ability to evaluate the real impact of the texture-based mesh segmentation approach in the adoption of mixed prototyping. Some factors that could affect the adoption of this technology were improved (i. e., the time required to prepare the 3D models, the requirement of 3D modelling competencies, and ease of use of the tools used). However, it is unclear if these improvements would be enough to make a significant change in adoption. Moreover, an effort was made to include the test subjects that closely represented the target users of this technology from the 3D modelling competence standpoint. However, it was not possible to evaluate the impact of the proposed approach on users that already knew how to prepare mixed prototypes and, at the same time, were external to the research group. Hence, this perspective that could have provided valuable insight was missed.

Another limitation of the methodological approach was related to the assessment of some variables of the input models. Multiple 3D models were used during the research activity with various geometries, segmentation requirements, mesh resolution, texture resolutions and surface reconstruction methods. However, some constraints regarding the resolution of the models and texture had to be put in place to maintain reasonable processing times and therefore lacked an evaluation on higher resolutions. Moreover, while two different surface reconstruction methods were used (i.e., structured light scanning and photogrammetry), different software could still generate different results regarding surface quality and generation of the UV map. In particular, while testing, the 3D models were created using Artec Studio, Metashape and Reality Capture, three professional surface reconstruction software. Hence, the quality of the output models and textures from this process was relatively high compared to other devices or methods, and the UV maps did not present high levels of segmentation. Due to this, the results did not provide enough information to assess these variables and therefore is unclear if other difficulties regarding the feature detection or segmentation would arise when using textured models generated through other methods or higher resolutions.

Finally, regarding the preparation process, the objects during testing had previously defined segmentation features and were suitable for surface reconstruction. However, in some cases, the final user will need to ensure these requirements before continuing. Therefore, the impact of this part of the process was not fully assessed.

9.3 Limitations of the texture-based mesh segmentation approach

The main limitation of the texture-based mesh segmentation approach proposed in this thesis is its high dependence on the input information in terms of the texture and the 3D model.

Regarding the input texture, a correct segmentation depends on detecting texture features that create closed boundaries. If those features are incomplete in the physical prototype or there is missing information in the texture after the surface reconstruction process, the detected boundaries will be incomplete. Hence, it will not be possible to detect the independent segments. The dilate iterations parameter was added to face this problem and join gaps of missing information, however, its effectivity is limited to small gaps, and it increases processing time considerably when high values are used. Moreover, the texture resolution can also affect the results, with low resolutions reducing the precision of the feature detection and high resolutions considerably increasing processing times for the feature detection step.

Concerning the input 3D model, a correct definition of the segments depends on the existence of a closed loop of vertices inside the detected features in the texture. Due to this requirement, the approach is unsuitable for the segmentation of low-resolution models, as there will be a higher probability of the loop being broken by vertices outside the detected features. Moreover, this need for high-resolution models as an input carries out other related problems, the clearest being increased processing times. Since the approach only focuses on the segmentation of the 3D model and does not modify the total number of faces or their shape, the segmented output models will be equally high resolution. This characteristic has two implications. First, the segments' boundaries will follow the flow of the existing vertices, which for reconstructed surface models is often highly irregular. Second, the model could require further optimization to be used in real-time applications or systems with limited processing power.

Conversely, regarding the output model and the reversibility of the operations, by applying the proposed mesh segmentation method there are two main changes to take in consideration. On the one hand, the model is segmented, hence multiple 3D objects are created using the information of the input model. Since the relative position and shape of the original polygons remains unchanged, this part of the process can be reverted by merging the duplicated vertices at the boundaries of each of the segments. On the other hand, the resulting segments will include a new UV map. In this case the original UV map used to transfer the selection of the boundary features is completely lost and replaced by the new one. Although this was not addressed during the PhD project, a potential solution to make this part of the process also reversible would be to save the original UV map on an additional UV layer.

Additionally, regarding the usability of the solution, as found out from the testing activities, the effect of the feature detection parameters can be challenging to understand for new users, limiting their ability to choose the ideal parameters. Moreover, even though the approach uses existing features to define the segmentation boundaries, the results could still not match the user expectations, as different persons can have different interpretations of the ideal segmentation.

Finally, although the proposed approach reduces the competence in 3D modelling needed to execute the authoring process, it also adds some requirements for preparing the physical prototype. Consequently, the total preparation time could increase if these requirements are not met by the prototype’s regular manufacturing process or imply modifications. Therefore, it becomes a trade-off to be considered by the final user.

Chapter 10: Conclusion

To overcome one of the adoption barriers to mixed prototyping, this thesis presented a texture-based mesh segmentation approach to support mixed prototyping content authoring. This chapter concludes the thesis by summarizing the key research outcomes concerning the main research objectives. Then is followed by a brief discussion of the research novelty and contributions. Finally, the chapter concludes with recommendations for future research based on observed limitations and potential improvements.

10.1 Concluding remarks

The research began with three main objectives towards the mixed prototyping content authoring process (see section 1.4). A brief outline of the research output referring to these objectives is presented below:

O1: Develop a workflow and supporting tool that can be used to streamline the mixed prototyping content authoring process.

A content authoring workflow was proposed considering the specific requirements of mixed prototyping, spanning from the availability of a physical prototype to creating a 3D model ready to be imported in a mixed prototyping platform. Considering the need for the geometrical similarity between the physical prototype and the 3D model used for the projection of virtual information, as well as the wide variety of existing methods to manufacture the physical prototypes, surface reconstruction methods were selected as the ideal technology for the creation of the 3D model. Moreover, taking advantage of the characteristics of these 3D models and considering the requirements of the input 3D models to be used in mixed prototyping platforms, an algorithm was developed to support this process. The algorithm applied a texture-based mesh segmentation approach based on computer vision that automated the preparation of the 3D model while requiring minimal user input. Moreover, two implementations of the algorithm were created, a standalone application and an add-on for Blender. Although both were usable, due to the limitations imposed by the Blender API and the current knowledge of the author in programming, the processing was considerably slower than the standalone implementation. Hence, the latter is a better alternative to be implemented in a real scenario.

The suitability of the proposed workflow and tool was demonstrated throughout the entire thesis. First, verifying the technical capacity of the tool on synthetic data and then testing the complete workflow with a wide variety of actual physical prototypes with different geometries and types of surface features to be used. Additionally, the quality of the resulting model and the approach's usability was addressed with positive results, validating the applicability of this approach in a realistic setup.

O2: Reduce the 3D modelling knowledge requirements to carry out the mixed prototyping content authoring process.

To reduce the 3D modelling competence requirements, we introduced an algorithm for mesh segmentation that uses the texture information provided by reconstructed surface models to conduct this process automatically. Based on this new approach, it was possible to eliminate the need for users to manually segment the 3D model, transferring part of this process to the physical world. Instead of defining the segments directly in the 3D model, the proposed approach requires the user to define the segment's boundaries in the physical prototype with high contrast marks. Once this has been ensured a textured model generated must be generated through surface reconstruction methods. Then, the user only needs to select the correct feature detection parameters and the desired number of segments in the output model. This approach's effectiveness in reducing 3D modelling knowledge was demonstrated in the tests presented in chapter 7, where the subjects did not have prior competence in 3D modelling but managed to segment the 3D models properly. Moreover, as shown in the test carried out in chapter 8, even for users with experience in 3D modelling, there was a significant improvement in the perceived ease of use compared to the standard tools of 3D modelling software. Hence, offering benefits to both inexperienced and experienced users in 3D modelling.

O3: Achieve a segmentation of similar quality to one made manually by an experienced user.

The results obtained with the texture-based mesh segmentation were compared against reference segmentations provided by experienced users of 3D modelling software to assess their quality. The results showed that the segmentation results of the proposed approach provided a low amount of error (i.e., an average of 1 to 3 millimetre difference in the boundaries of the segmentations). Analogous to misalignments in projection, several factors could affect the limit of acceptance of segmentation errors on projector-based augmented reality systems. Some of these factors include the overall size of the object, the geometrical complexity of the boundaries and the distance of the viewer. Moreover, the limit of acceptance is a matter that could vary between different users. Due to these uncertainties, another sources of error or misalignments could be used as a reference point to define a limit of acceptance for the obtained segmentations. Such sources include the limitations of the projector's rendering resolution, and the micromovements of the projection due to the constant updates of the tracking system. Both can introduce an alignment error around 1 to 5 millimetres, a value of similar magnitude than the one introduced by proposed mesh segmentation approach. Hence, the latter can be acceptable in the context of mixed prototyping.

Additionally, in some cases, the texture features of the objects and the used parameters generated unwanted segments that did not correlate with the subjects' expectations. Nevertheless, due to the order in which the resulting segments were defined, these unwanted segments affected only a small area of the total mesh.

10.2 Research novelty and contributions

This thesis aimed to contribute to the technological adoption of mixed prototyping by reducing the knowledge requirements of the current content authoring process. The research addressed the gaps related to the segmentation and UV mapping of 3D models, as it is time-consuming and requires specific 3D modelling skills to execute it. The literature analysis revealed the limitation of existing authoring methods to support this process. Moreover, it was found that existing mesh segmentation methods were unsuitable for mixed prototyping as they did not match the designer segmentation needs. Nevertheless, image segmentation was a promising alternative to support this process. The analysis concluded that in the context of mixed prototyping, where surface reconstruction methods are used to digitalize physical prototypes, the texture of 3D models is a valuable input that can be used to automate the segmentation process.

The novelty of the research conducted is based on developing a texture-based mesh segmentation approach to reduce the knowledge requirements during the content authoring of mixed prototypes. The main novelties and contributions are listed below:

- This research defines a processing algorithm and workflow to exploit surface characteristics from textured 3D models of physical prototypes to segment them following the specific needs of mixed prototyping.
- This research contributes to the reduction in 3D modelling knowledge requirement to segment 3D models. This process, typically carried out through the standard tools of 3D modelling software by experienced users, is replaced by simpler process that require the user only to choose the right parameters for feature detection and the number of segments wanted as an output.
- This research contributes to the literature on geometry processing of surface reconstructed models by taking advantage of the information provided by textures and the specific characteristics (i.e., high number of polygons) of such models to conduct an automated part-based segmentation.
- This research evaluates the impact of a texture-based mesh segmentation approach from a segmentation quality and user perspective, contributing to a deeper understanding of the potential that one solution of this type can have and the limitations of that must be considered.

10.3 Recommendations for further research

The texture-based mesh segmentation approach provides a viable alternative to support the preparation of 3D models for mixed prototyping when textured surface reconstructed models are available. However, while the presented thesis shows positive results, a continuation of the research could further increase the impact of the approach from multiple perspectives.

As presented in this thesis, the proposed approach was validated using different physical prototypes. While this provided helpful information to evaluate its suitability for mixed prototyping, it is unclear how these results would translate to other contexts. Hence, the approach’s capability to segment textured models from other contexts should be explored, for example, by testing it on existing textured 3D model datasets like the one proposed by Maggiordomo et al. (2020). This will not only provide more evidence of the capability and limitations of the proposed approach but also bring some insight into which areas, aside from mixed prototyping, could benefit from this segmentation approach.

An activity in this direction was an exploratory application of the proposed approach within a company working on product digitalization. While the processed objects did not match the surface features of mixed prototypes, the material changes still provided enough information to define segmentation boundaries in some cases. Although the results were not usable for the company in the current state, changes in the underlying compute vision algorithms could improve the results. Moreover, through an interview with the company, a potential algorithm variation focused on easing the transference of textures on reconstructed surface models to optimized models was also brought to attention.

Additionally, when evaluating the capability of users to select feature detection parameters that correctly categorized the boundary features, it was noted that a wide range of parameter combinations generated successful results. Hence, new research could also focus on automating the parameter selection for the feature detection step. Similarly, the definition of the number of output segments could also be explored. These changes could further reduce the user's knowledge requirements and speed up the segmentation process by eliminating the need for user input.

Another path to extend the research conducted is the integration of new computer vision algorithms. The use of an adaptive threshold combined with the dilate iterations demonstrated to be suitable for segmenting objects where high contrast features marked the boundaries of the segments to be separated. Nevertheless, other computer vision algorithms could use different texture features, such as detecting unique materials or repeating patterns in some segments. This expansion in the features that can be used as input for the segmentations could facilitate the use of the proposed approach on new applications.

Finally, considering that the original objective of the presented work was to improve the technological adoption of mixed prototyping, if the proposed approach's impact is not enough to overcome this barrier, the effort could be put into improving other parts of the process. In particular, the calibration of the tracking and projector system is a step that, although unrelated to content creation, suffers from similar problems. Its complexity requires users with high expertise in the process to execute it properly. Hence, it creates a dependence on the developers when inexperienced users want to install or modify the setup, which can become a barrier to adoption.

Appendix A: Mesh segmentation metrics

This definitions were extracted from the work of Chen et al. (2009), which also provided a benchmarking tool to calculate these metrics when an automatic and a ground truth segmentations are provided. For all of them a lower number, in a scale from 0 to 1, the better is the segmentation result.

A.1 Cut discrepancy

Assuming C_1 and C_2 are sets of all point on the segment boundaries of segmentations S_1 and S_2, respectively, and $d_G(p_1,p_2)$ measures the geodesic distance between two points on a mesh, then the geodesic distance from a point $p_1 \in C_1$ to a set of cuts C_2 is defined as:

$$d_G(p_1, C_2) = min\{d_G(p_1,p_2), \forall p_2 \in C_2\}$$

and the Directional Cut Discrepancy, $DCD(S_1 \Rightarrow S_2)$, of S_1 with respect to S_2 is defined as the mean of the distribution of $d_G(p_1, C_2)$ for all points $p_1 \in C_1$:

$$DCD(S_1 \Rightarrow S_2) = mean\{d_G(p_1, C_2), \forall p_1 \in C_1\}$$

We define the Cut Discrepancy, $CD(S_1,S_2)$, to be the mean of the directional functions in both directions, divided by the average Euclidean distance from a point on the surface to the centroid of the mesh $(avgRadius)$ to ensure symmetry of the metric and to avoid effects due to the scale:

$$CD(S_1,S_2) = \frac{DCD(S_1 \Rightarrow S_2) + DCD(S_2 \Rightarrow S_1)}{avgRadius}$$

A.2 Hamming distance

Given two mesh segmentation $S_1 = \{S_1^1, S_1^1, \ldots, S_1^m\}$ and $S_2 = \{S_2^1, S_2^1, \ldots, S_2^n\}$ with m and n segments, respectively, the Directional Hamming Distance is defined as

$$D_H(S_1 \Rightarrow S_2) = \sum_i \|S_2^i \setminus S_1^{i_t}\|$$

Where "\" is the set difference operator, $\|x\|$ is a measure for set x (e.g., the size of set x, or the total area of all faces in a face set), and $i_t = max_k \|S_2^i \cap S_1^k\|$. The general idea is to find a best corresponding segment in S_1 for each segment in S_2, and sum up the set difference.

If S_2 is regarded as the ground truth, then the Directional Hamming distance can be used to define the missing rate R_m and false alarm rate R_f as follows:

$$R_m(S_1,S_2) = \frac{D_H(S_1 \Rightarrow S_2)}{\|S\|}$$

$$R_f(S_1,S_2) = \frac{D_H(S_2 \Rightarrow S_1)}{\|S\|}$$

where $\|S\|$ is the total surface area of the polygonal model. The Hamming distance is simply defined as the average of the missing rate and false alarm rate:

$$HD(S_1,S_2) = \frac{1}{2}(R_m(S_1,S_2)\ +\ R_f(S_1,S_2))$$

A.3 Rand index

If we denote S_1 and S_2 as two segmentations, s_i^1 and s_i^2 as the segment IDs of face i in S_1 and S_2, and N as the number of faces in the polygonal mesh, $C_{ij}\ =\ 1\ iff\ \ s_i^1\ =\ s_j^1$, and $P_{ij}\ =\ 1\ iff\ \ s_i^2\ =\ s_j^2$, then we can define Rand Index as:

$$RI(S_1,S_2) = \binom{2}{n}^{-1} \sum_{i,j,i<j} \left[C_{ij}P_{ij} +\ (1\ -\ C_{ij})(1\ -\ P_{ij})\right]$$

$C_{ij}P_{ij}\ =\ 1$ indicates that face i and j have the same ID in both S_1 and S_2. $(1\ -\ C_{ij})(1\ -\ P_{ij})\ =\ 1$ indicates that face i and j have different IDs in both S_1 and S_2. Thus $RI(S_1,S_2)$ tells the proportion of face pairs that agree or disagree jointly on their segment group identities in segmentations S_1 and S_2.

As a slight departure from the standard practice, the benchmark report $1 -\ RI(S_1,S_2)$ to be consistent with the other metrics that report dissimilarities rather than similarities.

A.4 Consistency error

Denoting S_1 and S_2 as two segmentation results for a model, t_i as a mesh face, "\" as the set difference operator, and $\|x\|$ as a measure for set x (as in section A.2), $R(S,f_i)$ as the segment (a set of connected faces) in segmentation S that contain face f_i, and n as the number of faces in the polygonal model, the local refinement error is defined as:

$$E(S_1,S_2,f_i) = \frac{\|R(S_1,f_i) \setminus R(S_2,f_i)\|}{\|R(S_1,f_i)\|}$$

Given the refinement error for each face, two metrics are defined for the entire 3D mesh, Global Consistency Error (GCE) and Local Consistency Error (LCE), as follows.

$$GCE(S_1,S_2) = \frac{1}{n} min\left\{\sum_i E(S_1,S_2,f_i)\ ,\sum_i E(S_2,S_1,f_i)\right\}$$

$$LCE(S_1,S_2) = \frac{1}{n}\sum_i min\{E(S_1,S_2,f_i),E(S_2,S_1,f_i)\}$$

Both GCE and LCE are symmetric. The difference between them is that GCE forces all local refinements to be in the same direction, while LCE allows refinement in different directions in different parts of the 3D model. As a result, $GCE(S_1,S_2)\ \geq\ LCE(S_1,S_2)$.

Appendix B: Technology Acceptance Model survey

This survey was adapted from the original Technology Acceptance Model survey from Davis (Davis, 1989). It consists of 12 sentences to be evaluated using a Likert scale of agreement. In the context of the research, the second part, related to the perceived ease of use was duplicated to compare the proposed approach (B.2) to a mesh segmentation manually done using the existing tools of a 3D modelling software (B.3).

B.1 Perceived usefulness of SPARK

1. Using SPARK in my job would enable me to accomplish tasks more quickly
2. Using SPARK would improve my job performance
3. Using SPARK would increase my productivity
4. Using SPARK would enhance my effectiveness on the job
5. Using SPARK would make it easier to do my job
6. I would find SPARK useful in my job

B.2 Perceived ease of use of the developed tool

7. Learning to use the tool/add-on was easy for me
8. I found easy to get the tool/add-on to do what I wanted to do
9. My interaction with the tool/add-on was clear and understandable
10. I found the tool/add-on to be flexible to interact with
11. It would be easy to become skilful at using the tool/add-on
12. I found the tool/add-on easy to use

B.3 Perceived ease of use of the 3D modelling software

7. Learning to use the 3D modelling software was easy for me
8. I found easy to get the 3D modelling software to do what I wanted to do
9. My interaction with the 3D modelling software was clear and understandable
10. I found the 3D modelling software to be flexible to interact with
11. It would be easy to become skilful at using the 3D modelling software
12. I found the 3D modelling software easy to use

Appendix C: Associated publications

Piñones, E., Cascini, G., Caruso, G., & Morosi, F. (2021). **Overcoming Augmented Reality adoption barriers in design: A mixed prototyping content authoring tool supported by computer vision.** Proceedings of the Design Society 2021 ; 1 : 2359–68. https://doi.org/10.1017/pds.2021.497.

Abstract:

Enhancing the appearance of physical prototypes with digital elements, also known as mixed prototyping, has demonstrated to be a valuable approach in the product development process. However, the adoption is limited also due to the high time and competence required for authoring the digital contents. This paper presents a content authoring tool that aims to improve the user acceptance by reducing the specific competence required, which is needed for segmentation and UV mapping of the 3D model used to implement a mixed prototype. Part of the tasks related to 3D modelling software, in fact, has been transferred to simpler manual tasks applied onto the physical prototype. Moreover, the proposed tool can recognise these manual inputs thanks to a computer-vision algorithm and automatically manage the segmentation and UV mapping tasks, freeing time for the user in a task that otherwise would require complete engagement. To preliminarily evaluate effectiveness and potential of the tool, it has been used in a case study to build up the mixed prototype of a coffee machine. The result demonstrated that the tool can correctly segment the 3D model of a physical prototype in its relevant parts and generate their corresponding UV maps.

Piñones, E., Cascini, G., Morosi, F., & Caruso, G. (2022). **Texture-based mesh segmentation for Mixed Reality content authoring: Exploiting surface features in physical prototypes.** Virtual Reality. Submitted on December 16, 2022.

Abstract:

The use of mixed reality in the product prototyping process has demonstrated great potential to reduce costs and development time. However, its complex authoring process has hindered the adoption of this technology. This paper aims to evaluate the potential of a texture-based segmentation approach to properly segment a 3D model for mixed prototyping using added and already existing surface features. Additionally assessing the capability of users with low expertise in 3D modelling to carry out this process and the quality of the results in relation to manual segmentation. The results indicate that the proposed approach can successfully detect features in a wide range of physical prototypes to aid the segmentation of 3D models for mixed prototyping and other processes that require a part-based segmentation of reconstructed surface models. Moreover, subjects with little knowledge of 3D modelling generated a segmented 3D model that closely followed a ground truth segmentation without manually intervening in the mesh, demonstrating the capability of the approach in terms of reducing knowledge requirements for carrying out the mesh segmentation process needed for mixed prototyping in 3D modelling.

References

Agisoft. (2022). *Agisoft Metashape: Agisoft Metashape*. https://www.agisoft.com/

Akaoka, E., Ginn, T., & Vertegaal, R. (2010). DisplayObjects: Prototyping Functional Physical Interfaces on 3D Styrofoam, Paper or Cardboard Models. *Proceedings of the Fourth International Conference on Tangible, Embedded, and Embodied Interaction*, 49–56. https://doi.org/10.1145/1709886.1709897

Anabuki, M., & Ishii, H. (2007). AR-Jig: A Handheld Tangible User Interface for Modification of 3D Digital Form via 2D Physical Curve. *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, 55–66. https://doi.org/10.1109/ISMAR.2007.4538826

Arbeláez, P., Maire, M., Fowlkes, C., & Malik, J. (2011). Contour Detection and Hierarchical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(5), 898–916. https://doi.org/10.1109/TPAMI.2010.161

Arroyave-Tobón, S., Osorio-Gómez, G., & Cardona-McCormick, J. F. (2015). AIR-MODELLING: A tool for gesture-based solid modelling in context during early design stages in AR environments. *Computers in Industry*, *66*, 73–81. https://doi.org/10.1016/j.compind.2014.10.007

Artec3D. (2022a). *Best 3D Scanning Software | Artec Studio*. Professional 3D Scanning Solutions | Artec 3D. https://www.artec3d.com/3d-software/artec-studio

Artec3D. (2022b). *Wireless Handheld 3D Scanner | Artec Leo | Best 3D Scanning Solutions*. Professional 3D Scanning Solutions | Artec 3D. https://www.artec3d.com/portable-3d-scanners/artec-leo

Badampudi, D., Wohlin, C., & Petersen, K. (2015). Experiences from using snowballing and database searches in systematic literature studies. *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*, 1–10. https://doi.org/10.1145/2745802.2745818

Barbieri, L., Angilica, A., Bruno, F., & Muzzupappa, M. (2013). Mixed prototyping with configurable physical archetype for usability evaluation of product interfaces. *Computers in Industry*, *64*(3), 310–323. https://doi.org/10.1016/j.compind.2012.11.010

Barone Rodrigues, A., Dias, D. R. C., Martins, V. F., Bressan, P. A., & de Paiva Guimarães, M. (2017). WebAR: A Web-Augmented Reality-Based Authoring Tool with Experience API Support for Educational Applications. In M. Antona & C. Stephanidis (Eds.), *Universal Access in Human–Computer Interaction. Designing Novel Interactions* (pp. 118–128). Springer International Publishing. https://doi.org/10.1007/978-3-319-58703-5_9

Bauer, M., Bruegge, B., Klinker, G., MacWilliams, A., Reicher, T., Riss, S., Sandor, C., & Wagner, M. (2001). Design of a component-based augmented reality framework. *Proceedings IEEE and ACM International Symposium on Augmented Reality*, 45–54. https://doi.org/10.1109/ISAR.2001.970514

Becattini, N., Cascini, G., O’Hare, J. A., Morosi, F., & Boujut, J.-F. (2019). Extracting and Analysing Design Process Data from Log Files of ICT Supported Co-Creative Sessions. *Proceedings of the Design Society: International Conference on Engineering Design*, *1*(1), 129–138. https://doi.org/10.1017/dsi.2019.16

Bégout, P., Duval, T., Kubicki, S., Charbonnier, B., & Bricard, E. (2020). WAAT: A Workstation AR Authoring Tool for Industry 4.0. In L. T. De Paolis & P. Bourdot (Eds.), *Augmented Reality, Virtual Reality, and Computer Graphics* (pp. 304–320). Springer International Publishing. https://doi.org/10.1007/978-3-030-58468-9_22

Benhabiles, H., Vandeborre, J.-P., Lavoue, G., & Daoudi, M. (2009). A framework for the objective evaluation of segmentation algorithms using a ground-truth of human segmented 3D-models. *2009 IEEE International Conference on Shape Modeling and Applications*, 36–43. https://doi.org/10.1109/SMI.2009.5170161

References

Bergig, O., Hagbi, N., El-Sana, J., & Billinghurst, M. (2009). In-place 3D sketching for authoring and augmenting mechanical systems. *Proceedings of the 2009 8th IEEE International Symposium on Mixed and Augmented Reality*, 87–94. https://doi.org/10.1109/ISMAR.2009.5336490

Bhattacharya, B., & Winer, E. H. (2019). Augmented reality via expert demonstration authoring (AREDA). *Computers in Industry*, *105*, 61–79. https://doi.org/10.1016/j.compind.2018.04.021

Billinghurst, M., Kato, H., & Poupyrev, I. (2008). *Tangible Augmented Reality*. *7(2)*, 1–10.

Bimber, O., Coriand, F., Kleppe, A., Bruns, E., Zollmann, S., & Langlotz, T. (2005). Superimposing pictorial artwork with projected imagery. *IEEE MultiMedia*, *12*(1), 16–26. https://doi.org/10.1109/MMUL.2005.9

Bimber, O., & Raskar, R. (2005). *Spatial Augmented Reality: Merging Real and Virtual Worlds*. A K Peters/CRC Press. https://doi.org/10.1201/b10624

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Camburn, B., Viswanathan, V., Linsey, J., Anderson, D., Jensen, D., Crawford, R., Otto, K., & Wood, K. (2017). Design prototyping methods: State of the art in strategies, techniques, and guidelines. *Design Science*, *3*. https://doi.org/10.1017/dsj.2017.10

Cao, Y., Wang, T., Qian, X., Rao, P. S., Wadhawan, M., Huo, K., & Ramani, K. (2019). GhostAR: A Time-space Editor for Embodied Authoring of Human-Robot Collaborative Task with Augmented Reality. *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 521–534. https://doi.org/10.1145/3332165.3347902

Capturing Reality. (2022). *RealityCapture—Photogrammetry Software*. https://www.capturingreality.com/realitycapture

Carulli, M., Bordegoni, M., & Cugini, U. (2013). An approach for capturing the Voice of the Customer based on Virtual Prototyping. *Journal of Intelligent Manufacturing*, *24*(5), 887–903. https://doi.org/10.1007/s10845-012-0662-5

Cascini, G., O'Hare, J., Dekoninck, E., Becattini, N., Boujut, J.-F., Ben Guefrache, F., Carli, I., Caruso, G., Giunta, L., & Morosi, F. (2020). Exploring the use of AR technology for co-creative product and packaging design. *Computers in Industry*, *123*, 103308. https://doi.org/10.1016/j.compind.2020.103308

Castillo, R. I. B., Sánchez, V. G. C., Villegas, O. O. V., & Lozoya, A. L. (2016). Node based visual editor for mobile augmented reality. *International Journal of Combinatorial Optimization Problems and Informatics*, *7*(2), 35–48.

Cavallo, M., & Forbes, A. G. (2019). CAVE-AR: A VR Authoring System to Interactively Design, Simulate, and Debug Multi-user AR Experiences. *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 872–873. https://doi.org/10.1109/VR.2019.8798148

Chen, Q., Chen, H.-M., & Kazman, R. (2007). Investigating antecedents of technology acceptance of initial eCRM users beyond generation X and the role of self-construal. *Electronic Commerce Research*, *7*(3), 315–339. https://doi.org/10.1007/s10660-007-9009-2

Chen, X., Golovinskiy, A., & Funkhouser, T. (2009). A Benchmark for 3D Mesh Segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, *28*(3). https://doi.org/10.1145/1531326.1531379

Chen, Z., Su, Y., Wang, Y., Wang, Q., Qu, H., & Wu, Y. (2020). MARVisT: Authoring Glyph-Based Visualization in Mobile Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics*, *26*(8), 2645–2658. https://doi.org/10.1109/TVCG.2019.2892415

References

Cherry, E., & Latulipe, C. (2014). Quantifying the Creativity Support of Digital Tools through the Creativity Support Index. *ACM Trans. Comput.-Hum. Interact.*, *21*(4). https://doi.org/10.1145/2617588

Chidambaram, S., Huang, H., He, F., Qian, X., Villanueva, A. M., Redick, T. S., Stuerzlinger, W., & Ramani, K. (2021). ProcessAR: An augmented reality-based tool to create in-situ procedural 2D/3D AR Instructions. *Designing Interactive Systems Conference 2021*, 234–249. https://doi.org/10.1145/3461778.3462126

Christie, E. J., Jensen, D. D., Buckley, R. T., Menefee, D. A., Ziegler, K. K., Wood, K. L., & Crawford, R. H. (2012). *Prototyping Strategies: Literature Review and Identification of Critical Variables*. 25.1091.1-25.1091.22. https://peer.asee.org/prototyping-strategies-literature-review-and-identification-of-critical-variables

Coquillart, S., Göbel, M., & others. (2004). Authoring of mixed reality applications including multi-marker calibration for mobile devices. *Eurographics Symposium on Virtual Environments (2004)*, 1–9.

Cubillo, J., Martin, S., Castro, M., & Boticki, I. (2015). Preparing augmented reality learning content should be easy: UNED ARLE—an authoring tool for augmented reality learning environments. *Computer Applications in Engineering Education*, *23*(5), 778–789. https://doi.org/10.1002/cae.21650

Davila, J. M., Oyedele, L., Demian, P., & Beach, T. (2020). A research agenda for augmented and virtual reality in architecture, engineering and construction. *Advanced Engineering Informatics*, *45*, 101122. https://doi.org/10.1016/j.aei.2020.101122

Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, *13*(3), 319–340. https://doi.org/10.2307/249008

de Souza Cardoso, L. F., Mariano, F. C. M. Q., & Zorzal, E. R. (2020). A survey of industrial augmented reality. *Computers & Industrial Engineering*, *139*, 106159. https://doi.org/10.1016/j.cie.2019.106159

Dörner, R., Geiger, C., Haller, M., & Paelke, V. (2003). Authoring Mixed Reality—A Component and Framework-Based Approach. In R. Nakatsu & J. Hoshino (Eds.), *Entertainment Computing: Technologies and Application* (pp. 405–413). Springer US. https://doi.org/10.1007/978-0-387-35660-0_49

Egger, J., & Masood, T. (2020). Augmented reality in support of intelligent manufacturing – A systematic literature review. *Computers & Industrial Engineering*, *140*, 106–195. https://doi.org/10.1016/j.cie.2019.106195

Ehnes, J., Hirota, K., & Hirose, M. (2004). Projected augmentation—Augmented reality using rotatable video projectors. *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, 26–35. https://doi.org/10.1109/ISMAR.2004.47

Felizardo, K. R., Mendes, E., Kalinowski, M., Souza, É. F., & Vijaykumar, N. L. (2016). Using Forward Snowballing to update Systematic Reviews in Software Engineering. *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 1–6. https://doi.org/10.1145/2961111.2962630

Feng, L., Yang, X., & Xiao, S. (2017). MagicToon: A 2D-to-3D creative cartoon modeling system with mobile AR. *2017 IEEE Virtual Reality (VR)*, 195–204. https://doi.org/10.1109/VR.2017.7892247

Fujimoto, Y., Yamamoto, G., Taketomi, T., Sandor, C., & Kato, H. (2015). [POSTER] Pseudo Printed Fabrics through Projection Mapping. *2015 IEEE International Symposium on Mixed and Augmented Reality*, 174–175. https://doi.org/10.1109/ISMAR.2015.51

Gattullo, M., Uva, A. E., Fiorentino, M., Scurati, G. W., & Ferrise, F. (2017). From Paper Manual to AR Manual: Do We Still Need Text? *Procedia Manufacturing*, *11*, 1303–1310. https://doi.org/10.1016/j.promfg.2017.07.258

Gebler, O. F., Goudswaard, M., Hicks, B., Jones, D., Nassehi, A., Snider, C., & Yon, J. (2021). A comparison of structured light scanning and photogrammetry for digitisation of physical prototypes. *Proceedings of the Design Society*, *1*, 11–20. https://doi.org/10.1017/pds.2021.2

References

George, D., Xie, X., & Tam, G. K. (2018). 3D mesh segmentation via multi-branch 1D convolutional neural networks. *Graphical Models*, *96*, 1–10. https://doi.org/10.1016/j.gmod.2018.01.001

Ghosh, C., Majumder, S., Ray, S., Datta, S., & Mandal, S. N. (2020). Different EDGE Detection Techniques: A Review. In P. K. Mallick, P. Meher, A. Majumder, & S. K. Das (Eds.), *Electronic Systems and Intelligent Computing* (pp. 885–898). Springer. https://doi.org/10.1007/978-981-15-7031-5_84

Gimeno, J., Morillo, P., Orduña, J. M., & Fernández, M. (2013a). An Easy-to-Use AR Authoring Tool for Industrial Applications. In G. Csurka, M. Kraus, R. S. Laramee, P. Richard, & J. Braz (Eds.), *Computer Vision, Imaging and Computer Graphics. Theory and Application* (pp. 17–32). Springer. https://doi.org/10.1007/978-3-642-38241-3_2

Gimeno, J., Morillo, P., Orduña, J. M., & Fernández, M. (2013b). A new AR authoring tool using depth maps for industrial procedures. *Computers in Industry*, *64*(9), 1263–1271. https://doi.org/10.1016/j.compind.2013.06.012

Giunta, L., Guefrache, F. B., Dekoninck, E., Gopsill, J., O'Hare, J., & Morosi, F. (2019). Investigating the Impact of Spatial Augmented Reality on Communication between Design Session Participants—A Pilot Study. *Proceedings of the Design Society: International Conference on Engineering Design*, *1*(1), 1973–1982. https://doi.org/10.1017/dsi.2019.203

Giunta, L., O'Hare, J., Gopsill, J., & Dekoninck, E. (2018). *A Review of Augmented Reality Research for Design Practice: Looking to the Future*. DS 91: Proceedings of NordDesign 2018, Linköping, Sweden, 14th - 17th August 2018.

Glenn, T., Ipsita, A., Carithers, C., Peppler, K., & Ramani, K. (2020). StoryMakAR: Bringing Stories to Life With An Augmented Reality & Physical Prototyping Toolkit for Youth. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–14. https://doi.org/10.1145/3313831.3376790

Grimm, P., Haller, M., Paelke, V., Reinhold, S., Reimann, C., & Zauner, R. (2002). AMIRE - authoring mixed reality. *The First IEEE International Workshop Agumented Reality Toolkit,* 2 pp.-. https://doi.org/10.1109/ART.2002.1107008

Guo, Y., Liu, Y., Georgiou, T., & Lew, M. S. (2018). A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval*, *7*(2), 87–93. https://doi.org/10.1007/s13735-017-0141-z

Ha, T., Billinghurst, M., & Woo, W. (2012). An interactive 3D movement path manipulation method in an augmented reality environment☆. *Interacting with Computers*, *24*(1), 10–24. https://doi.org/10.1016/j.intcom.2011.06.006

Ha, T., & Woo, W. (2007). Graphical tangible user interface for a ar authoring tool in product design environment. *International Symposium on Ubiquitous VR*, 1.

Ha, T., Woo, W., Lee, Y., Lee, J., Ryu, J., Choi, H., & Lee, K. (2010). ARtalet: Tangible User Interface Based Immersive Augmented Reality Authoring Tool for Digilog Book. *2010 International Symposium on Ubiquitous Virtual Reality*, 40–43. https://doi.org/10.1109/ISUVR.2010.20

Hagbi, N., Grasset, R., Bergig, O., Billinghurst, M., & El-Sana, J. (2010). In-Place Sketching for content authoring in Augmented Reality games. *2010 IEEE Virtual Reality Conference (VR)*, 91–94. https://doi.org/10.1109/VR.2010.5444806

Hampshire, A., Seichter, H., Grasset, R., & Billinghurst, M. (2006). Augmented reality authoring: Generic context from programmer to designer. *Proceedings of the 18th Australia Conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments*, 409–412. https://doi.org/10.1145/1228175.1228259

Han, S., Wang, S., & Wang, P. (2020). An Augmented Reality Approach to 3D Solid Modeling and Demonstration. In C. Stephanidis, J. Y. C. Chen, & G. Fragomeni (Eds.), *HCI International 2020 – Late Breaking Papers: Virtual and*

Augmented Reality (pp. 121–140). Springer International Publishing. https://doi.org/10.1007/978-3-030-59990-4_11

Haralick, R. M., & Shapiro, L. G. (1992). *Computer and robot vision*. Addison-Wesley Pub. Co.

Harazono, Y., Tamura, T., Omoto, Y., Ishii, H., Shimoda, H., Tanaka, Y., & Takahashi, Y. (2022). Development of an Augmented Reality Training Environment Construction System Using Puzzle-Like Programming in a Real Environment. *International Journal of Human–Computer Interaction*, *0*(0), 1–14. https://doi.org/10.1080/10447318.2022.2116529

Hoffman, H. G., Hollander, A., Schroder, K., Rousseau, S., & Furness, T. (1998). Physically touching and tasting virtual objects enhances the realism of virtual experiences. *Virtual Reality*, *3*(4), 226–234. https://doi.org/10.1007/BF01408703

Huang, D., Jin, X., & Coghlan, A. (2021). Advances in consumer innovation resistance research: A review and research agenda. *Technological Forecasting and Social Change*, *166*, 120594. https://doi.org/10.1016/j.techfore.2021.120594

Huang, Q., & Dom, B. (1995). Quantitative methods of evaluating image segmentation. *Proceedings., International Conference on Image Processing*, *3*, 53–56 vol.3. https://doi.org/10.1109/ICIP.1995.537578

Huang, T.-L., & Liao, S. (2015). A model of acceptance of augmented-reality interactive technology: The moderating role of cognitive innovativeness. *Electronic Commerce Research*, *15*(2), 269–295. https://doi.org/10.1007/s10660-014-9163-2

Huo, K., Vinayak, & Ramani, K. (2016). Window-Shaping: 3D Design Ideation in Mixed Reality. *Proceedings of the 2016 Symposium on Spatial User Interaction*, 189. https://doi.org/10.1145/2983310.2989189

Ifrim, A.-C., Moldoveanu, F., Moldoveanu, A., & Grădinaru, A. (2021). LibrARy – Enriching the Cultural Physical Spaces with Collaborative AR Content. In J. Y. C. Chen & G. Fragomeni (Eds.), *Virtual, Augmented and Mixed Reality* (pp. 626–638). Springer International Publishing. https://doi.org/10.1007/978-3-030-77599-5_43

Inzerillo, L., Di Paola, F., & Alogna, Y. (2019). High quality texture mapping process aimed at the optimization of 3D structured light models. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLII-2-W9*, 389–396. https://doi.org/10.5194/isprs-archives-XLII-2-W9-389-2019

Isaksson, O., Eckert, C., Panarotto, M., & Malmqvist, J. (2020). YOU NEED TO FOCUS TO VALIDATE. *Proceedings of the Design Society: DESIGN Conference*, *1*, 31–40. https://doi.org/10.1017/dsd.2020.116

Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., & Fitzgibbon, A. (2011). KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera. *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, 559–568. https://doi.org/10.1145/2047196.2047270

Jacobson, A., Panozzo, D., & others. (2018). *libigl: A simple C++ geometry processing library*.

Jam, J., Kendrick, C., Walker, K., Drouard, V., Hsu, J. G.-S., & Yap, M. H. (2021). A comprehensive review of past and present image inpainting methods. *Computer Vision and Image Understanding*, *203*, 103147. https://doi.org/10.1016/j.cviu.2020.103147

Jang, J., Ko, Y., Shin, W. S., & Han, I. (2021). Augmented Reality and Virtual Reality for Learning: An Examination Using an Extended Technology Acceptance Model. *IEEE Access*, *9*, 6798–6809. https://doi.org/10.1109/ACCESS.2020.3048708

Jee, H.-K., Lim, S., Youn, J., & Lee, J. (2011). An Immersive Authoring Tool for Augmented Reality-Based E-Learning Applications. *2011 International Conference on Information Science and Applications*, 1–5. https://doi.org/10.1109/ICISA.2011.5772399

References

Jimeno-Morenilla, A., Sánchez-Romero, J. L., & Salas-Pérez, F. (2013). Augmented and Virtual Reality techniques for footwear. *Computers in Industry*, *64*(9), 1371–1382. https://doi.org/10.1016/j.compind.2013.06.008

Jou, M., & Wang, J. (2013). Observations of achievement and motivation in using cloud computing driven CAD: Comparison of college students with high school and vocational high school backgrounds. *Computers in Human Behavior*, *29*(2), 364–369. https://doi.org/10.1016/j.chb.2012.08.001

Jung, J., Hong, J., Park, S., & Yang, H. S. (2012). Smartphone as an augmented reality authoring tool via multi-touch based 3D interaction method. *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, 17–20. https://doi.org/10.1145/2407516.2407520

Kang, N., Sah, Y. J., & Lee, S. (2021). Effects of visual and auditory cues on haptic illusions for active and passive touches in mixed reality. *International Journal of Human-Computer Studies*, *150*, 102613. https://doi.org/10.1016/j.ijhcs.2021.102613

Katz, S., & Tal, A. (2003). Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Transactions on Graphics (TOG)*. https://dl.acm.org/doi/abs/10.1145/882262.882369

Kent, L., Snider, C., Gopsill, J., & Hicks, B. (2021). Mixed reality in design prototyping: A systematic review. *Design Studies*, *77*, 101046. https://doi.org/10.1016/j.destud.2021.101046

Knopfle, C., Weidenhausen, J., Chauvigne, L., & Stock, I. (2005). Template based authoring for AR based service scenarios. *IEEE Proceedings. VR 2005. Virtual Reality, 2005.*, 237–240. https://doi.org/10.1109/VR.2005.1492779

Koleva, B., Egglestone, S. R., Schnädelbach, H., Glover, K., Greenhalgh, C., Rodden, T., & Dade-Robertson, M. (2009). Supporting the creation of hybrid museum experiences. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1973–1982. https://doi.org/10.1145/1518701.1519001

Kong, J., Sabha, D., Bigham, J. P., Pavel, A., & Guo, A. (2021). TutorialLens: Authoring Interactive Augmented Reality Tutorials Through Narration and Demonstration. *Symposium on Spatial User Interaction*. https://doi.org/10.1145/3485279.3485289

Krieter, P., & Breiter, A. (2018). Analyzing mobile application usage: Generating log files from mobile screen recordings. *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*, 1–10. https://doi.org/10.1145/3229434.3229450

Kruijff, E., Swan, E., & Feiner, S. (2010). *Perceptual issues in augmented reality revisited*. 3–12. https://doi.org/10.1109/ISMAR.2010.5643530

Kwan, K. C., & Fu, H. (2019). Mobi3DSketch: 3D Sketching in Mobile AR. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–11. https://doi.org/10.1145/3290605.3300406

Ladwig, P., & Geiger, C. (2018). A Literature Review on Collaboration in Mixed Reality. *REV*.

Langlotz, T., Mooslechner, S., Zollmann, S., Degendorfer, C., Reitmayr, G., & Schmalstieg, D. (2012). Sketching up the world: In situ authoring for mobile Augmented Reality. *Personal and Ubiquitous Computing*, *16*(6), 623–630. https://doi.org/10.1007/s00779-011-0430-0

Ledermann, F., & Schmalstieg, D. (2005). APRIL: A high-level framework for creating augmented reality presentations. *IEEE Proceedings. VR 2005. Virtual Reality, 2005.*, 187–194. https://doi.org/10.1109/VR.2005.1492773

Lee, G. A., Kim, G. J., & Billinghurst, M. (2005). Immersive authoring: What You eXperience Is What You Get (WYXIWYG). *Communications of the ACM*, *48*(7), 76–81. https://doi.org/10.1145/1070838.1070840

References

Lee, Y. (2019). Using Self-Organizing Map and Clustering to Investigate Problem-Solving Patterns in the Massive Open Online Course: An Exploratory Study. *Journal of Educational Computing Research*, *57*(2), 471–490. https://doi.org/10.1177/0735633117753364

Lee, Y., Lee, S., Shamir, A., Cohen-Or, D., & Seidel, H.-P. (2005). Mesh scissoring with minima rule and part salience. *Computer Aided Geometric Design*, *22*(5), 444–465. https://doi.org/10.1016/j.cagd.2005.04.002

Leiva, G., Nguyen, C., Kazi, R. H., & Asente, P. (2020). Pronto: Rapid Augmented Reality Video Prototyping Using Sketches and Enaction. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–13. https://doi.org/10.1145/3313831.3376160

Lengauer, S., Sipiran, I., Preiner, R., Schreck, T., & Bustos, B. (2021). A Benchmark Dataset for Repetitive Pattern Recognition on Textured 3D Surfaces. *Computer Graphics Forum*, *40*(5), 1–8. https://doi.org/10.1111/cgf.14352

Lévy, B., Petitjean, S., Ray, N., & Maillot, J. (2002). Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics*, *21*(3), 362–371. https://doi.org/10.1145/566654.566590

Liker, J. K., & Pereira, R. M. (2018). Virtual and Physical Prototyping Practices: Finding the Right Fidelity Starts With Understanding the Product. *IEEE Engineering Management Review*, *46*(4), 71–85. https://doi.org/10.1109/EMR.2018.2873792

Lucrecia, M., Cecilia, S., Patricia, P., & Sandra, B. (2013). AuthorAR: Authoring tool for building educational activities based on Augmented Reality. *2013 International Conference on Collaboration Technologies and Systems (CTS)*, 503–507. https://doi.org/10.1109/CTS.2013.6567277

Luh, Y.-P., Wang, J.-B., Chang, J.-W., Chang, S.-Y., & Chu, C.-H. (2013). Augmented reality-based design customization of footwear for children. *Journal of Intelligent Manufacturing*, *24*(5), 905–917. https://doi.org/10.1007/s10845-012-0642-9

Maggiordomo, A., Cignoni, P., & Tarini, M. (2021). Texture Defragmentation for Photo-Reconstructed 3D Models. *Computer Graphics Forum*, *40*(2), 65–78. https://doi.org/10.1111/cgf.142615

Maggiordomo, A., Ponchio, F., Cignoni, P., & Tarini, M. (2020). Real-World Textured Things: A repository of textured models generated with modern photo-reconstruction tools. *Computer Aided Geometric Design*, *83*, 101943. https://doi.org/10.1016/j.cagd.2020.101943

Marner, M. R., Smith, R. T., Porter, S. R., Broecker, M. M., Close, B., & Thomas, B. H. (2011). Large Scale Spatial Augmented Reality for Design and Prototyping. In B. Furht (Ed.), *Handbook of Augmented Reality* (pp. 231–254). Springer New York. https://doi.org/10.1007/978-1-4614-0064-6_10

Marner, M. R., & Thomas, B. H. (2010). Augmented foam sculpting for capturing 3D models. *2010 IEEE Symposium on 3D User Interfaces (3DUI)*, 63–70. https://doi.org/10.1109/3DUI.2010.5444720

Marner, M. R., & Thomas, B. H. (2014). Spatial augmented reality user interface techniques for room size modelling tasks. *Proceedings of the Fifteenth Australasian User Interface Conference - Volume 150*, 39–45.

Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, *2*, 416–423 vol.2. https://doi.org/10.1109/ICCV.2001.937655

Masood, T., & Egger, J. (2019). Augmented reality in support of Industry 4.0—Implementation challenges and success factors. *Robotics and Computer-Integrated Manufacturing*, *58*, 181–195. https://doi.org/10.1016/j.rcim.2019.02.003

References

Masood, T., & Egger, J. (2020). Adopting augmented reality in the age of industrial digitalisation. *Computers in Industry*, *115*, 103–112. https://doi.org/10.1016/j.compind.2019.07.002

Mendes, P. R. C., de Albuquerque Azevedo, R. G., de Oliveira, R. G. S. G., & de Salles Soares Neto, C. (2018). Exploring an AR-based User Interface for Authoring Multimedia Presentations. *Proceedings of the ACM Symposium on Document Engineering 2018*, 1–9. https://doi.org/10.1145/3209280.3209534

Mikes, S., & Haindl, M. (2021). Texture Segmentation Benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1. https://doi.org/10.1109/TPAMI.2021.3075916

Milgram, P., & Kishino, F. (1994). *A Taxonomy of Mixed Reality Visual Displays*.

Misal, Khan, I. M., Babar, M., & Mudassir, M. U. (2019). Implementation of 3D Scanning Technique for Object Model Replication and Customization Using Rapid Prototyping. *2019 International Conference on Engineering and Emerging Technologies (ICEET)*, 1–6. https://doi.org/10.1109/CEET1.2019.8711836

Mohr, P., Kerbl, B., Donoser, M., Schmalstieg, D., & Kalkofen, D. (2015). Retargeting Technical Documentation to Augmented Reality. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 3337–3346. https://doi.org/10.1145/2702123.2702490

Mohr, P., Mandl, D., Tatzgern, M., Veas, E., Schmalstieg, D., & Kalkofen, D. (2017). Retargeting Video Tutorials Showing Tools With Surface Contact to Augmented Reality. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 6547–6558. https://doi.org/10.1145/3025453.3025688

Morosi, F., Carli, I., Caruso, G., Cascini, G., Dhokia, V., & Ben Guefrache, F. (2018). *ANALYSIS OF CO-DESIGN SCENARIOS AND ACTIVITIES FOR THE DEVELOPMENT OF A SPATIAL-AUGMENTED REALITY DESIGN PLATFORM*. DS 92: Proceedings of the DESIGN 2018 15th International Design Conference. https://doi.org/10.21278/idc.2018.0504

Morosi, F., & Caruso, G. (2020). High-fidelity Rendering of Physical Colour References for Projected-based Spatial Augmented Reality Design Applications. *Computer-Aided Design and Applications*, *18*, 343–356. https://doi.org/10.14733/cadaps.2021.343-356

Müller, L., Pfeuffer, K., Gugenheimer, J., Pfleging, B., Prange, S., & Alt, F. (2021). SpatialProto: Exploring Real-World Motion Captures for Rapid Prototyping of Interactive Mixed Reality. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–13. https://doi.org/10.1145/3411764.3445560

Nebeling, M., Lewis, K., Chang, Y.-C., Zhu, L., Chung, M., Wang, P., & Nebeling, J. (2020). XRDirector: A Role-Based Collaborative Immersive Authoring System. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–12. https://doi.org/10.1145/3313831.3376637

Nebeling, M., & Madier, K. (2019). 360proto: Making Interactive Virtual Reality & Augmented Reality Prototypes from Paper. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–13. https://doi.org/10.1145/3290605.3300826

Ng, A., Lepinski, J., Wigdor, D., Sanders, S., & Dietz, P. (2012). Designing for Low-latency Direct-touch Input. *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, 453–464. https://doi.org/10.1145/2380116.2380174

O'Hare, J., Dekoninck, E., Giunta, L., Masclet, C., & Cascini, G. (2018, February 1). *Exploring the performance of augmented reality technologies in co-creative sessions: Initial results from controlled experiments*. 5th International Conference on Design Creativity. https://doi.org/10.21278/idc.2018.0391

Park, J.-S. (2011). AR-Room: A rapid prototyping framework for augmented reality applications. *Multimedia Tools and Applications*, *55*(3), 725–746. https://doi.org/10.1007/s11042-010-0592-1

References

Park, M. ki, Lim, K. J., Seo, M. K., Jung, S. J., & Lee, K. H. (2015). Spatial augmented reality for product appearance design evaluation. *Journal of Computational Design and Engineering*, *2*(1), 38–46. https://doi.org/10.1016/j.jcde.2014.11.004

Piñones, E., Cascini, G., Caruso, G., & Morosi, F. (2021). Overcoming augmented reality adoption barriers in design: A mixed prototyping content authoring tool supported by computer vision. *Proceedings of the Design Society*, *1*, 2359–2368. https://doi.org/10.1017/pds.2021.497

Prouzeau, A., Wang, Y., Ens, B., Willett, W., & Dwyer, T. (2020). Corsican Twin: Authoring In Situ Augmented Reality Visualisations in Virtual Reality. *Proceedings of the International Conference on Advanced Visual Interfaces*, 1–9. https://doi.org/10.1145/3399715.3399743

Punpongsanon, P., Iwai, D., & Sato, K. (2015). SoftAR: Visually Manipulating Haptic Softness Perception in Spatial Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics*, *21*(11), 1279–1288. https://doi.org/10.1109/TVCG.2015.2459792

Punpongsanon, P., Iwai, D., & Sato, K. (2018). FleXeen: Visually Manipulating Perceived Fabric Bending Stiffness in Spatial Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics*, 1–1. https://doi.org/10.1109/TVCG.2018.2871044

Qian, X., He, F., Hu, X., Wang, T., Ipsita, A., & Ramani, K. (2022). ScalAR: Authoring Semantically Adaptive Augmented Reality Experiences in Virtual Reality. *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 1–18. https://doi.org/10.1145/3491102.3517665

Radu, I., & MacIntyre, B. (2009). Augmented-reality scratch: A children's authoring environment for augmented-reality experiences. *Proceedings of the 8th International Conference on Interaction Design and Children*, 210–213. https://doi.org/10.1145/1551788.1551831

Rand, W. M. (1971). Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, *66*(336), 846–850. https://doi.org/10.1080/01621459.1971.10482356

Raskar, R., Welch, G., & Fuchs, H. (1999). Spatially Augmented Reality. *Proceedings of the International Workshop on Augmented Reality : Placing Artificial Objects in Real Scenes: Placing Artificial Objects in Real Scenes*, 63–72. http://dl.acm.org/citation.cfm?id=322690.322696

Rodrigues, R. S. V., Morgado, J. F. M., & Gomes, A. J. P. (2015). A contour-based segmentation algorithm for triangle meshes in 3D space. *Computers & Graphics*, *49*, 24–35. https://doi.org/10.1016/j.cag.2015.04.003

Rumiński, D., & Walczak, K. (2014a). CARL: A Language for Modelling Contextual Augmented Reality Environments. In L. M. Camarinha-Matos, N. S. Barrento, & R. Mendonça (Eds.), *Technological Innovation for Collective Awareness Systems* (pp. 183–190). Springer. https://doi.org/10.1007/978-3-642-54734-8_21

Rumiński, D., & Walczak, K. (2014b). Dynamic composition of interactive AR scenes with the CARL language. *IISA 2014, The 5th International Conference on Information, Intelligence, Systems and Applications*, 329–334. https://doi.org/10.1109/IISA.2014.6878808

Saakes, D., Yeo, H.-S., Noh, S.-T., Han, G., & Woo, W. (2016). Mirror Mirror: An On-Body T-shirt Design System. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 6058–6063. https://doi.org/10.1145/2858036.2858282

Sagnier, C., Loup-Escande, E., Lourdeaux, D., Thouvenin, I., & Valléry, G. (2020). User Acceptance of Virtual Reality: An Extended Technology Acceptance Model. *International Journal of Human–Computer Interaction*, *36*(11), 993–1007. https://doi.org/10.1080/10447318.2019.1708612

Salonen, T., Sa¨a¨ski, J., Woodward, C., Korkalo, O., Marstio, I., & Rainio, K. (2009). *Data Pipeline From CAD to AR Based Assembly Instructions*. 165–168. https://doi.org/10.1115/WINVR2009-705

References

Sammut, C., & Webb, G. (Eds.). (2017). *Encyclopedia of Machine Learning and Data Mining*. https://link.springer.com/book/10.1007/978-1-4899-7687-1

Sander, P. V., Snyder, J., Gortler, S. J., & Hoppe, H. (2001). Texture mapping progressive meshes. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, 409–416. https://doi.org/10.1145/383259.383307

Savio, E., De Chiffre, L., & Schmitt, R. (2007). Metrology of freeform shaped parts. *CIRP Annals*, *56*(2), 810–835. https://doi.org/10.1016/j.cirp.2007.10.008

Schindler, K., & Förstner, W. (2021). *Photogrammetry* (pp. 968–970). https://doi.org/10.1007/978-3-030-63416-2_139

Scurati, G. W., Gattullo, M., Fiorentino, M., Ferrise, F., Bordegoni, M., & Uva, A. E. (2018). Converting maintenance actions into standard symbols for Augmented Reality applications in Industry 4.0. *Computers in Industry*, *98*, 68–79. https://doi.org/10.1016/j.compind.2018.02.001

Sezgin, M., & Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, *13*(1), 146–165. https://doi.org/10.1117/1.1631315

Shamir, A. (2008). A survey on Mesh Segmentation Techniques. *Computer Graphics Forum*, *27*(6), 1539–1556. https://doi.org/10.1111/j.1467-8659.2007.01103.x

Shekhar, S., Maheshwari, P., J, M., Singhal, A., Singh, K. K., & Krishna, K. (2019). ARComposer: Authoring Augmented Reality Experiences through Text. *The Adjunct Publication of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 63–65. https://doi.org/10.1145/3332167.3357116

Shu, Z., Yang, S., Wu, H., Xin, S., Pang, C., Kavan, L., & Liu, L. (2022). 3D Shape Segmentation Using Soft Density Peak Clustering and Semi-Supervised Learning. *Computer-Aided Design*, *145*, 103181. https://doi.org/10.1016/j.cad.2021.103181

Speicher, M., Hall, B. D., & Nebeling, M. (2019). What is Mixed Reality? *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–15. https://doi.org/10.1145/3290605.3300767

Speicher, M., Lewis, K., & Nebeling, M. (2021). Designers, the Stage Is Yours! Medium-Fidelity Prototyping of Augmented & Virtual Reality Interfaces with 360theater. *Proceedings of the ACM on Human-Computer Interaction*, *5*(EICS), 205:1-205:25. https://doi.org/10.1145/3461727

Stanescu, A., Mohr, P., Schmalstieg, D., & Kalkofen, D. (2022). Model-Free Authoring By Demonstration of Assembly Instructions in Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics*, 1–11. https://doi.org/10.1109/TVCG.2022.3203104

Takezawa, T., Iwai, D., Sato, K., Hara, T., Takeda, Y., & Murase, K. (2019). Material Surface Reproduction and Perceptual Deformation with Projection Mapping for Car Interior Design. *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 251–258. https://doi.org/10.1109/VR.2019.8797923

Tang, J. K. T., Duong, T.-Y. A., Ng, Y.-W., & Luk, H.-K. (2015). Learning to create 3D models via an augmented reality smartphone interface. *2015 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 236–241. https://doi.org/10.1109/TALE.2015.7386050

Tang, W., & Qiu, G. (2021). Dense graph convolutional neural networks on 3D meshes for 3D object segmentation and classification. *Image and Vision Computing*, *114*, 104265. https://doi.org/10.1016/j.imavis.2021.104265

Tarini, M., Yuksei, C., & Lefebvre, S. (2017). *Rethinking Texture Mapping*. SIGGRAPH, Los Angeles. https://doi.org/10.1145/3084873.3084911

References

Ullman, S., & Brenner, S. (1979). The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, *203*(1153), 405–426. https://doi.org/10.1098/rspb.1979.0006

Ulrich, K., & Eppinger, S. (2015). *Product Design and Development* (6th ed.). McGraw Hill.

Uva, A. E., Gattullo, M., Manghisi, V. M., Spagnulo, D., Cascella, G. L., & Fiorentino, M. (2018). Evaluating the effectiveness of spatial augmented reality in smart manufacturing: A solution for manual working stations. *The International Journal of Advanced Manufacturing Technology*, *94*(1), 509–521. https://doi.org/10.1007/s00170-017-0846-4

van Lopik, K., Sinclair, M., Sharpe, R., Conway, P., & West, A. (2020). Developing augmented reality capabilities for industry 4.0 small enterprises: Lessons learnt from a content authoring case study. *Computers in Industry*, *117*, 103208. https://doi.org/10.1016/j.compind.2020.103208

Vera, F., & Sánchez, J. A. (2016). A model for in-situ augmented reality content creation based on storytelling and gamification. *Proceedings of the 6th Mexican Conference on Human-Computer Interaction*, 39–42. https://doi.org/10.1145/2967175.2967385

Verlinden, J. C., de Smit, A., Peeters, A. W. J., & van Gelderen, M. (2003). Development of a flexible augmented prototyping system. *Journal of WSCG*, *11*(3), 496–503.

Villanueva, A., Zhu, Z., Liu, Z., Peppler, K., Redick, T., & Ramani, K. (2020). Meta-AR-app: An authoring platform for collaborative augmented reality in STEM classrooms. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–14.

Wan, H., Zou, S., Dong, Z., Lin, H., & Bao, H. (2011). MRStudio: A mixed reality display system for aircraft cockpit. *2011 IEEE International Symposium on VR Innovation*, 129–135. https://doi.org/10.1109/ISVRI.2011.5759615

Wang, G., Wang, P., Cao, D., & Luo, X. (2020). Predicting behavioural resistance to BIM implementation in construction projects: An empirical study integrating technology acceptance model and equity theory. *Journal of Civil Engineering and Management*, *26*(7), Article 7. https://doi.org/10.3846/jcem.2020.12325

Wang, J., Zeng, C., Liu, Y., & Li, S. (2013). Integrated content authoring for augmented reality based product manual assembly process instruction. *Proceedings of International Conference on Computers and Industrial Engineering, CIE*, *1*, 242–251.

Ware, C., & Rose, J. (1999). Rotating Virtual Objects with Real Handles. *ACM Trans. Comput.-Hum. Interact.*, *6*(2), 162–180. https://doi.org/10.1145/319091.319102

Wnuk, K. (2017). *The impact of snowballing literature review on the domain knowledge confidence and cognitive complexity in their thesis reports*. https://doi.org/10.13140/RG.2.2.12494.56645

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. *EASE ’14*.

Yang, M.-D., Chao, C.-F., Huang, K.-S., Lu, L.-Y., & Chen, Y.-P. (2013). Image-based 3D scene reconstruction and exploration in augmented reality. *Automation in Construction*, *33*, 48–60. https://doi.org/10.1016/j.autcon.2012.09.017

Yang, Y., Shim, J., Chae, S., & Han, T.-D. (2016). Mobile Augmented Reality Authoring Tool. *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, 358–361. https://doi.org/10.1109/ICSC.2016.42

Yoo, H. K., & Lee, J. W. (2014). Mobile augmented reality system for in-situ 3D modeling and authoring. *2014 International Conference on Big Data and Smart Computing (BIGCOMP)*, 282–285. https://doi.org/10.1109/BIGCOMP.2014.6741453

References

Yu, J., Jeon, J., Park, J., Park, G., Kim, H., & Woo, W. (2017). Geometry-Aware Interactive AR Authoring Using a Smartphone in a Wearable AR Environment. In N. Streitz & P. Markopoulos (Eds.), *Distributed, Ambient and Pervasive Interactions* (pp. 416–424). Springer International Publishing. https://doi.org/10.1007/978-3-319-58697-7_31

Zhang, Z., Wu, Y., Pan, Z., Li, W., & Su, Z. (2022). A novel animation authoring framework for the virtual teacher performing experiment in mixed reality. *Computer Applications in Engineering Education*, *30*(2), 550–563. https://doi.org/10.1002/cae.22472

Zhou, J., Lee, I., Thomas, B., Menassa, R., Farrant, A., & Sansome, A. (2012). In-Situ Support for Automotive Manufacturing Using Spatial Augmented Reality. *International Journal of Virtual Reality*, *11*(1), 33–41. https://doi.org/10.20870/IJVR.2012.11.1.2835

Zhou, Q., & Jacobson, A. (2016). *Thingi10K: A Dataset of 10,000 3D-Printing Models* (arXiv:1605.04797). arXiv. https://doi.org/10.48550/arXiv.1605.04797

Zhou, X., & Rau, P.-L. P. (2019). Determining fidelity of mixed prototypes: Effect of media and physical interaction. *Applied Ergonomics*, *80*, 111–118. https://doi.org/10.1016/j.apergo.2019.05.007

Zhu, J., Ong, S. K., & Nee, A. Y. C. (2013). An authorable context-aware augmented reality system to assist the maintenance technicians. *The International Journal of Advanced Manufacturing Technology*, *66*(9), 1699–1714. https://doi.org/10.1007/s00170-012-4451-2

Zogopoulos, V., Geurts, E., Gors, D., & Kauffmann, S. (2022). Authoring Tool for Automatic Generation of Augmented Reality Instruction Sequence for Manual Operations. *Procedia CIRP*, *106*, 84–89. https://doi.org/10.1016/j.procir.2022.02.159