




D5.3 Extended Reality lifelong learning tools/Interfaces for integrated CPSoS

<i>Authors</i>	UPAT, CTL, PASEU, CRF, USI
<i>Work Package</i>	WP5 - CPSoSaware Integration and Cross-layer Optimization supporting design-operation continuum
	<p data-bbox="418 733 516 763">Abstract</p> <div data-bbox="423 795 1382 1427" style="border: 1px solid black; padding: 10px;"> <p>The overall aim of this deliverable is to develop an AR –based CPHS user training toolkit so as to help the user adapt to changes in the environment and the dynamic CPSoS, whether these may concern a new machine that is added to the system or some new task process. Users often encounter strong outer constraints such as time or occupation, thus more immersive technologies aim to better exploit the uniqueness of AR and designing more effective virtual environments to improve the learning process. AR superimposes a computer-generated image on a user's view of the real world. Within this task we will use AR technologies to support on-site learning by giving contextually relevant and personally adapted guidance to the user. Augmented reality tools and virtual workplace simulations make the guidance lively and engaging. By capturing the user performance, the novice user can experience the tasks through the expert’s eyes, or return to his own previous performance in order to self-evaluate the learning process. The task will also develop off-site training solutions. This task will include the definition of the AR capturing and rendering components and development of appropriate networked virtual worlds for each Use Case and cater for collaborative training of users (drivers/operators). Virtual training scenarios will cover a broad range of user-desired activities, while performance and adaptation mechanisms collect data on trainee performance per task and generate recommendations to better assign users to specific tasks.</p> </div> <p data-bbox="418 1446 1308 1475">Funded by the Horizon 2020 Framework Programme of the European Union</p> <div data-bbox="418 1478 521 1548" style="text-align: center;">  </div>

Deliverable Information	
<i>Work Package</i>	WP5 - CPSoSaware Integration and Cross-layer Optimization supporting design-operation continuum
<i>Task</i>	T5.3 Extended Reality lifelong learning tools/Interfaces for integrated CPSoS
<i>Deliverable title</i>	AR based life long learning tools
<i>Dissemination Level</i>	PU
<i>Status</i>	Final
<i>Version Number</i>	1.0
<i>Due date</i>	M28
Project Information	
<i>Project start and duration</i>	01/01/2020 – 31/12/2022, 36 months
<i>Project Coordinator</i>	Industrial Systems Institute, ATHENA Research and Innovation Center 26504, Rio-Patras, Greece
<i>Partners</i>	<ol style="list-style-type: none"> 1. ATHINA-EREVNITIKO KENTRO KAINOTOMIAS STIS TECHNOLOGIES TIS PLIROFORIAS, TON EPIKOINONION KAI TIS GNOSIS (ISI) the Coordinator 2. FUNDACIO PRIVADA I2CAT, INTERNET I INNOVACIO DIGITAL A CATALUNYA (I2CAT) 3. IBM ISRAEL - SCIENCE AND TECHNOLOGY LTD (IBM ISRAEL) 4. ATOS SPAIN SA (ATOS) 5. PANASONIC AUTOMOTIVE SYSTEMS EUROPE GMBH (PASEU) 6. EIGHT BELLS LTD (8BELLS) 7. UNIVERSITA DELLA SVIZZERA ITALIANA (USI), 8. TAMPEREEN KORKEAKOULUSAATIO SR (TAU) 9. UNIVERSITY OF PELOPONNESE (UoP) 10. CATALINK LIMITED (CATALINK) 11. ROBOTEC.AI SPOLKA Z OGRANICZONA ODPOWIEDZIALNOSCIA (RTC) 12. CENTRO RICERCHE FIAT SCPA (CRF) 13. PANEPISTIMIO PATRON (UPAT)
<i>Website</i>	www.cpsosaware.eu

Control Sheet

VERSION	DATE	SUMMARY OF CHANGES	AUTHOR
0.1	15/03	Initial draft	UPAT
0.2	01/04	Pre-final draft	UPAT, CTL, PASEU, CRF, USI
1.0	10/04	Finalization	UPAT

	NAME
Prepared by	UPAT
Reviewed by	CTL, CRF
Authorized by	ISI

DATE	RECIPIENT
28/04	Coordinator
30/04	European Commission

Table of Contents

List of Figures	4
List of Terms and definitions	4
Executive Summary	5
1 Introduction	6
1.1 Scope of the Deliverable	6
1.2 Background	6
2 Methodology	8
3 3D Geometry Modeling and Preprocessing Tools	9
3.1 Utilizing 3D Saliency Mapping in Industrial Applications	10
3.2 Registration and 3D Model Retrieval	11
3.2.1 Broad-Phase Registration	12
3.2.2 Scene Segmentation and Model-to-Object Matching	12
3.2.3 Narrow-Phase Registration	14
3.3 Saliency Mapping Evaluation	14
4 Semantic and Interaction Layer	18
5 XR Training Framework	20
5.1 XR Tutorial Creation Module	20
5.2 XR Tutorial Execution Module	21
5.3 Visualization of safety zones and collision risk in space	22
6 Application of the XR-based Lifelong Learning Tools in the Manufacturing Pillar	24
7 AR-based CPHS User Training Toolkit for the Automotive Pillar	27
7.1 Interface Design	27
7.2 User Evaluation Study	29
8 Conclusions	38
Appendix A	39

List of Figures

Figure 1: [First line] (a) Original models. (b) Scaled and deformed models. [Second line] Enlarged representations of (b) with (c) red cycles for highlighting the deformed areas. (d) Heatmap visualization of HD applied to vertices. (e) Heatmap visualization of HD applied both to vertices and salient values. 10

Figure 2: Digital twin of gear model in an early stage and after three consecutive temporal moments. 11

Figure 3: (a) Original model, and heatmaps visualization of saliency mapping based on (b) the eigenvalues of small patches (spectral analysis), (c) the RPCA approach (geometrical analysis), as described in Section IV-A, (d) Wei et al., (e) Tao et al., (f) Lee et al., (g) Song et al., (h) Guo et al., (i) Song et al. (CNN), and (j) our approach. 14

Figure 4: Simplification of 3-D models using the saliency mapping of different methods, namely (I) Wei et al., (II) Tao et al., (III) Lee et al., (IV) Song et al., (V) Guo et al., (VI) Song et al. (CNN), and (VII) our approach, respectively (from up to down). (a) Heatmap visualization of the saliency mapping and simplified results using different simplification approaches, Cad (19 398 points): (b) 2000 (~ 10.3%), (c) 4000 (~ 20.6%), Block (8771 points): (b) 2000 (~ 22.8%), (c) 4000 (~ 45.6%), Part Lp (4261 points): (b) 500 (~ 11.7%), (c) 1000 (~ 23.4%), Coverrear Lp (7872 points): (b) 2000 (~ 25.4%), (c) 3000 (~ 38.1%). 15

Figure 5: Ideal patch selection based on the proposed saliency mapping. 16

Figure 6: Heatmaps of saliency mapping and denoising results using the methods. (a) Curvature co-occurrence histogram. (b) Entropy-based salient model. (c) Mesh saliency. (d) Mesh saliency via spectral processing. (e) Point-wise saliency detection. (f) Mesh saliency via CNN. (g) Our approach. 16

Figure 7: Example of the interface of some components. The drilling machine is customized by the designer. The drill lever game-object contains a trackable component, a ghost indicator, a tutorial command and the continuous lever interactable component with its appropriate public settings exposed on the inspector. 18

Figure 8: Example of tractable and interactable components. 19

Figure 9: CPSoSAAware XR tutorial framework. 20

Figure 10: 2D Visualization Panel of the Tutorial Manager during tutorial creation. 21

Figure 11: 2D Visualization Panel of the Tutorial Manager during tutorial execution. 22

Figure 12: The main menu of the XR training framework toolkit. 22

Figure 13: Example of safety zones and collision risk in space. 23

Figure 15: Screenshots of the XR training framework during the build mode. 25

Figure 16: Screenshots of the XR training framework during the train mode. 26

Figure 17: Example of the visualization of the collision risk factor. 26

Figure 18: Steering wheel chair. 27

Figure 19: Two set ups of display options. 27

Figure 20: Example of the simulator in a mixed traffic urban environment. 28

Figure 21: Pedestrians and electrical scooters are also available. 28

Figure 22: Example of the simulator in an open road rural environment with less traffic. 28

Figure 23: Example of the color annotation. 29

Figure 24: Arrow visualization method presenting (a) three different categories of occluded objects (i.e., red, cyan and yellow arrows), (b) two categories of occluded objects, (c) one category of occluded objects, (d) everything is visible to the driver. 29

List of Terms and definitions

Abbreviation	Definition
CRUD	Create Read Update Delete
JS	JavaScript
API	Application Programing Interface
JSON	JavaScript Object Notation
RDBMS	Relational Database Management System
VR	Virtual Reality
AR	Augmented Reality
XR	Extended Reality
MR	Mixed Reality
VUM	Virtual User Model
EJS	Embedded Javascript
ICP	Iterative Closest Point
HTML	Hypertext Markup Language
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
CI/CD	Continuous Integration/Continuous Development

Executive Summary

In this deliverable, we present a set of Extended Reality (XR) lifelong learning tools and interfaces implemented within three main components:

- (i) A geometry processing toolkit that includes a set of mesh and point cloud processing techniques required for modeling the 3D environment.
- (ii) A VR-based industrial training framework that includes also a semantic layer and focuses mainly on the needs of human-machine interaction for the manufacturing pillar.
- (iii) An AR-based CPHS user training toolkit customized for the automotive pillar to allow the evaluation of different interfaces.

Specifically, the VR-based training framework helps operators adapt to changes in the environment and in the dynamic CPSoS, whether these may concern a new machine that is added to the system or some new task process. It utilizes models of the workplace environment equipped with functionalities (a semantic layer) to allow interaction with a human operator (worker) in virtual reality. It can be used to transfer knowledge from experience to novice workers by allowing first to build personal training plans of a job task in a virtual gamified environment, and then to get trained in the completion of the specific task by following the recorded steps. By capturing the user performance, the novice user can experience the tasks through the expert's eyes, or return to his own previous performance in order to self-evaluate the learning process. This component is demonstrated in the manufacturing industry pillar through an inspection and repair scenario with a collaborative robot. The second component, the AR-based CPHS user training toolkit, is aimed at training a human operator in the use of new AR-based interfaces (human machine interfaces, HMIs), developed for increasing situational awareness and formal guidance, and assessing their utility and user acceptance.

1 Introduction

1.1 Scope of the Deliverable

An AR –based CPHS user training toolkit will be developed to help the user adapt to changes in the environment and the dynamic CPSoS, whether these may concern a new machine that is added to the system or some new task process. Users often encounter strong outer constraints such as time or occupation, thus more immersive technologies aim to better exploit the uniqueness of AR and design more effective virtual environments to improve the learning process. AR superimposes a computer-generated image on a user's view of the real world. Within this task we will use AR technologies to support on-site learning by giving contextually relevant and personalized guidance to the user. On-site learning has close connection to knowledge sharing, as learning can be supported both by formal guidance and knowledge shared by peers. Augmented reality tools and virtual workplace simulations make the guidance lively and engaging. Personal training plans can be built on the user models from WP2, as those models indicate personal development needs. Furthermore, re-enacting and re-creating the on-site learning experience can be realized by using sensors, AR-tools, 3D scanning and rendering and 360 video. By capturing the user performance, the novice user can experience the tasks through the expert's eyes or return to his own previous performance in order to self-evaluate the learning process. The task will also develop off-site training solutions. This task will include the definition of the AR capturing and rendering components and development of appropriate networked virtual worlds for each Use Case and cater for collaborative training of users (drivers/operators). Virtual training scenarios will cover a broad range of user-desired activities, while performance and adaptation mechanisms collect data on trainee performance per task and generate recommendations to better assign users to specific tasks (see WP3).

Below, we provide the video link for the discussed demos:

<https://drive.google.com/drive/folders/1w6Yb3TqtpXIVsBne9F3Kq6gxmfxRqWOG?usp=sharing>

1.2 Background

Technologies related to the extension of the physical world (Virtual/Augmented reality) have recently started to gain substantial traction, as their applications range from industrial to medical and entertainment purposes. This trend is also supported by the amount of capital invested in these solutions, not only hardware-wise but also from the multiple published scientific papers, which indicate great interest by the research community. As a result, high quality, immersive experiences are being delivered to the end-user at faster rates, contributing to different areas of interest.

Regarding the manufacturing sector, extended reality (XR) technologies may provide a great range of applications in any industrial environment for multiple reasons. To elaborate, the realistic 3D model representation in a virtual environment along with the respective spatial auditory feedback, can achieve great levels of embodiment for the end-user.

Likewise, the deployment of this technology in the field of manufacturing is evolving rapidly due to the new possibilities provided by AR/VR/MR applications in industrial processes. The realistic representation of 3D models inside a virtual environment can offer a sense of embodiment, helping the design and preview of workspaces, adjusted on ergonomics, efficiency and other measurable factors. Spatial augmented reality visualizations can reduce the errors during industrial processes¹, while the use of advanced user interactions within the virtual environment facilitates substantially the simulation of real-world training scenarios². Thereby, simulations of work tasks in virtual environments allow avoiding onsite learning on real devices with possibly costly materials, or to practice on potentially dangerous actions (e.g., to train inexperienced health care professionals without the risk of harming patients), while it also provides the possibility for unlimited remote training experience.

Several applications introduced maintenance task training with VR³ and MR⁴ setups but lacked authoring tools for the creation of the tutorial inside the virtual environment, or required the physical presence of the trainer

¹ Sreekanta, M. H., Sarode, A., and George, K. (2020). "Error detection using augmented reality in the subtractive manufacturing process," in 2020 10th annual computing and communication workshop and conference, Las Vegas, NV, January 6–8, 2020 CCWC. doi:10.1109/CCWC47524.2020.9031141

² Azizi, A., Yazdi, P. G., and Hashemipour, M. (2019). Interactive design of storage unit utilizing virtual reality and ergonomic framework for production optimization in manufacturing industry. *Int. J. Interact Des. Manuf.* 13, 373–381. doi:10.1007/s12008-018-0501-9

³ Dias Barkokebas, R., Ritter, C., Sirbu, V., Li, X., and Al-Hussein, M. (2019). "Application of virtual reality in task training in the construction manufacturing industry," in Proceedings of the 2019 international symposium on automation and robotics in construction (ISARC), Banff, Canada, May 21–24, 2019. doi:10.22260/ISARC2019/0107

⁴ Gonzalez-Franco, M., Pizarro, R., Cermeron, J., Li, K., Thorn, J., Hutabarat, W., et al. (2017). Immersive mixed reality for manufacturing training. *Front. Robot. AI* 4, 3. doi:10.3389/frobt.2017.00003

accordingly. Wang et al. proposed⁵ a different MR collaborative setup, with the expert streaming his/her AR environment and supervising the trainee interacting with his/her own VR environment. A platform for the design of generic virtual training procedures was proposed by Gerbaud et al.⁶, accommodating an authoring tool for the design of interactive tutorials and including a generic model to describe reusable behaviors of 3D objects and reusable interactions between those objects.

⁵ Wang, P., Bai, X., Billinghamurst, M., Zhang, S., Han, D., Lv, H., et al. (2019). "An mr remote collaborative platform based on 3d cad models for training in industry," in IEEE international symposium on mixed and augmented reality adjunct, Beijing, China, October 10–18, 2019, ISMAR-Adjunct, 91–92. doi:10.1109/ ISMAR-Adjunct.2019.00038

⁶ Gerbaud, S., Mollet, N., Ganier, F., Arnaldi, B., and Tisseau, J. (2008). "Gvt: a platform to create virtual environments for procedural training," in 2008 IEEE virtual reality conference, Reno, NV, March 8–12, 2008. 225–232. doi:10.1109/VR.2008.4480778

2 Methodology

The overarching goal is to empower human operators (being part of the cyber-physical system) and train them for better human-machine interaction or machine operation. The developed XR-based toolkit provides a set of knowledge sharing and lifelong learning tools for this purpose. Training in VR helps avoiding risks of real operating environments and allows to evaluate (with user-in-the-loop) different designs. This is achieved through four main components:

1. Developing 3D models of the environment in virtual reality to be used for simulations.
2. Add a semantic and interaction layer on the 3D models.
3. Provide authoring tools that will allow to create/build a training tutorial (to transmit knowledge).
4. Provide the interface that will allow to execute/run a training tutorial (to receive knowledge).

In other words, our developed framework provides an improved user interface that facilitates the introduction of a semantic layer into the geometric objects and the design of a tutorial by attaching plug-n-play components to the Unity game-objects. The main requirement (that has to be addressed by the designer beforehand) is the segmentation of the 3D models (e.g., of a machine or a robot) into individual parts, which may appear as separate game-objects when imported to Unity.

In the following sections, we first describe the developed techniques that allow to model the 3D environment and improve the quality of the 3D objects represented as meshes or point clouds (Section 3). For this purpose, different point cloud processing techniques were developed for denoising, as well as a point cloud registration technique that allows to map and replace a scanned (thus noisy) model with the corresponding template (i.e. ideal) model. The analysis that will be presented in the Section 3 has been published in IEEE Transactions on Multimedia⁹ and IEEE Transactions on Industrial Informatics¹⁵. Denoising and simplification is a very important step in our framework because it facilitates the definition and implementation of a semantic and interaction layer on the 3D models, as will be described in Section 4. Upon the introduction of a semantic layer on the modeled environment, Section 5 presents the XR-based training framework that allows creating and executing training tutorials providing support and lifelong learning⁷.

⁷ Pavlou, M., Laskos, D., Zacharaki, E.I., Risvas, K., Moustakas, K.: XRSISE: an XR training system for interactive simulation and ergonomics assessment. *Front. Virtual Reality* 2 (2021)

3 3D Geometry Modeling and Preprocessing Tools

The new generation 3D scanner devices have revolutionized the way information from 3D objects is acquired, making the process of scene capturing and digitization straightforward. However, the effectiveness and robustness of conventional algorithms for real scene analysis are usually deteriorated due to challenging conditions, such as noise, low resolution, and bad perceptual quality. The 3D scanning technologies are expected to create a revolution for Industry 4.0, facilitating a large number of virtual manufacturing tools and systems. Such applications require the accurate representation of physical objects and/or systems achieved through saliency estimation mechanisms that identify certain areas of the 3D model, leading to a meaningful and easier to analyze representation of a 3D object. 3D saliency mapping is, therefore, guiding the selection of feature locations and is adopted in a large number of low-level 3D processing applications including denoising, compression, simplification, and registration.

The scanning and digitization of 3D objects of the physical world has recently attracted a lot of attention. Nowadays, there are many applications in different areas (e.g., entertainment, industry, medical visualization, military, heritage, etc.) that utilize 3D objects, either in the form of point clouds or 3D meshes. Future trends show that both this type of applications and the need for reliable 3D object representation will continue to increase. However, in practical scenarios, there are many factors that inevitably affect the quality of the acquired 3D objects, such as illumination conditions or relative motion between device and target during the scanning process, which can create random fluctuation of the data, the formation of additional and unnecessary points on the surface and points away from the surface (outliers). The device itself may also generate a pattern of systematic noise that is added to the surface of the 3D object. Additionally, due to time limitations or a random non-ideal acquisition angle, the point clouds may be incomplete or deformed, which can cause errors in matching and registration⁸. Researchers strive to overcome the existing limitations, trying to provide robust solutions that can be used in realistic circumstances and challenging scenarios. One of the most common research problems upon digitization is the recognition of partially observed objects in cluttered scenes, which is fundamental in numerous applications of computer vision, such as intelligent surveillance, remote manipulation of robots in manufacturing, autonomous vehicles, automatic assembly, remote sensing, retrieval, automatic object completion. We assume the existence of scanned point clouds that have been acquired using low-resolution and low-cost 3D scanning devices. These noisy point clouds represent real cluttered scenes consisting of different partially-observed objects, denoted as *query* models. Additionally, we assume the existence of high-quality and complete 3D models, denoted as *target* models, which serve as the ideal representation of the *query* models. The *target* models have been acquired using high-resolution scanning devices and have also been post-processed to remove noise and outliers. Even though the *query* and *target* models may represent the same object, they have different resolution, orientation, while the *query* object is subject to occlusion, making the processes of matching and registration an arduous task⁹.

Visual computing technologies play an important role in several manufacturing tasks. Particularly nowadays, their role is crucial due to the new Industry 4.0 applications including manufacturing inspection¹⁰, quality control¹¹, reverse engineering¹², digital twin¹³, as well as autonomous repair operations. While the use of this new type of applications will expand, the number of the digital 3D models will also be increased, resulting in the interest for more accurate 3D model processing. The resolution and accuracy of the modern 3D scanners are constantly improving, making them even more attractive in several vision-based manufacturing tasks, allowing the accurate generation of dynamic virtual representations of physical objects which are then used for inspection. Inspecting the parts and repairing the damages or degradations are very basic tasks for many engineering or manufacturing products. More specifically, surface defect inspection is of primary importance for engineering part quality inspection, since surface defects affect not only the appearance of parts, but also their functionality, efficiency, and stability. This task mostly depends on human visual inspection by skilled inspectors. Human visual inspection is costly, labor-intensive, time-consuming, and prone to errors due to inspectors' lack of experience or fatigue, bad environmental conditions, etc. Hence, automatic inspection of the surfaces using computational techniques, which is faster, more consistent, and robust, is highly desired¹⁴. We are motivated by the fact that there are a lot

⁸ B. Lu and Y. Wang, "Matching algorithm of 3d point clouds based on multiscale features and covariance matrix descriptors," *IEEE Access*, vol. 7, pp. 137 570–137 582, 2019.

⁹ G. Arvanitis, E. Zacharaki, L. Vasa and K. Moustakas, "Broad-to-Narrow Registration and Identification of 3D Objects in Partially Scanned and Cluttered Point Clouds," in *IEEE Transactions on Multimedia*, doi: 10.1109/TMM.2021.3089838.

¹⁰ S. von Enzberg and A. Al-Hamadi, "A multiresolution approach to modelbased 3-D surface quality inspection," *IEEE Trans. Ind. Informat.*, vol. 12, no. 4, pp. 1498–1507, Aug. 2016.

¹¹ C. Piciarelli, D. Avola, D. Pannone, and G. L. Foresti, "A vision-based system for internal pipeline inspection," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3289–3299, Jun. 2019.

¹² Z. Jakovljevic, R. Puzovic, and M. Pajic, "Recognition of planar segments in point cloud based on wavelet transform," *IEEE Trans. Ind. Informat.*, vol. 11, no. 2, pp. 342–352, Apr. 2015.

¹³ F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2405–2415, Apr. 2019.

¹⁴ Y. Wang et al., "Vision based hole crack detection," in *Proc. IEEE 10th Conf. Ind. Electron. Appl.*, Jun. 2015, pp. 1932–

of new-era industrial applications that require the digitization of physical objects or systems (e.g., inspection, digital twin, Industry 4.0, quality control, reverse engineering, etc.) creating or using already scanned 3D objects. However, this digitized information is massive and raw, leading to the need of new essential and meaningful identification of features that will facilitate robust processing in various applications. These facts stress the need to focus on the development of computational models of visual attention, whose well-known outcomes are the saliency maps. Saliency maps are compact 3D representations, generated by simplifying, annotating, and/or changing the representation of a physical object/system giving more emphasis to geometrically meaningful parts. The salient features also typically satisfy important requirements such as scaling, rotation, and resolution invariance that can simplify industrial processes¹⁵.

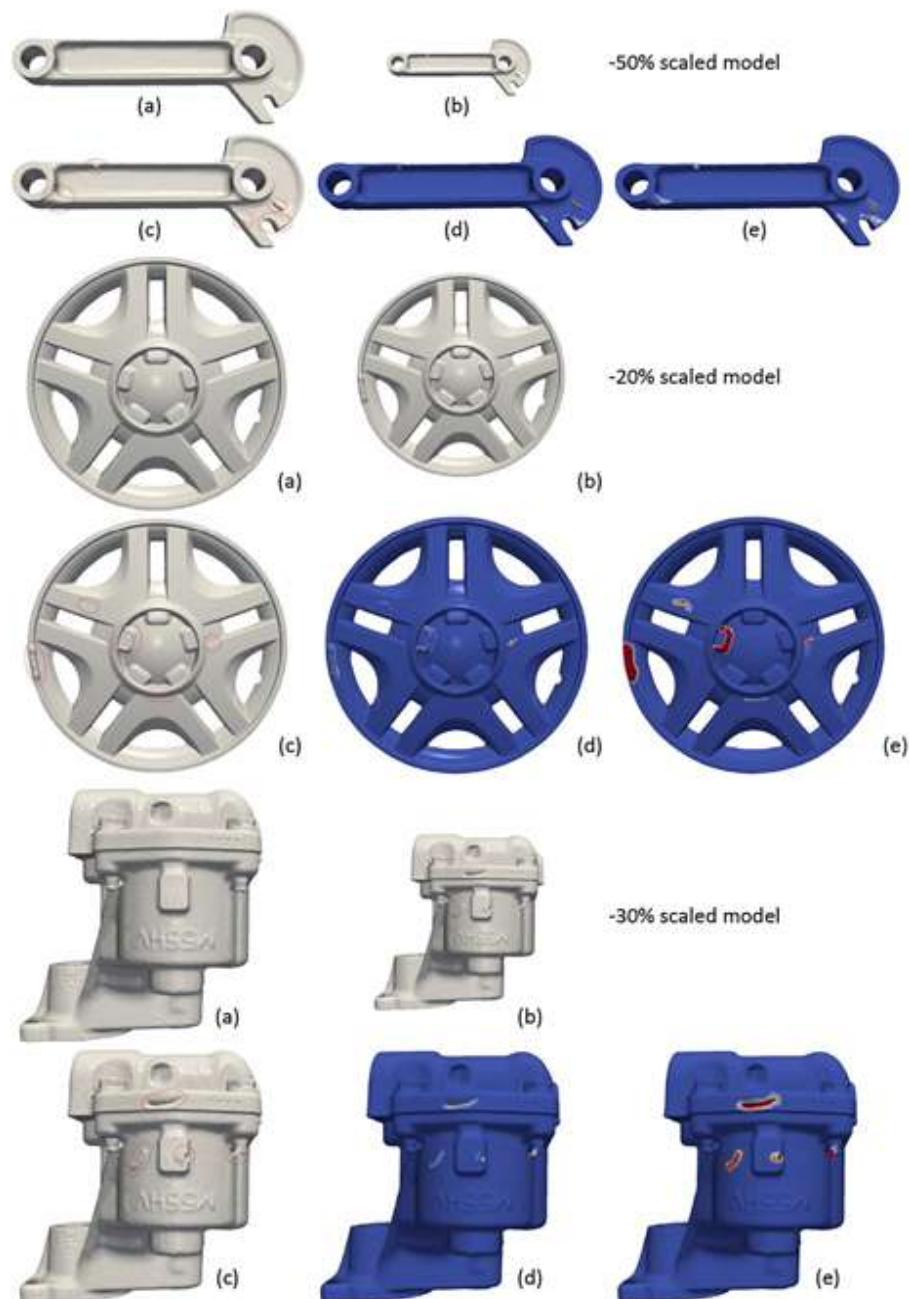


Figure 1: [First line] (a) Original models. (b) Scaled and deformed models. [Second line] Enlarged representations of (b) with (c) red cycles for highlighting the deformed areas. (d) Heatmap visualization of HD applied to vertices. (e) Heatmap visualization of HD applied both to vertices and salient values.

3.1 Utilizing 3D Saliency Mapping in Industrial Applications

This subsection presents some indicative industrial applications in which the proposed saliency mapping can be utilized, facilitating several visual tasks.

1) *Utilization in the Manufacturing Industry for Quality Control Inspections:* In the manufacturing industry, it is very common for objects to be produced in different sizes, retaining however the same form with the prototype model. Nonetheless, to assure quality, the reconstructed objects must satisfy a range of statutory and contractual obligations. In this case, inspection is used to verify and certify that the new scaled object has been manufactured

1936.

¹⁵ G. Arvanitis, A. S. Lalos and K. Moustakas, "Robust and Fast 3-D Saliency Mapping for Industrial Modeling Applications," in *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1307-1317, Feb. 2021, doi: 10.1109/TII.2020.3003455.

in full compliance with all specified requirements and constraints. Figure 1 presents examples of inspection between real-scanned industrial objects, denoted as prototype models [Figure 1 (a)], and their corresponding scaled and deformed 3D objects [Figure 1 (b)]. Our purpose is to inspect if the new manufactured 3D object has the exact same design details as the original (regarding the fidelity of its form) and also to ensure that it has not been affected by irregularities encountered during the manufacturing processes. In Figure 1 (c), we present an enlarged representation of the scaled model, presented in Figure 1 (b), with red cycles that specify the deformed areas. The purpose of this application is to automatically identify deformations or other abnormalities from the surface of the manufactured 3D object in comparison with the original model. For easier comparison, we provide a heatmap visualization of the difference between the original and the constructed model. Blue color means that there is no difference between the compared models while red color indicates a big difference. Our method can find and highlight possible differences between two objects with similar shapes comparing the saliency values of their surface. In this way, it is capable to automatically inspect degradations of the surface standards of manufactured objects despite the constraints posed by scaled manufactured objects or objects created by different materials. For the comparisons between the original and the reconstructed models, we used two different approaches. In the first approach, we deployed the Hausdorff distance (HD) [Figure 1 (d)] of the normalized models (with values in the range [0–1]), while in the second approach, we used both HD and the salient values [Figure 1 (e)].

2) *Utilization for the Creation of Digital Twins and Aging Inspection:* The proposed method supports detecting changes that can be caused by aging, comparing the saliency mapping of a 3D object having been acquired in two or more different temporal moments. In this way, our approach could be used to identify surface differences of the same object, affected by mechanical stress (e.g., a gear of a machine) or deteriorated due to environmental conditions (e.g., an ancient statue or columns). In Figure 2, we present visual representations of the same gear in four different occasions (i.e., in an early stage and after three consecutive temporal moments). This figure shows that our method is able to capture differences due to aging, so indiscernible, that even the human eye could not easily notice.

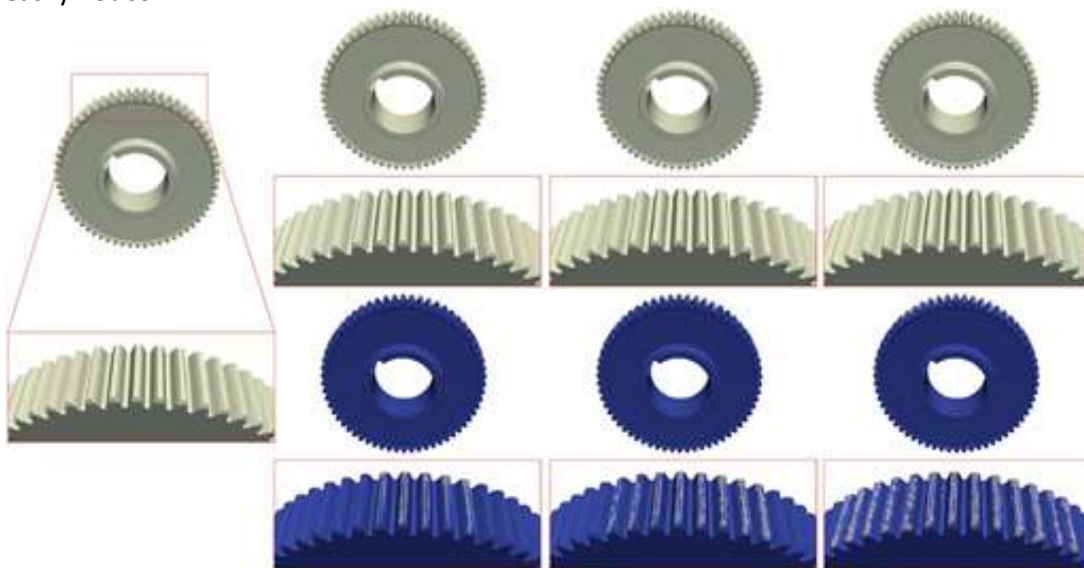


Figure 2: Digital twin of gear model in an early stage and after three consecutive temporal moments.

3.2 Registration and 3D Model Retrieval

In this task, we focus on point clouds P consisting of n vertices v . The i -th vertex v_i is represented by the Cartesian coordinates, denoted $v_i = [x_i, y_i, z_i]^T$, $\forall i = 1, \dots, n$. Thus, all the vertices can be represented as a matrix $V = [v_1, v_2, \dots, v_n] \in \mathbb{R}^{3 \times n}$. Let's also denote with Ψ_i^K the set of the K nearest neighbors of point i . Each neighboring point j can be indicated through its vertex coordinates ($v_j \in \Psi_i$) or, for simplicity, only through its index ($j \in \Psi_i$). Point cloud P represents the scanned scene, consisting of different partially visible 3D objects. The objective of this analysis is to match and replace these objects with the corresponding high-quality 3D objects that are assumed to be

available beforehand. The irrelevant objects and the noise seriously affect the optimization process^{16,17,18,19,20,21,22}. The main idea is that the obtained alignment solution can be improved if registration is guided by the most relevant salient part of the scene. The subsequent steps after the broad-phase registration include feature extraction, similarity assessment and saliency estimation. These steps are more computationally efficient when a part of the point cloud is used rather than the whole scene, as in some cases the scene might be very big. In the next sections, we describe how we robustify and accelerate computations at the same time by identifying and focusing only on salient parts of the scene that potentially correspond to the target model. This data selection step not only accelerates the calculations as it allows to search for pairs of landmarks (necessary for the final registration step) in a reduced space, but also leads to a reduction of the impact of the outliers.

- 1) Broad-phase registration: First, a fast global registration technique²³ is applied, which helps both for the decision of the matching and the final fine registration process, providing a better initial alignment between the query and the target object.
- 2) Segmentation, robustification, feature extraction and matching: In parallel, the whole scene is divided into clusters using our parameter-free implementation of the popular density-based clustering algorithm²⁴. Scene clusters, which are geometrically more similar to the registered point cloud of the previous step, are merged to create the *query* object. A robustification step is applied to facilitate the identification and removal of spurious point sets (obtained by imperfect scanning) that might blur the object boundaries affecting the registration and the execution time. The proposed feature vectors, combining pose with local multi-scale geometric information, are then extracted and used as descriptors for model to object correspondence assessment. Finally, based on the defined point similarity criterion, the best-related pairs of vertices between the matched (complete and partial) objects are identified.
- 3) Narrow-phase registration: The final step includes the calculation of a rigid transformation that brings the previously identified pairs of corresponding points into alignment.

3.2.1 Broad-Phase Registration

The first step of the matching process is to align each target model to the scanned scene by global registration, without incorporating knowledge of the model class. We have selected a recently proposed global registration algorithm²³ that has shown very good performance in different realistic datasets. The algorithm finds a number of candidate transformations by matching pairs in a roughly uniformly distributed subset of vertices of the input objects based on local shape properties (i.e., principal curvatures and the first principal direction). The optimal transformation is selected by localizing a density peak in the space of candidate rigid transformations. In order to find the density peak, a metric $d(T_1, T_2)$ is needed, which measures the distance of a transformation T_1 from a transformation T_2 .

3.2.2 Scene Segmentation and Model-to-Object Matching

1) *Point Cloud Segmentation by Density-based Clustering*: The semantic segmentation of the scene is often challenging, as the 3D objects lying in the scene might appear tangled with each other, due to abnormalities

¹⁶ H. Chen, M. Wei, Y. Sun, X. Xie, and J. Wang, "Multi-patch collaborative point cloud denoising via low-rank recovery with graph constraint," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 11, pp. 3255–3270, 2020.

¹⁷ W. Hu, X. Gao, G. Cheung, and Z. Guo, "Feature graph learning for 3d point cloud denoising," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2841–2856, 2020.

¹⁸ F. Pistilli, G. Fracastoro, D. Valsesia, and E. Magli, "Learning robust graph-convolutional representations for point cloud denoising," *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–1, 2020.

¹⁹ S. Luo and W. Hu, *Differentiable Manifold Reconstruction for Point Cloud Denoising*. New York, NY, USA: Association for Computing Machinery, 2020, p. 13301338. [Online]. Available: <https://doi.org/10.1145/3394171.3413727>

²⁰ K. Sarkar, F. Bernard, K. Varanasi, C. Theobalt, and D. Stricker, "Structured low-rank matrix factorization for point-cloud denoising," in *2018 International Conference on 3D Vision (3DV)*, 2018, pp. 444–453.

²¹ E. Mattei and A. Castrodad, "Point cloud denoising via moving rpca," *Computer Graphics Forum*, vol. 36, no. 8, pp. 123–137, 2017. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13068>

²² S. Fleishman, D. Cohen-Or, and C. T. Silva, "Robust moving least-squares fitting with sharp features," *ACM Trans. Graph.*, vol. 24, no. 3, p. 544552, Jul. 2005. [Online]. Available: <https://doi.org/10.1145/1073204.1073227>

²³ L. Hrudá, J. Dvořák, and L. Váňa, "On evaluating consensus in ransac surface registration," *Computer Graphics Forum*, vol. 38, no. 5, pp. 175–186, 2019.

²⁴ M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. AAAI Press, 1996, pp. 226–231.

created by imperfect scanning. Let's also note that supervised²⁵ or semi-supervised²⁶ learning techniques that exploit prior knowledge in the form of shape priors or large training datasets with semantic annotations cannot be applied here to facilitate segmentation, because such large scale annotations are not always available. Our method is based on the assumption that, even if different objects overlap, (i.e. their distance is small in some regions), the local point density within each object is larger than these across different objects in the scene. Therefore for parcellation of the scene, we formulate a density-based algorithm, i.e., the Density-Based Spatial Clustering of Applications with Noise (DBSCAN)²⁷, and implement a parameter-free approach, as explained in the sequel. More specifically, DBSCAN is used for the automated segmentation of a point cloud scene into separate clusters, which can be potentially used for matching and registration, reducing the total execution time.

2) *Salient Points Detection*: The purpose of this step is to identify if each high-quality target model $\mathbf{T} \in \mathbb{R}^{n_t \times 3}$ and each segmented query object $\mathbf{Q} \in \mathbb{R}^{n_q \times 3}$ (where $n_t \geq n_q$ due to occlusion, low-quality, etc), represents the same structure. To define similarity between each set of point clouds we propose descriptors that encode spectral saliency. In the following, we describe the proposed features, and how they are used to extract point-to-point correspondences, necessary for the final registration step. The feature descriptors that we use are related to the saliency map of the point cloud. Saliency is a value assigned to each vertex of a point cloud that represents its perceived importance. In the case of raw point clouds without context information, saliency characterizes the geometric properties. High values of saliency represent more perceptually protruding vertices. In this work, we assume that geometric lines, corners, and edges are more distinctive perceptually than flat areas, according to the theory of visual saliency of sight.

3) *Multi-Scale Feature Extraction*: First, the saliency values of the two compared models are normalized according to:

$$\begin{aligned} s_{t_i} &= 1 - \frac{e^{-K_s * s_{t_i}}}{s_{\max}} \quad \forall i = 1, \dots, n_t \\ s_{q_i} &= 1 - \frac{e^{-K_s * s_{q_i}}}{s_{\max}} \quad \forall i = 1, \dots, n_q \end{aligned} \quad (1)$$

where $s_{\max} = \max(\max(s_t), \max(s_q))$. Then, we perform spatial smoothing of the saliency map with a uniform kernel of increasing size and use the obtained values to form a feature vector with the multi-scale saliency values. The neighborhood size is selected as $\Psi^{k \times k}$ with $K = 5$ and $k = 1, \dots, 5$, although these parameters may vary. Smaller scales increase feature vector specificity, while larger scales smooth out noise and irregularities making the shape descriptor more robust. The use of multiple scales allows us to combine both properties and leads to unique and accurate correspondences. This process is applied for each point cloud in \mathbf{T} as well as \mathbf{Q} . Specifically, for each vertex i , we create a corresponding vector $\mathbf{a}_i \in \mathbb{R}^5$.

$$\mathbf{a}_i = \left[\frac{\sum_{j \in \Psi_i^K} s_j}{K} \quad \frac{\sum_{j \in \Psi_i^{2K}} s_j}{2K} \quad \dots \quad \frac{\sum_{j \in \Psi_i^{5K}} s_j}{5K} \right]^T \quad (2)$$

Then, we concatenate the multi-scale saliency values with the vertex coordinates to obtain the final feature representation. Finally, for each one vertex, we create the augmented feature vector $\mathbf{f} \in \mathbb{R}^8$, consisting of the vertex coordinates and the corresponding values of the vector \mathbf{a} :

$$\mathbf{f} = \mathbf{v} \cup \mathbf{a} = \begin{bmatrix} \mathbf{v}^T \\ \mathbf{a}^T \end{bmatrix} \quad (3)$$

4) *Model-to-Object Correspondence Estimation*: The feature vectors \mathbf{f} , calculated using Eq. (3), are used for the evaluation of similarity between the *query* and *target* point clouds, looking for their unique pairs of vertices \mathbf{p} which exhibit the smallest feature vector distance. It is expressed through the l_2 -norm:

$$\mathbf{p} = (\mathbf{v}_t, \mathbf{v}_q) = \arg \min_{\mathbf{v}_t, \mathbf{v}_q} \|\mathbf{f}_t - \mathbf{f}_q\|_2 \quad (4)$$

²⁵ L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4558–4567

²⁶ G. Erus, E. I. Zacharaki, and C. Davatzikos, "Individualized statistical learning from medical image databases: Application to identification of brain lesions," Medical image analysis, vol. 18, no. 3, pp. 542–554, 2014.

²⁷ M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, ser. KDD'96. AAAI Press, 1996, pp. 226–231

Finally, we keep only the first K_p pairs having the highest feature vector similarity. These K_p pairs are the best-identified correspondences between model \mathbf{T} and object \mathbf{Q} . Let's note here that we use the augmented feature vectors \mathbf{f} , which include multiscale geometric descriptors in addition to 3D location, to avoid erroneous surface mapping obtained by chance due to accidentally good local geometric fit (small Euclidean distance) of the partial point cloud. These augmented feature vectors can ensure not only spatial proximity, but also local shape similarity. At this point of the methodology, the obtained pairs of corresponding vertices include matches for each target model \mathbf{T} to each query object \mathbf{Q}^j , $j \in \{1, \dots, m_q\}$, where the m_q denotes the number of the *query* objects. To identify the correspondences, we introduce a dissimilarity factor c_j which is defined as the mean distance of the K_p pairs of the best related vertices between model \mathbf{T} and each object \mathbf{Q}^j .

$$c_j = \frac{1}{K_p} \sum_{i=1}^{K_p} \|\mathbf{f}_{t_i} - \mathbf{f}_{q_i^j}\|_2 \quad (5)$$

The lower the value of c , the more similar are the two point clouds.

3.2.3 Narrow-Phase Registration

Having identified and matched the target and query object pairs in the scene, the fine registration is achieved by identifying a set of corresponding points and then finding the optimal transformation that brings those pairs of points (control points) into alignment. In this step, we initialize the registration with the solution obtained from the global initial alignment and refine it using a weighted Iterative Closest Point (ICP) approach. The objective is, given a set of control points $\mathbf{p} = (\mathbf{v}_t, \mathbf{v}_q)$ with $\mathbf{v}_t \in \mathbf{T}$ and $\mathbf{v}_q \in \mathbf{Q}$, to estimate a rigid transformation T that minimizes a distance (or more general an error) function.

3.3 Saliency Mapping Evaluation

The proposed saliency mapping was evaluated using a) heatmaps visualization, b) 3D mesh simplification based on the saliency of the vertices, and c) a denoising application using the saliency values for finding the ideal patches.

It should be emphasized that, in most cases, there is no ground truth saliency map or a reliable metric that can be used for benchmarking purposes. The typical way to evaluate a saliency map is via subjective evaluation. The subjective evaluation can clearly show if a specific saliency mapping has achieved its purpose, applied in a specific application, and provides a fair comparison with the results of other salient mapping methods.

1) Heatmap Visualization of Saliency Mapping: Figure 3 presents the heatmap visualization of the 3D saliency mapping applied in different industrial 3D models. For easier comparison, all the results are normalized, taking values [0–1]. The used colormap for the visualization is the “jet” contenting 64 colors (deep blue = 0, deep red = 1). Saliency mapping of a 3D object must provide visual information that can be easily recognizable. This means that different areas with different characteristics will be highlighted with a different color. On the other hand, different areas with the same characteristics will be highlighted with the same color. The experimental results show that our method [in Figure 3 (e)] successfully follows this direction providing more robust and meaningful results than the other approaches. More specifically, the highest values (red colors) represent very distinctive vertices (e.g., corners), while the lowest values (blue colors) represent flat areas.

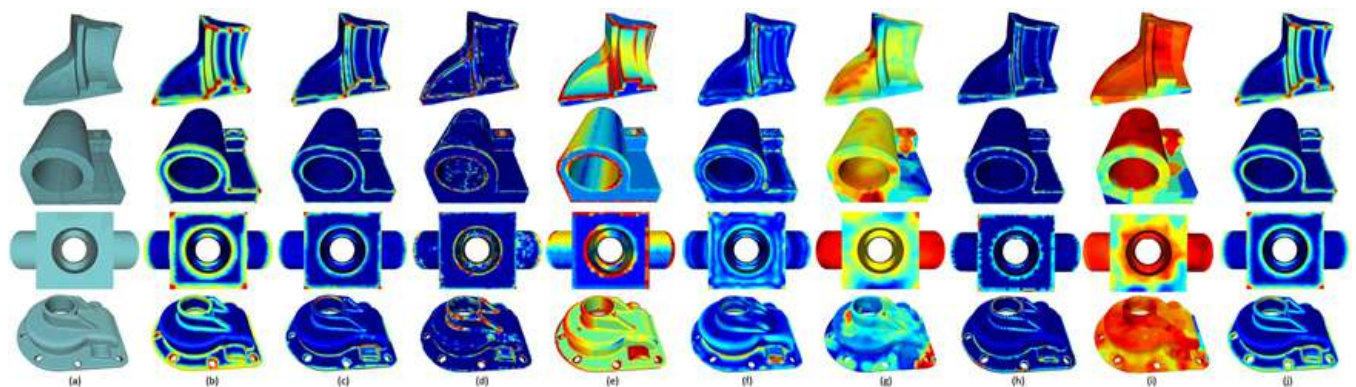


Figure 3: (a) Original model, and heatmaps visualization of saliency mapping based on (b) the eigenvalues of small patches (spectral analysis), (c) the RPCA approach (geometrical analysis), as described in Section IV-A, (d) Wei et al.²⁸, (e) Tao et al.²⁹, (f) Lee et al.³⁰, (g) Song

²⁸ N. Wei, K. Gao, R. Ji, and P. Chen, “Surface saliency detection based on curvature co-occurrence histograms,” *IEEE Access*, vol. 6, pp. 54536–54541, 2018.

²⁹ P. Tao, L. Zhang, J. Cao, and X. Liu, “Mesh saliency detection based on entropy,” in *Proc. 6th Int. Conf. Digit. Home*, Dec. 2016, pp. 288–295.

³⁰ C. H. Lee, A. Varshney, and D.W. Jacobs, “Mesh saliency,” in *Proc. ACM SIGGRAPH 2005 Papers*, New York, NY, USA, 2005, pp. 659–666.

et al.³¹, (h) Guo et al.³², (i) Song et al. (CNN)³³, and (j) our approach.

2) Simplification Based on the Saliency of Vertices: Due to the easiness of creating digital 3D content nowadays, a great amount of information can be captured and stored. The information, acquired by 3D scanners, is usually huge, creating dense 3D models that are very difficult to be efficiently handled by other applications (i.e., high computational complexity). This information must be simplified, keeping only of the most representative information, and removing least important information. Simplification is a low-level application that focuses on representing an object using a lower resolution mesh without errors or with errors that cannot be easily perceived. The main objective of a successful simplification approach is to remove only those vertices which do not offer significant geometric information to the simplified 3D object and their removal will not change significantly the shape or perceptual details of the 3D object. Following this line of thought, we suggest removing the least perceptually important vertices, preserving only the most salient vertices for the reconstruction of the new simplified 3D model. More specifically, the steps of the suggested simplification process are as follows: i) All vertices are sorted based on their salient values; ii) the Kth vertices with the higher salient values remain; iii) the rest $n - K$ less salient vertices are removed and the k-nn algorithm is used for the recreation of the new connectivity (triangulation). Figure 4 presents the simplified meshes under different simplification scenarios.

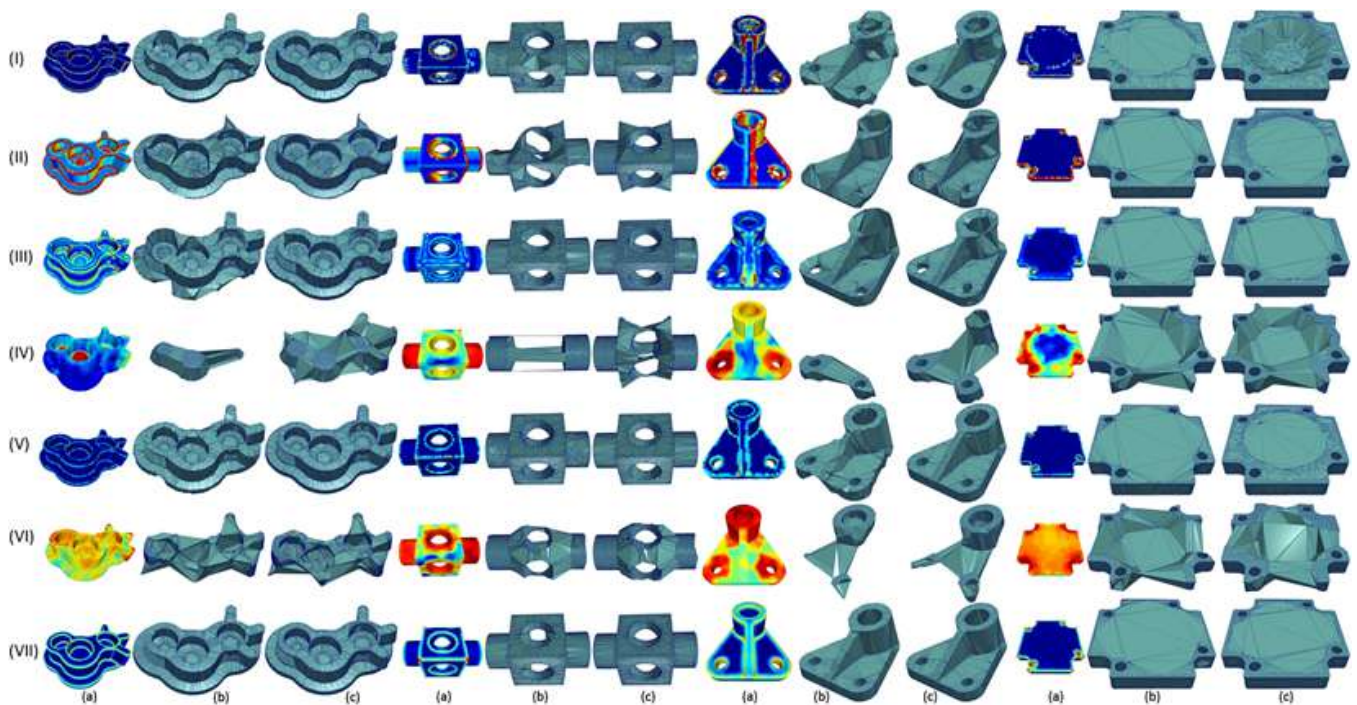


Figure 4: Simplification of 3-D models using the saliency mapping of different methods, namely (I) Wei et al., (II) Tao et al., (III) Lee et al., (IV) Song et al., (V) Guo et al., (VI) Song et al. (CNN), and (VII) our approach, respectively (from up to down). (a) Heatmap visualization of the saliency mapping and simplified results using different simplification approaches, Cad (19 398 points): (b) 2000 (~ 10.3%), (c) 4000 (~ 20.6%), Block (8771 points): (b) 2000 (~ 22.8%), (c) 4000 (~ 45.6%), Part Lp (4261 points): (b) 500 (~ 11.7%), (c) 1000 (~ 23.4%), Coverrear Lp (7872 points): (b) 2000 (~ 25.4%), (c) 3000 (~ 38.1%).

3) Feature-Aware Denoising Based on the Saliency of Vertices: Guided normals filtering has been used in previous works³⁴ providing excellent denoising results. We follow the same line of thought but we use a different way for the estimation of the ideal patch. More specifically, we select the patch that has the smallest value of Ψ (Figure 5), since it consists of “less salient” faces (flat areas that are depicted with deep blue color).

³¹ R. Song, Y. Liu, R. R. Martin, and P. L. Rosin, “Mesh saliency via spectral processing,” *ACM Trans. Graph.*, vol. 33, no. 1, pp. 6:1–6:17, Feb. 2014.

³² Y. Guo, F. Wang, and J. Xin, “Point-wise saliency detection on 3D point clouds via covariance descriptors,” *Vis. Comput.*, vol. 34, no. 10, p. 1325–1338, Oct. 2018.

³³ R. Song, Y. Liu, and P. Rosin, “Mesh saliency via weakly supervised classification-for-saliency CNN,” *IEEE Trans. Visualization Comput. Graph.*, to be published, doi: 10.1109/TVCG.2019.2928794.

³⁴ G. Arvanitis, A. S. Lalos, K. Moustakas, and N. Fakotakis, “Feature preserving mesh denoising based on graph spectral processing,” *IEEE Trans. Visualization Comput. Graph.*, vol. 25, no. 3, pp. 1513–1527, Mar. 2019.

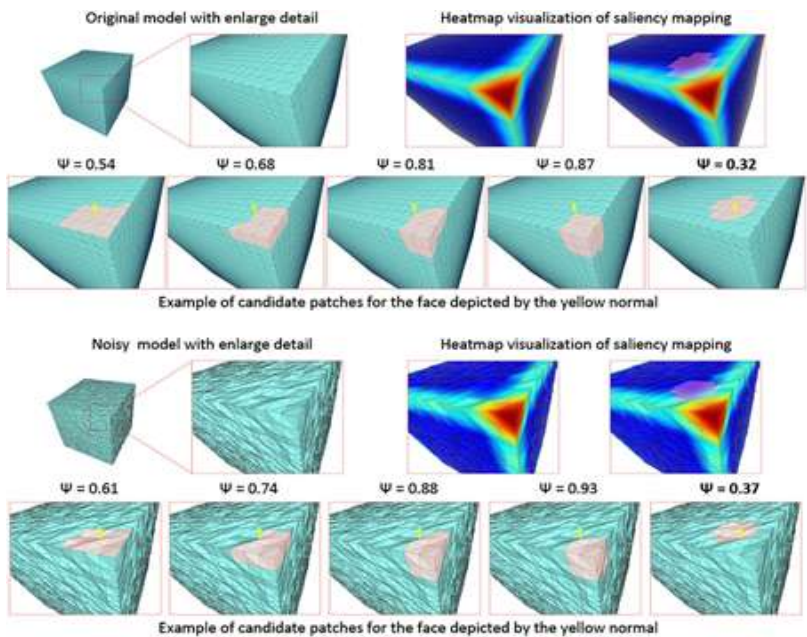


Figure 5: Ideal patch selection based on the proposed saliency mapping.

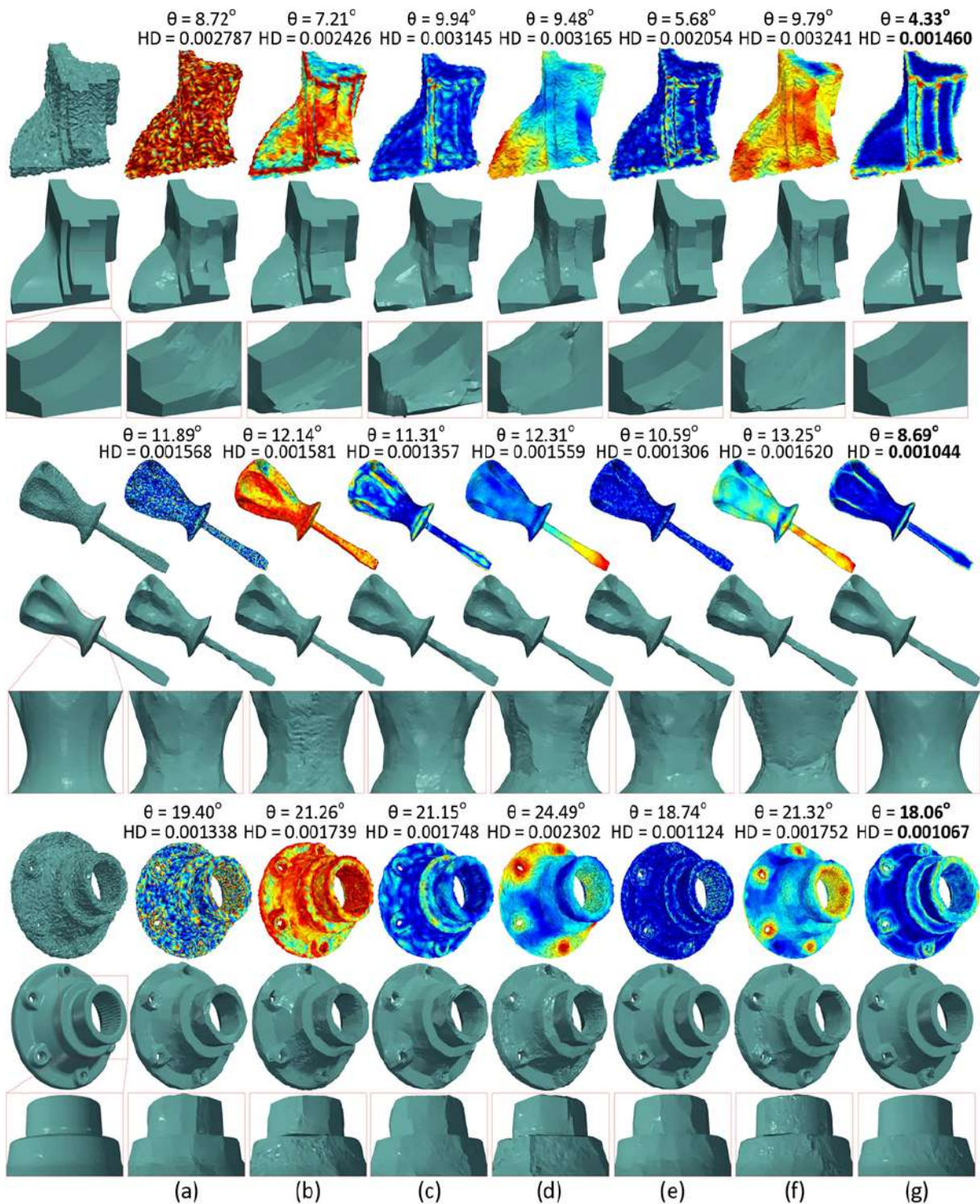


Figure 6: Heatmaps of saliency mapping and denoising results using the methods. (a) Curvature co-occurrence histogram³⁵. (b) Entropy-based

³⁵ N. Wei, K. Gao, R. Ji, and P. Chen, "Surface saliency detection based on curvature co-occurrence histograms," IEEE Access, vol. 6, pp. 54536–54541, 2018.

*salient model*³⁶. (c) *Mesh saliency*³⁷. (d) *Mesh saliency via spectral processing*³⁸. (e) *Point-wise saliency detection*³⁹. (f) *Mesh saliency via CNN*⁴⁰. (g) *Our approach*.

In Figure 5, we present an example of five candidate patches (for the face which is depicted by the yellow normal). In these examples, we show that the selected ideal patch is the one with the lowest value of Ψ (i.e., $\Psi = 0.32$ and $\Psi = 0.37$), representing the area with the less salient features. As we can observe, both the first and the last patches represent totally flat areas; however, they do not have the same Ψ value since the first patch consists of more salient triangles in comparison to the last patch, so the last area is preferable to represent the ideal patch. We also can observe that our method provides reliable results of saliency mapping even under the presence of noise, which makes it ideal for use in applications with noisy 3D models. The purpose of this example is the estimation of the most representative centroid normal (i.e., guided normal) in order to use it for a more efficient bilateral filtering⁴¹. The ideal selected patch must consist of normals with similar direction (in order to satisfy the normals' consistency). The patches that have a lot of corners or edges must be banned (i.e., high salient values in our case) since they consist of normals lying in different directions. As a result, the value of Ψ would be totally misleading since it would not represent a specific planar area. Figure 6 presents the denoising results with enlarged regions for easier comparisons. The quality of the reconstructed models is evaluated using the metrics: i) θ representing the mean angle between the normals of the ground truth and the reconstructed faces and ii) the HD.

³⁶ P. Tao, L. Zhang, J. Cao, and X. Liu, "Mesh saliency detection based on entropy," in Proc. 6th Int. Conf. Digit. Home, Dec. 2016, pp. 288–295.

³⁷ C. H. Lee, A. Varshney, and D. W. Jacobs, "Mesh saliency," in Proc. ACM SIGGRAPH 2005 Papers, New York, NY, USA, 2005, pp. 659–666.

³⁸ R. Song, Y. Liu, R. R. Martin, and P. L. Rosin, "Mesh saliency via spectral processing," ACM Trans. Graph., vol. 33, no. 1, pp. 6:1–6:17, Feb. 2014.

³⁹ Y. Guo, F. Wang, and J. Xin, "Point-wise saliency detection on 3D point clouds via covariance descriptors," Vis. Comput., vol. 34, no. 10, p. 1325–1338, Oct. 2018.

⁴⁰ R. Song, Y. Liu, and P. Rosin, "Mesh saliency via weakly supervised classification-for-saliency CNN," IEEE Trans. Visualization Comput. Graph

⁴¹ Y. Zheng, H. Fu, O. K. Au, and C. Tai, "Bilateral normal filtering for mesh denoising," IEEE Trans. Visualization Comput. Graph., vol. 17, no. 10, pp. 1521–1530, Oct. 2011.

4 Semantic and Interaction Layer

For the simulation of the industrial processes and the creation of training tutorials for lifelong learning, the 3D objects being part of the virtual environment have to be enriched with a semantic layer that describes the functionality of the components, to be used in the CPSoSaware use cases. For this purpose, we have developed a 3D Simulation framework that can be used for the interconnection of the semantic and the geometric layer (Figure 7). Our purpose was not to provide an automatic way to model the 3D scene, e.g., using input from depth cameras, but rather to develop a set of tools that allow to add and control properties of the objects, and simulate working tasks and scenarios. Such information is the **Interaction Type** which describes the kind of interaction with the component (i.e. *Pressable*, *Rotatable*, *Grabbable*, *Pullable*, *Sittable*, *Wearable*). For some rotational or translational semantics, a handful information is the **Axis** of transformation while a necessary attribute is the **Bounding Box** of the mesh for attaching colliders.

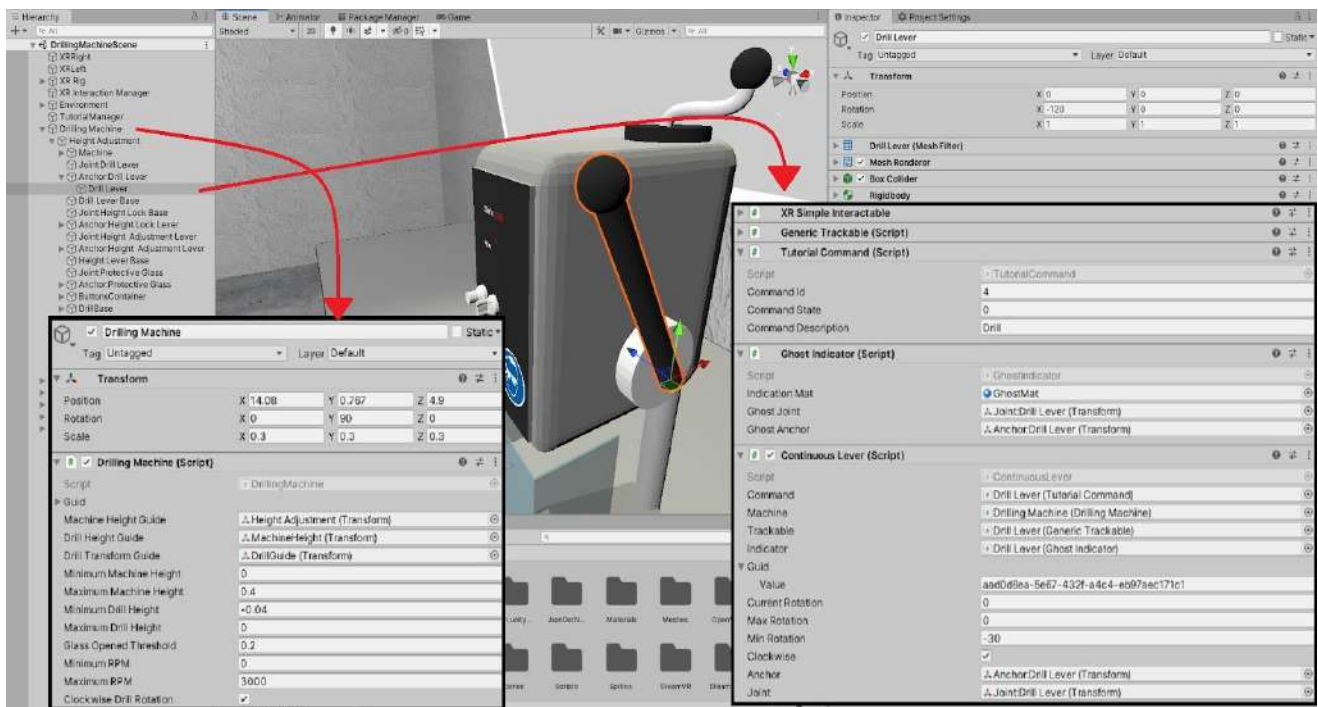


Figure 7: Example of the interface of some components. The drilling machine is customized by the designer. The drill lever game-object contains a trackable component, a ghost indicator, a tutorial command and the continuous lever interactable component with its appropriate public settings exposed on the inspector.

In order to allow interaction of a human user in virtual reality with the 3D models and tracking of this interaction, four component types are used, i.e., *interactable component*, *machine component*, *trackable component*, and a *Tutorial Manager*, as explained next. The *interactable component* holds information about the type of interaction. Interactable object instances inform the process they belong to about their state update. The *process component* contains all the interactable objects. It handles all the valid requests made by interactions and decides to act appropriately according to the specifications (Unity Components) designed within the framework. While the user interacts with an interactable object, the *trackable component* tracks the state of the object and communicates with the *Tutorial Manager* (in either build or play mode) to save the interaction or check if the interaction follows the orders of the tutorial. The Tutorial Manager handles the flow of the created tutorial and informs the user on the performed steps through a 2D visualization panel. More details on those components are provided next.

- **Interactable component:** An interactable component is a property of any object requiring physical interaction with the user, such as discrete/continuous rotating levers, pressable buttons, squeezable knobs and grabbable objects that can be placed inside specific areas. Interactable objects can hold information about the type and the state of the interaction and the input required from the VR system (e.g., controllers) to activate them. The interactable component acquires and interprets the input of the user and then communicates with its parent machine component through a message passing mechanism, resulting in the dissociation of the human interaction and the behaviour of the machine, as followed in⁴². The tutorial designer can easily attach the interactable component to a 3D object, requiring physical interaction, through the Unity inspector and tune the settings each interactable provides, such as min/max rotation values, axis of rotation, push offset etc. The interactable component design is based on the core concepts of the XR Interaction Toolkit⁴³. The core of the Interaction Toolkit is composed of a set of base Interactor and Interactable components, an Interaction Manager and helper components for improved functionality in drawing visuals and designing custom interaction events. The lack of

⁴² Tarriverdi, V., and Jacob, R. J. (2001). "Vrid: a design model and methodology for developing virtual reality interfaces," in Proceedings of the ACM symposium on virtual reality software and technology. (New York, NY:Association for Computing Machinery), VRST '01, 175–182. doi:10.1145/505008.505042

⁴³ XR interaction toolkit 1.0.0. Available at: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@1.0/manual/index.html>

available interactables (only the grab interactable was supported) led us to exploit the helper classes and design new interactables based on the framework.

- **Machine component:** It stores the set of actual machine functionalities in the form of dynamic properties, thereby allowing the operation modeling of a physical machine. The use of machine components helps in the design of self-aware machine-based behaviour and not just a collection of independent pairs of actions and their resulting effects. Through a message passing mechanism, the machine receives inputs from its child interactable components. The operational execution model of this component can determine the reactive behaviour of the machine, through a set of rules that determine the execution hierarchy of the requested actions and a set of activities-actions, using a methodology similar to Cremer et al⁴⁴. A simple example describing the aforementioned control mechanism is a machine that does not execute any action requested from any interactable, until it gets a request to get enabled (i.e., an interactable 3D button that notifies the machine component to turn on). It should be noted that, as every machine has different rules of activation of behaviours, the “design” of a machine component is the only part requiring manual effort from the designer, without the need of high expertise in software development. The designer must specify distinct commands for the parts of the machine. The commands are modeled through a simple Unity component class (named “Tutorial command”) containing the ID of the command and the dynamic state which can be then passed through the message mechanism to the machine. Next the designer has to devise the machine component (which inherits the abstract MachineBase class of the framework) with the rules and actions activated when the machine receives interactions from its “children” interactables. The disjunction and abstraction of the tracking procedure from the interaction itself, facilitates the further extension of the framework with new and more complex interactions in the future.

- **Trackable component:** Depending on the type of the performed interaction this component maintains record of the continuous or discrete state of the interactive objects, or the 3D position of the grabbable objects and the areas they are placed into. Its use is necessary only for interactions that shall be required to be tracked, while the user (instructor) creates the training scenario. The use of a trackable component instance together with the interactable component instance results in the tracking of the interaction by the Tutorial Manager (Figure 8).

- **Tutorial Manager:** Differently from the previous three components the Tutorial Manager class handles the flow of the created tutorial and informs the user on the performed steps through a 2D visualization panel.

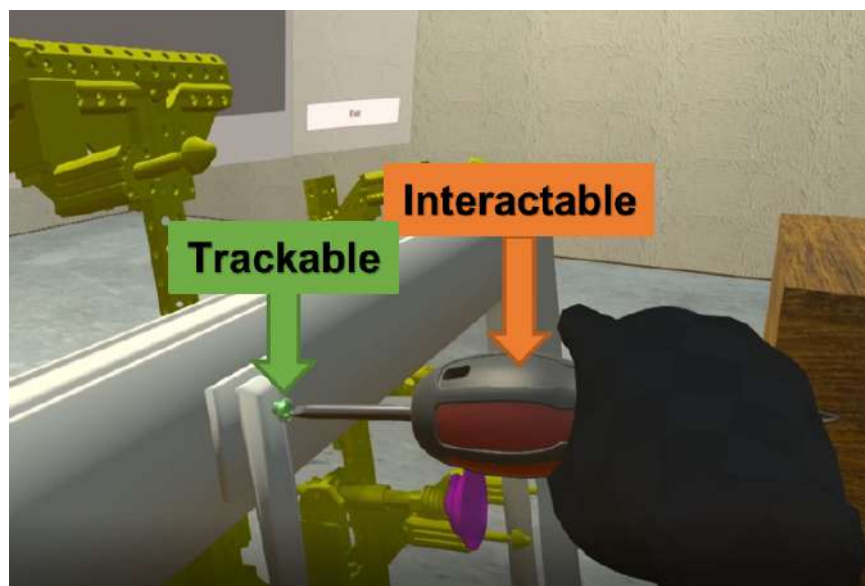


Figure 8: Example of trackable and interactable components.

⁴⁴ Cremer, J., Kearney, J., and Papelis, Y. (1995). Hcsm: a framework for behavior and scenario control in virtual environments. *ACM Trans. Model. Comput. Simul.* 5, 242–267. doi:10.1145/217853.217857

5 XR Training Framework

The simulation framework is an in-house tool in which we load 3D (geometric) models of the environment that is created from scratch. This could be either through 3D scanning and post-processing techniques or through the use of primitive shapes (which we call "templates" or ideal models in some of the described techniques) and a computer graphics software, such as Blender.

The developed XR-based toolkit consists of two modules:

1. The *XR Tutorial Creation* module which allows an (experienced) operator to create a tutorial/procedure of specific steps in VR, record these and then upload them to the CPSoSaware platform for future use.
2. The *XR Tutorial Execution* module which allows a novice operator to use the developed tutorials to get trained in VR and learn how to interact with the available CPSs. The operator follows the indications about the steps he/she needs to execute, while a "difficulty" parameter determines the number of visual hints and indications provided to support the trainee.

Both modules share common VR/XR-based technological tools mainly targeting the interaction of a real user with objects in a modeled 3D scene. Their development is based on Unity3D which facilitates the design of interactive XR training scenarios. Specifically, we exploited the Unity XR Plug-in framework, which provides the ability to integrate cross platform XR applications regardless of the utilized hardware. The user interactions are based on the extension of the XR Interaction Toolkit⁴⁵, a customizable high level interaction system, as described in the previous section.

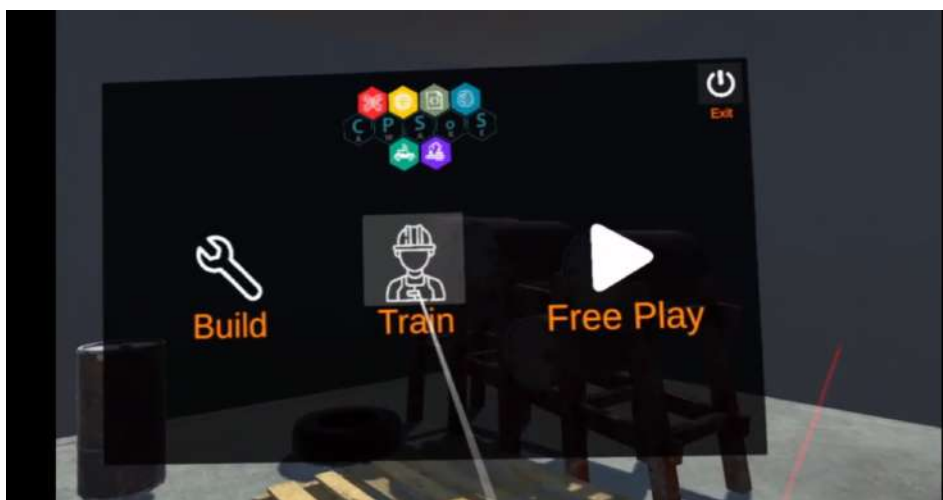


Figure 9: CPSoSaware XR tutorial framework.

For the purposes of an easy and intuitive navigation of the user inside our framework, we developed a main menu scene, through which the user can proceed on setting up the desired environment (Figure 9). Specifically, a choice is provided to the user including the selection of the virtual environment and the operating mode. Three choices for the operating mode:

- **Build Mode** to enter the *Tutorial Creation Module* and navigate in the selected virtual environment to record the steps for a certain procedure.
- **Train Mode** to enter the *Tutorial Execution Module* and to get trained selecting one of the available (previously created) tutorials in the virtual environment.
- **Free Play Mode** for both novice and experienced operators, who want to enter the selected virtual environment to get familiar with all the interactions that have been created.

5.1 XR Tutorial Creation Module

The XR tutorial creation module allows to create a training tutorial in order to educate inexperienced operators on the use of machines or industrial control panels in simple or more complex scenarios, avoiding dangers and risks inherent during the real (physical) job assignments (Figure 10). It utilizes a set of XR tools designed to simplify

⁴⁵ XR interaction toolkit 1.0.0. Available at: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@1.0/manual/index.html>

the process of creating or editing training tutorials. The XR tutorial creation module is initiated by first loading a previously designed 3D workplace of interest. Depending on the device the user can interact with objects of the environment using his bare hands or the controllers. The implementation is based on the semantic layer including the machine components, the trackable components and the tutorial manager, which are described in Section 4. Every 3D interaction is tracked automatically and marked as a new step of the training tutorial. If the user makes a mistake through the process, he/she can delete the tracked step through the 2D panel appearing in front of him/her. When all the steps are completed, the user presses the (virtual) save button, and the Tutorial Manager encodes the tutorial in a JSON formatted file.

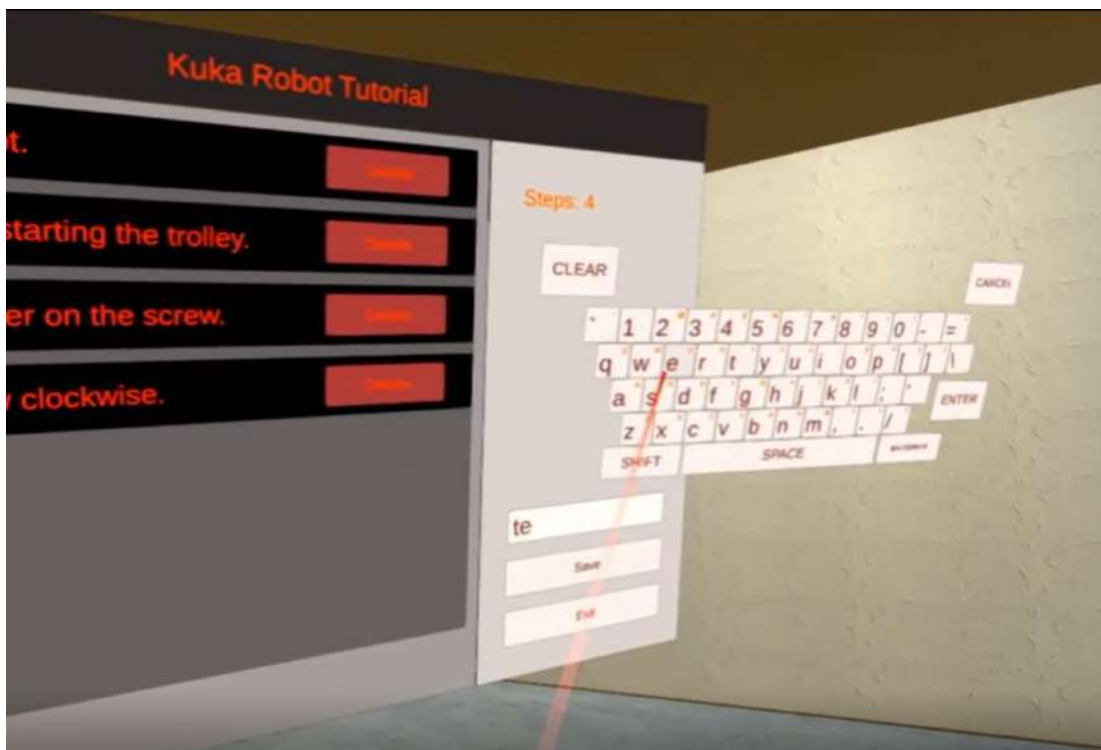


Figure 10: 2D Visualization Panel of the Tutorial Manager during tutorial creation.

5.2 XR Tutorial Execution Module

The XR tutorial execution module allows to rollout previously developed training programs, i.e., JSON encoded tutorials for learning and self-evaluation. The module includes the same components with the XR tutorial creation module, i.e., interactable components, machine components, trackable components, and a Tutorial Manager, although their utilization is different. The Tutorial Manager operates as an intermediate control mechanism for both modules. The machine, after receiving a request from an interactable component and validating it, communicates with the Tutorial Manager and sends the interaction data instance. While the role of the Tutorial Manager in the tutorial creation module was to hold every new valid interaction data, in this module the Tutorial Manager first loads the tutorial steps from a JSON file, and then, when receiving a new interaction data instance, compares the instance variables and notifies the user if the respective tutorial step was performed correctly.

The identification of a step as correct is based on the common information shared between the machine component and the tutorial manager. This information is practically a struct containing four elements: the machine UUID (Universally Unique IDentifier), the interactable UUID, the tutorial command and the command state. The first three elements are unique in determining the action. The command state is the result of the interaction which is passed from the interactable to its parent machine. It can be a button press (resulting in a Boolean state) or a lever rotation (which produces a continuous output). The machine component encodes this information, based on the rules specified by the developer, into the state it is programmed to. The interactions currently supported by the framework are simple and can result in categorical states with two or more categories or continuous-valued states (e.g., rotation angle of a handle). The latter are quantized into discrete values to facilitate the state check (by the tutorial manager), which obviously is ensured only within the level of quantization precision.

A 2D visualization panel projected in the users' front view is also utilized here to guide the trainee through the process by illustrating the sequence of performed and required actions and by displaying his/her performance after each execution task (Figure 11). The user gets notified through the panel so that he/she can proceed to the next step.



Figure 11: 2D Visualization Panel of the Tutorial Manager during tutorial execution.

Moreover, to facilitate novice trainees, several types of indications are available in this module; i.e., every interaction is visualized on the (unfamiliar) machine by highlights in non-moving parts and animation in dynamic parts (Figure 12). These ghost animations are created through a dynamic copy of the game-object's 3D mesh and recreate the motion which is required to reach the target state. In this way the trainee can easily recognize the position of the interactive components and the type of anticipated interaction. The amount of users' guidance through visual hints is determined through the selection of the "difficulty" level that is performed at the start of the XR tutorial execution module through a user interface.

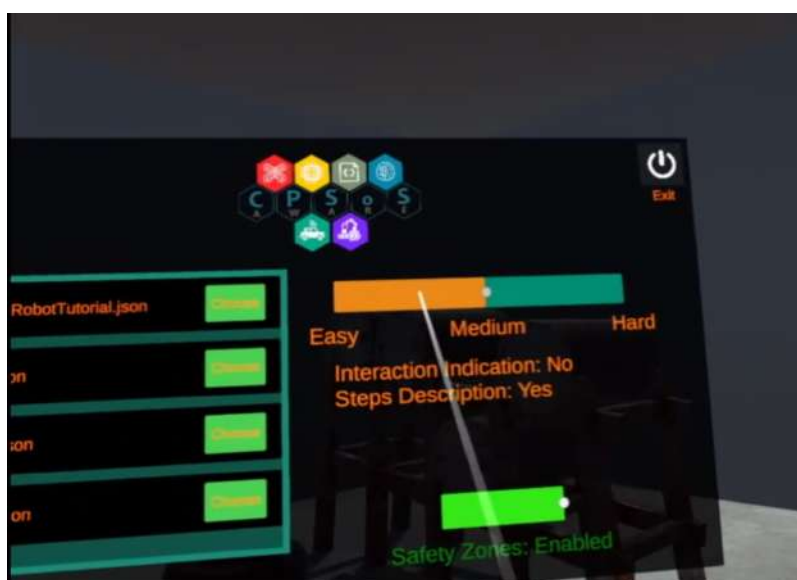


Figure 12: The main menu of the XR training framework toolkit.

In addition the user/operator can select if he/she wants to execute the tutorial with the illustration of the safety zones and collision risk factor, as explained next.

5.3 Visualization of safety zones and collision risk in space

Visualization of the simulated environment and rendering of the safety zones is performed with the Unity 3D game engine. For the visualization in the virtual world, Oculus Rift head-mounted display is employed, while the navigation of the user around the environment and the interaction with objects is rendered possible fusing the Oculus controllers. In AR, HoloLens 2 equipment is utilized. Details on the calculation of the occupancy mapping and human-robot collision risk in the volumetric space are provided in deliverable D3.4. Here we have integrated the calculated collision risk in the XR tutorial execution module, for scenarios when the user wants to get trained having also an overview of the safety zones (Figure 16).



Figure 13: Example of safety zones and collision risk in space.

6 Application of the XR-based Lifelong Learning Tools in the Manufacturing Pillar

In this section we will describe and evaluate the selected use case for our XR framework. The training scenario describes the steps required for the operation of an industrial windshield assembly task. The training tutorial includes actions for safety assurance for the machine and the worker, as well as operation instructions of drilling a hole in an object of a certain material. It consists of the following steps:

Collaborative windshield assembly main expected phases

	ROBOT	OPERATOR
1	Picks up one windshield and goes to an interactive position for the visual check	Other operations on the workcell
2	Goes to the assembly position (defined by anthropometric adaptation)	Goes to logistics containers
3	Stationary position in golden zone with assembly phases of two sensors, the rearview mirror and a cable set	Picks up the first towel and sensor
4		Performs the assembly
5		Goes to logistics containers
6	Cyclic repetition (to completed assembly number 3 to 5)	
7	Stationary position in golden zone	Releases the robot and exits the interactive zone
8	Assembles the windshield to the chassis	Performs other operations on the workcell

During all above operations the operator is capable to interact with the robot only from the front part of the windshield or from the gripper itself.

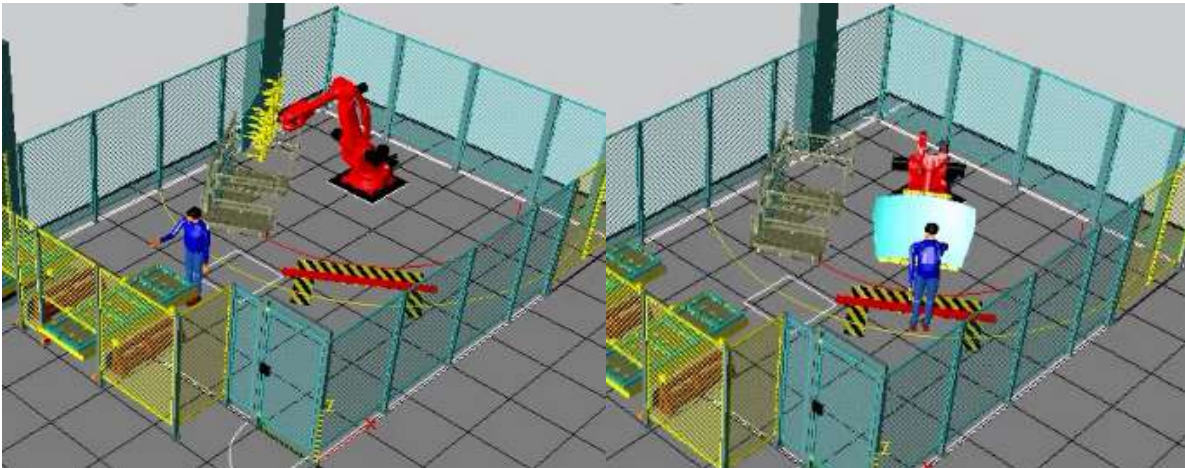


Figure 12: Some of the steps followed in the tutorial execution mode alongside with the animation and highlight indications.

XR training framework - Build mode

The following Figure 15 shows some screenshots of the XR training framework toolkit, when it is used during the build mode for the creation of a training tutorial, presenting the navigation of the user in the selected virtual environment to record the steps for a certain procedure.



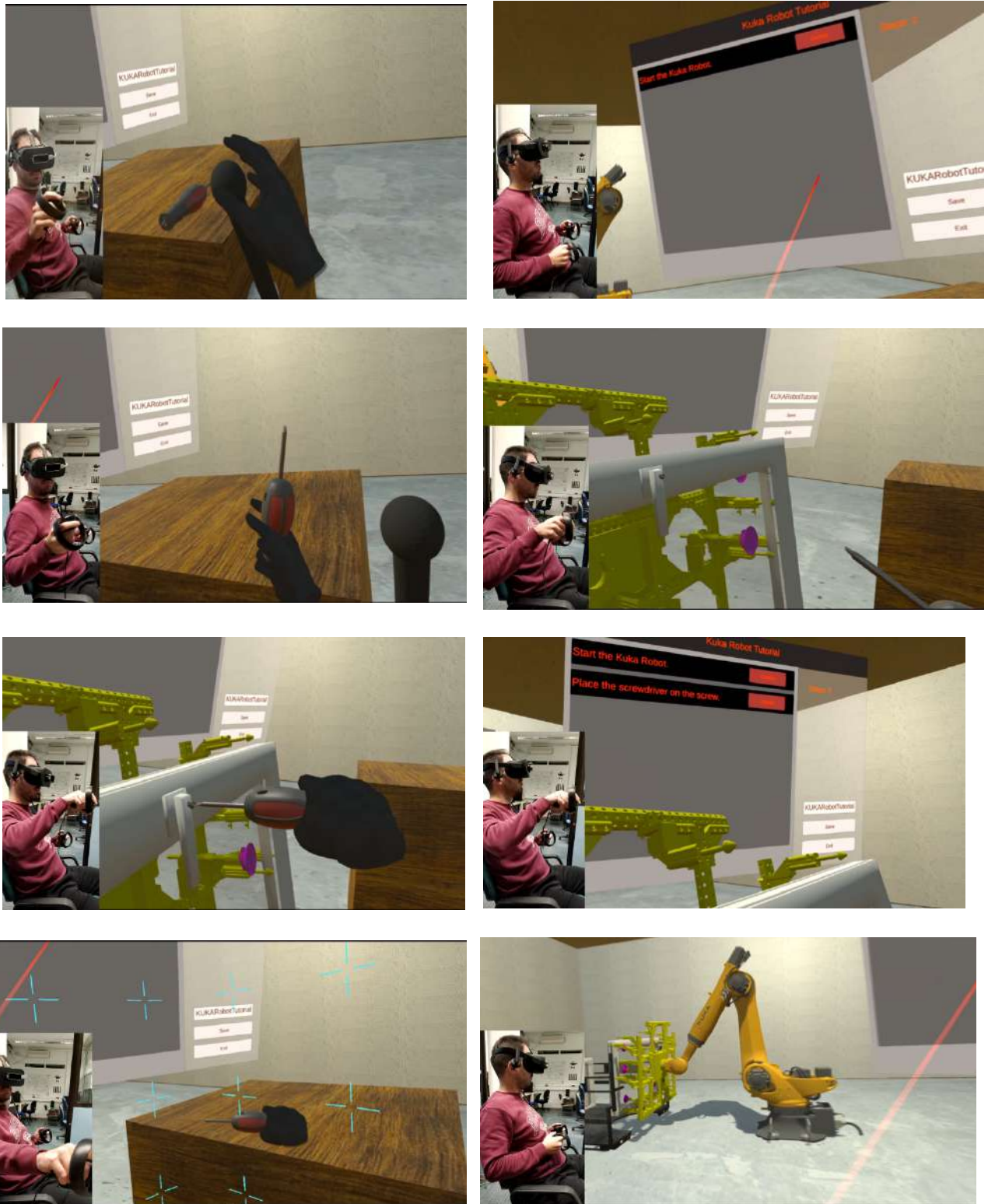


Figure 14: Screenshots of the XR training framework during the build mode.

XR training framework - Train mode

The following Figure 15 shows screenshots of the XR training framework, when it is used under the train mode, presenting how the trained operator utilize the simulator by selecting one of the available (previously created) tutorials in the virtual environment.

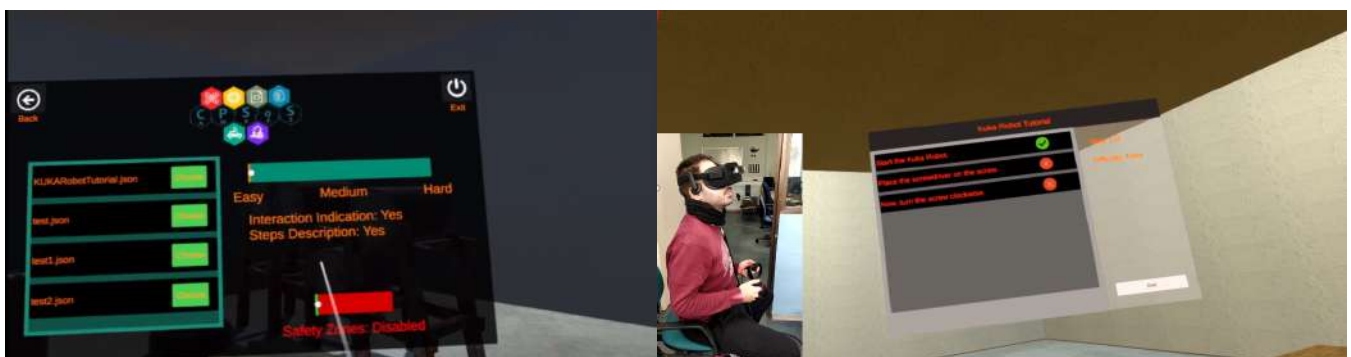




Figure 15: Screenshots of the XR training framework during the train mode.

Visualization of the collision risk factor

For the calculation and the rendering of the dangerous areas around a robotic arm, the information regarding the possible locations that the robot can occupy has to be estimated and stored. This type of information is accordingly used to define the collision risk factor. In D3.4, we present in more detail three ways for the calculation of the collision risk factor that are based on the increasing level of available information. For the sake of completeness, we present here examples (Figure 16) of the visualization of the collision risk factor throughout the simulation of the CPSoSARware XR training framework.

The simplest approach for occupancy risk prediction is to consider all possible positions in which the robot can lie at any time. The exhaustive scanning of all possible configurations is performed by recursively applying a step modification to each joint's value and calculating the position of all components at each iteration. To further improve the short-term prediction, we take into account (in addition to the robot's pose) also the angular velocity of each robotic joint at each time point. This provides a state-aware collision risk with directionality, obtained by penalizing changes in the rotation of the robotic joints and thereby improving estimation in the immediate future. The visualization allows the operators to be informed about the coming dangerous areas around them. In the dynamic area rendering, when considering the fog particle system, we also provided a top view of the safety zones, which are rendered on the board, as shown in Figure 16.

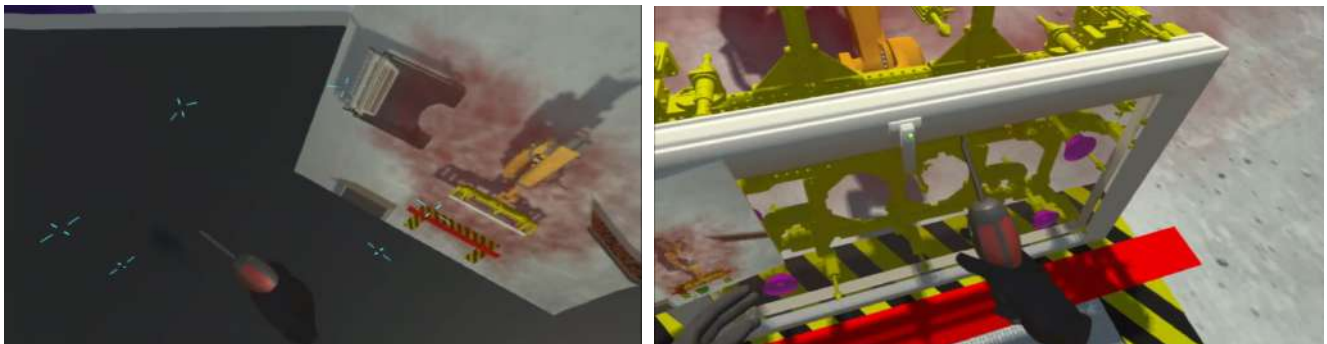


Figure 16: Example of the visualization of the collision risk factor.

7 AR-based CPHS User Training Toolkit for the Automotive Pillar

7.1 Interface Design

Modern augmented and virtual reality (AR/VR) technologies open multiple new capabilities in the way people interact, collaborate and deliver or receive information via digital interfaces. Drivers of autonomous or semi-autonomous vehicles, in cooperative driving scenarios, may especially benefit from such solutions that allow them to get trained and be familiarized, using digital copies of the environment simulating realistic traffic scenarios.

The CPSoSAAware driving simulator provides advanced interfaces, in order to support drivers of mixed traffic environments, while their situational awareness enhancement is further empowered by advanced VR and AR-based lifelong learning tools. The virtual reality tools provide the necessary framework for more experienced workers to create a VR tutorial and load it into the application. In this section, we will describe how the driving simulator that has been developed by UPAT will be used for training and familiarization of the users with the AR technologies used for the visualization of visual augment content while they are driving (more details about the visualization part of the simulator are presented in D3.4). The simulator is able to provide different types of visual information in virtual reality for situational awareness of the driver (Figure 17).



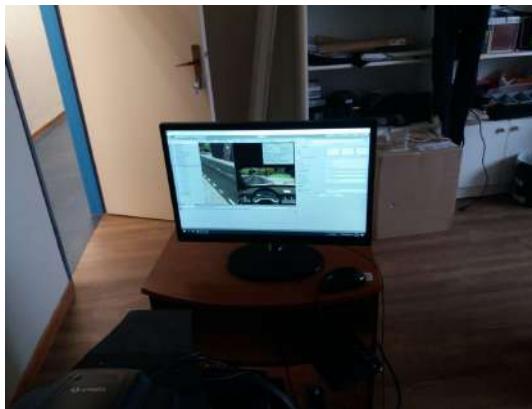
Figure 17: Steering wheel chair.

Two setups for the visualization of the AR content are available, using:

- A VR display device with leap motion sensor, Figure 18 (a)
- A monitor screen, Figure 18 (b)



(a)



(b)

Figure 18: Two set ups of display options.

The former increases the feeling of immersion while the latter can be used in combination with a capturing system (camera) in order to register and evaluate the drivers' reactions, their responses to warnings and visual content, gaze activity, the response time between finding the displayed information with their eyes and the final reaction, etc.

The objective of this task is to allow users to:

1. Learn and become more familiar with new AR visual hints.
2. Get trained to be aware of virtual signs in different traffic conditions (Figure 19-Figure 21)
3. Personalize the design of the visual information by choosing the preferable type of visualization.

4. Evaluate the efficiency, benefit and acceptability of different types of visual warnings



Figure 19: Example of the simulator in a mixed traffic urban environment.



Figure 20: Pedestrians and electrical scooters are also available.



Figure 21: Example of the simulator in an open road rural environment with less traffic.

The AR content is provided in 6 different types, which are shortly described below:

- **Arrow.** This visualization method involves a stick and an arrow tip. Each 3D Arrow's direction follows the paired object's position and the length of the stick is linearly scaled as a function of the correspondent distance.
- **3D Minimap.** This method is composed of three layers of concentric spheres, providing an estimation of the object's distance from the vehicle's position.
- **Radar.** In this method, the objects are represented as small squares in a radar-like area.
- **Wedge 3D.** This method is based on rendering objects in a form of a pyramid-like scheme. The height of the pyramid is linearly scaled w.r.t the distance between the user's vehicle reference and the object.
- **Occluded Meshes As is.** This visualization method presents the occluded information at the appropriate distance by transparently rendering the object's silhouette.

- Meshes Sphere. Similar to the previous method, however, the occluded objects are represented by spheres.

Moreover, all these visualization methods (except from the method using 3D meshes "as is") use different colours to represent different categories of objects. For example, pedestrians are represented in red, vehicles are represented in cyan and electrical scooters in yellow (Figure 22).



Figure 22: Example of the color annotation.

In the following figure, we present an example using the arrow visualization method. In Figure 23 (a), the direction, distance (based on the length of the arrow's stick body) and category (based on the arrow's colour) of some occluded objects are presented. In Figure 23 (b), the pedestrian is now visible so the red arrow disappears, the same in Figure 23 (c), where the electrical scooter appears and the yellow arrow disappears and finally, in Figure 23 (d) there are no occluded objects.

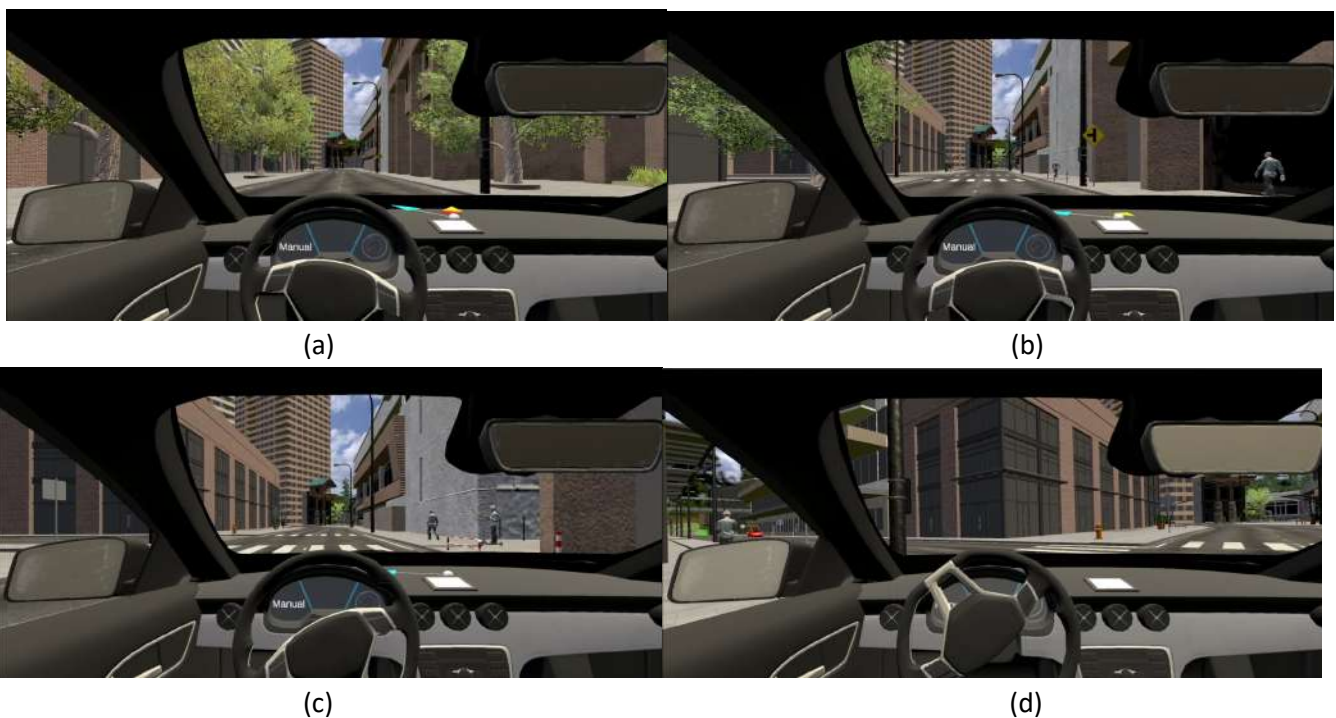


Figure 23: Arrow visualization method presenting (a) three different categories of occluded objects (i.e., red, cyan and yellow arrows), (b) two categories of occluded objects, (c) one category of occluded objects, (d) everything is visible to the driver.

7.2 User Evaluation Study

For this task, 12 adult attendees participate in the experimental process (4 females and 8 males). Most of them are employees or students of different levels (master, PhD, etc) of the University of Patras. The mean age is 27.5 years and the range is between 23 and 45.

The experimental process takes place in two parts. In the first part, the attendees became familiar with the simulator. They do not follow any specific instructions or have any time constraints. They freely drive in any

direction they want and for any time duration they feel comfortable. In this part, the simulated vehicles follow the driving rules and they are safely moving on the road. The attendees are able to use all of the provided visualization methods during their exploration of the environment of the simulator.

In the second part, the attendees have to follow a specific route in which they will face some road danger situations that have been pre-defined based on the scenario. They will utilize all the different visualization methods as a help to increase their situational awareness. Spatial indicators show the correct path that has to be followed. In this part, more aggressive behaviour of the simulated driving styles is apparent, and there are more road dangers (e.g., pedestrians and electrical scooters suddenly jump in the middle of the road).

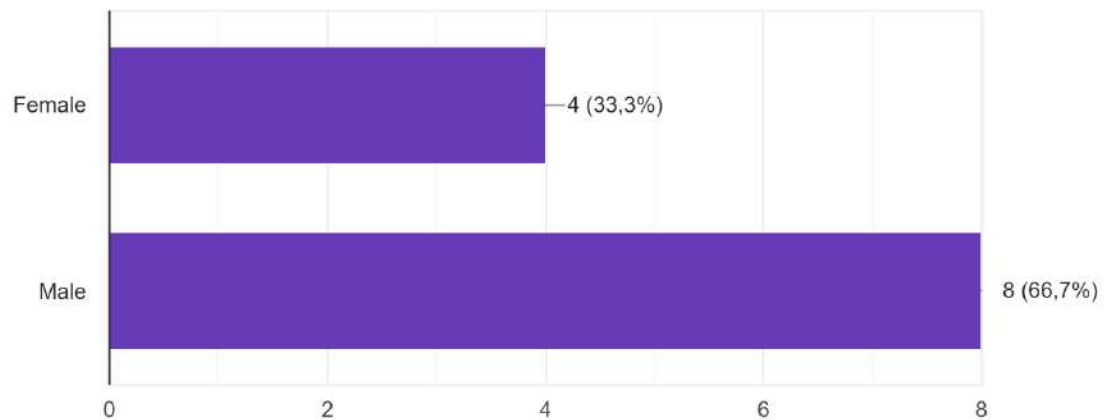
Finally, the attendees have to evaluate if the provided information via the visualization techniques satisfies their personal expectations and increases the feeling of trustworthiness and acceptance, and finally if the information is understandable and non-distractive. For this purpose, a questionnaire is provided after the end of the experimental process. The form of the questionnaire is presented in **Appendix A**.

Below, we present the results of the questionnaires.

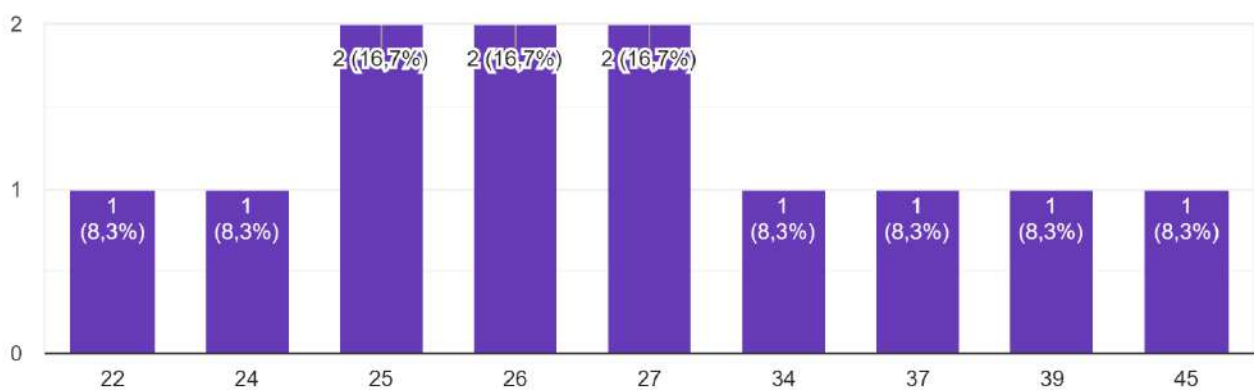
BASELINE SURVEYS

PERSONAL INFORMATION

GENDER
12 answers

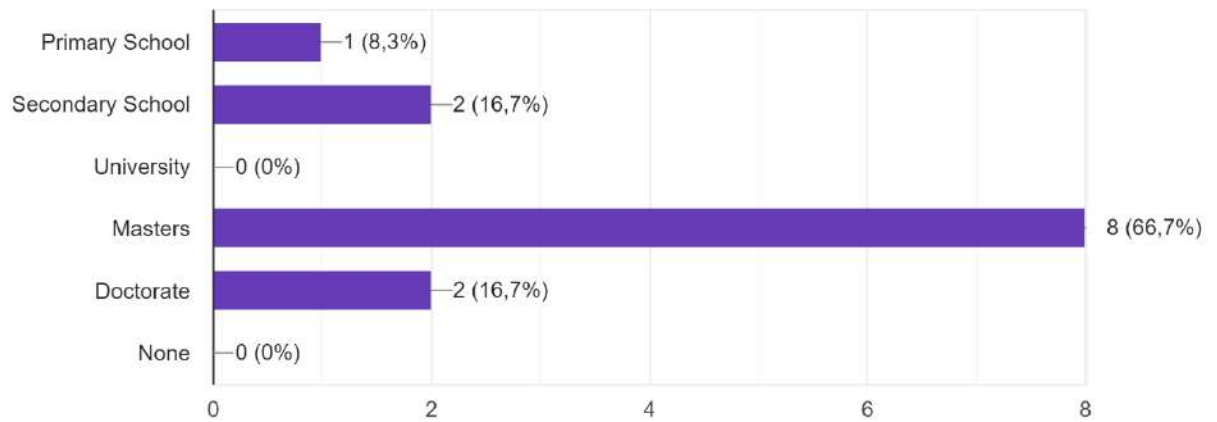


AGE
12 answers



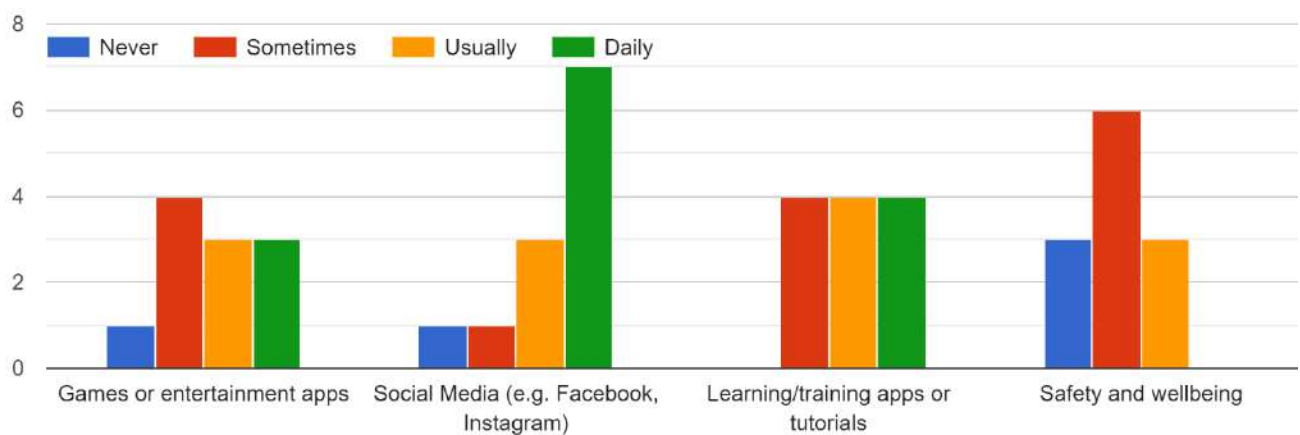
HIGHEST LEVEL OF EDUCATION YOU HAVE COMPLETED

12 answers



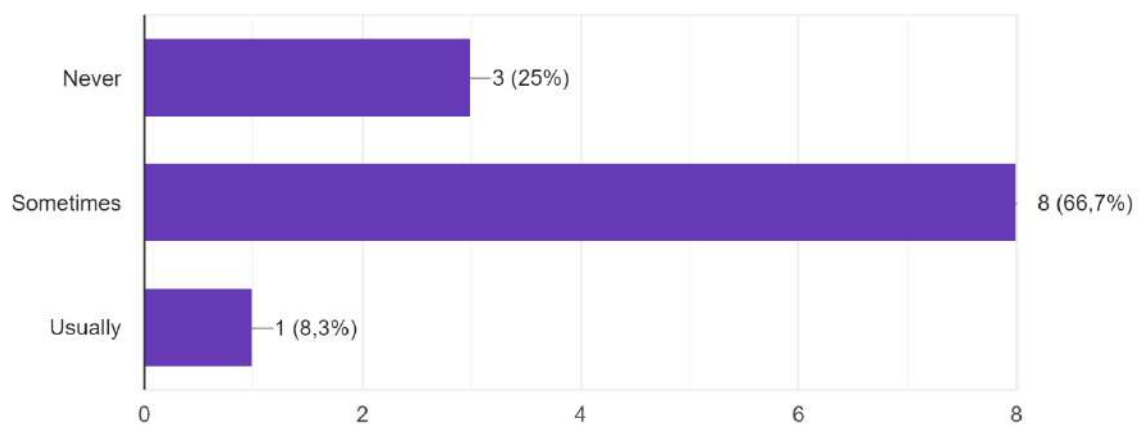
FREQUENCY IN TECHNOLOGY USE

What kind of apps do you use more frequently?



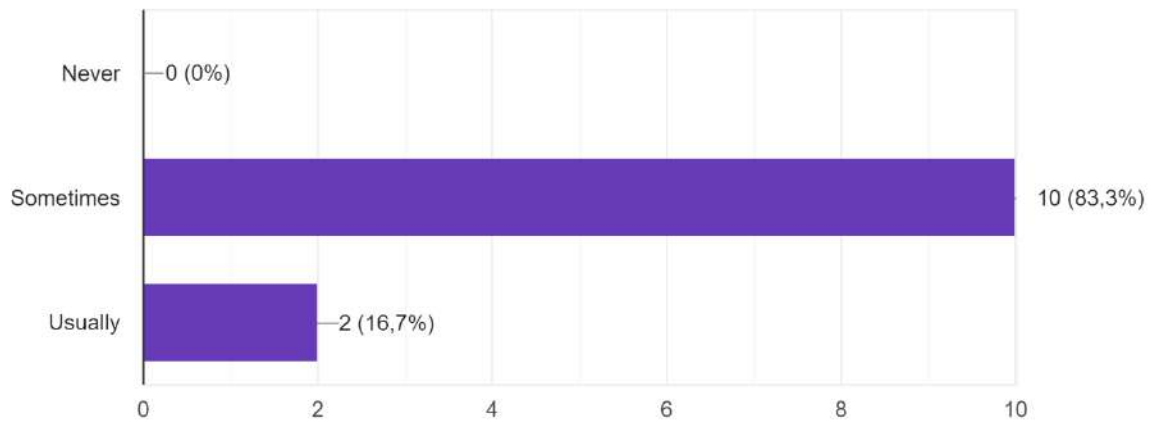
Have you ever used Augmented Reality devices or tools?

12 answers



Have you ever used Virtual Reality devices or tools?

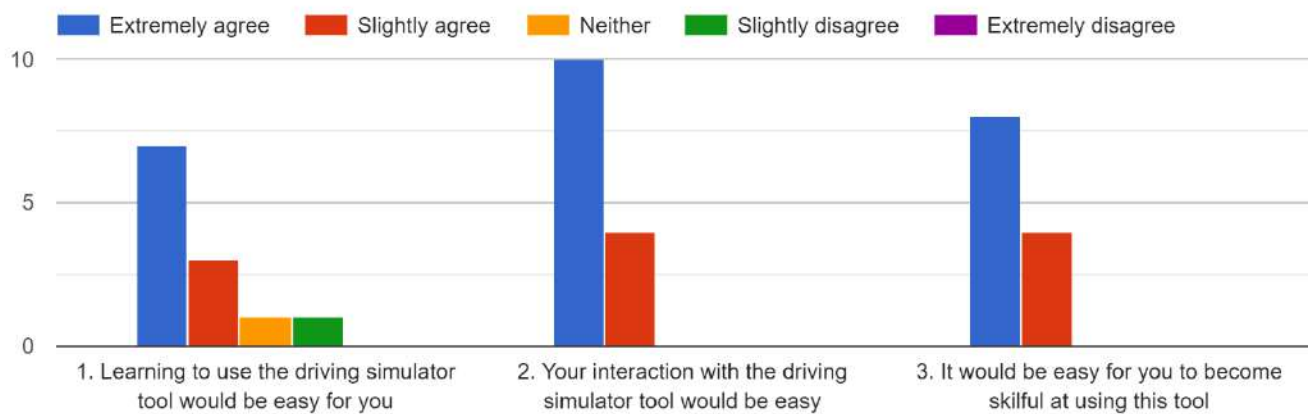
12 answers



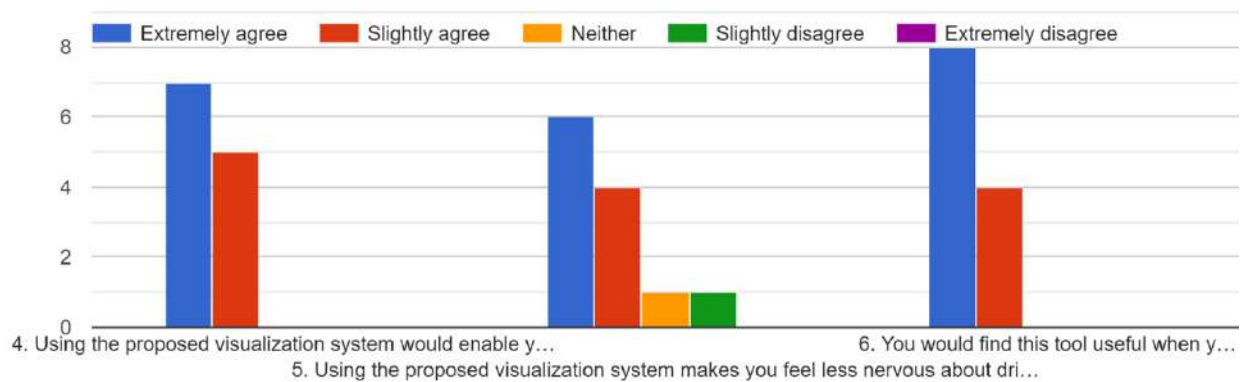
Situational awareness for cooperative driving applications in AR/VR

TECHNOLOGY ACCEPTANCE

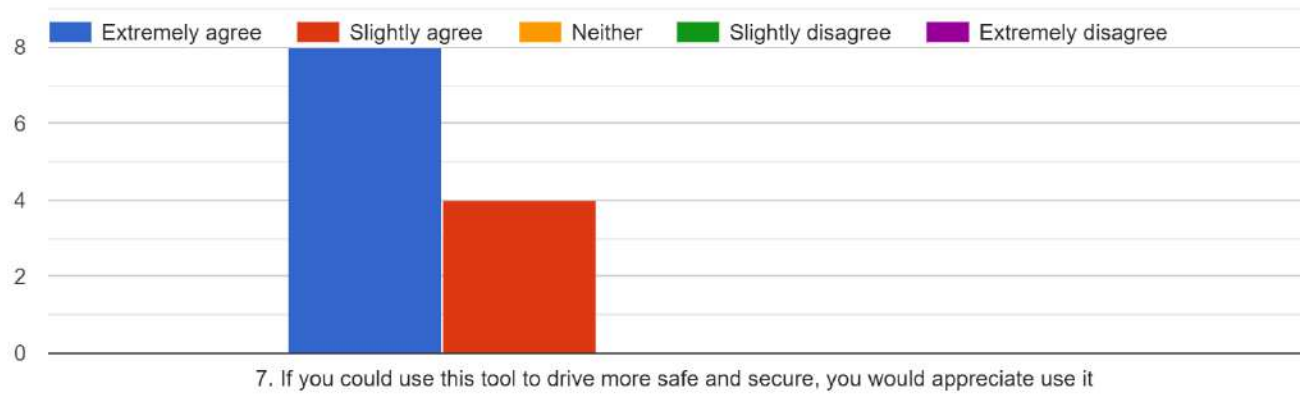
Perceived Ease of use Questions (PEU)



Perceived Usefulness Questions (PU)

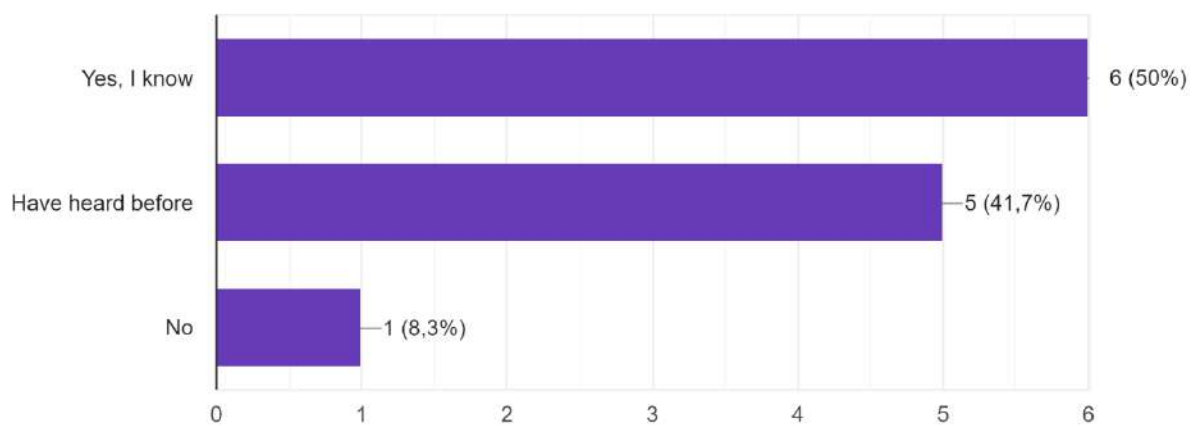


Behavioural Intention to Use (BI)



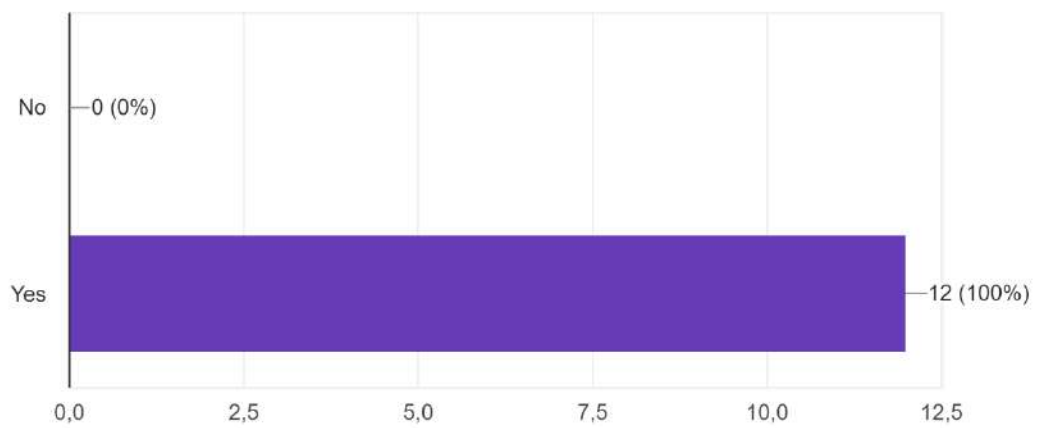
Are you aware of VR/AR training tools?

12 answers



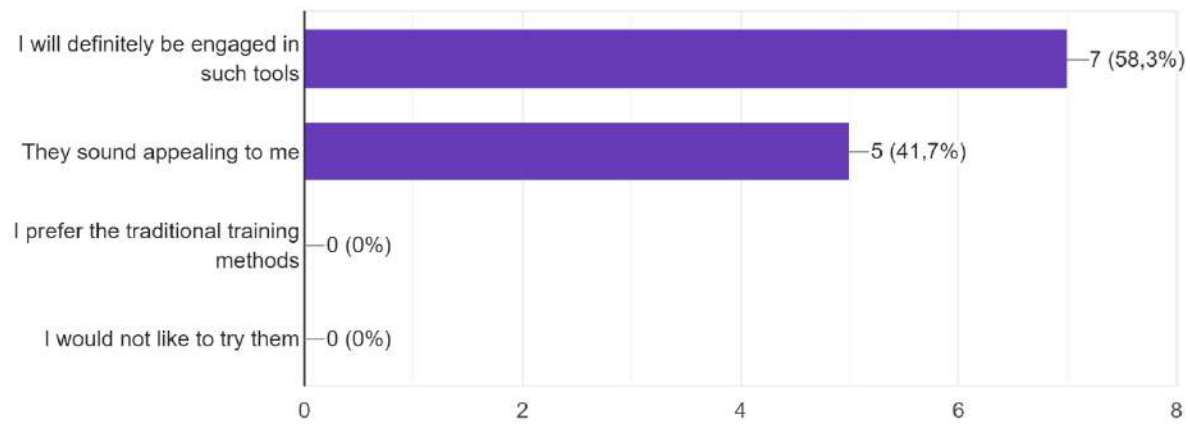
You would be interested in using VR/AR as a training/learning tool?

12 answers



Would you think that the VR/AR tools could help you to drive more safe and being well aware about the surrounding traffic environment?

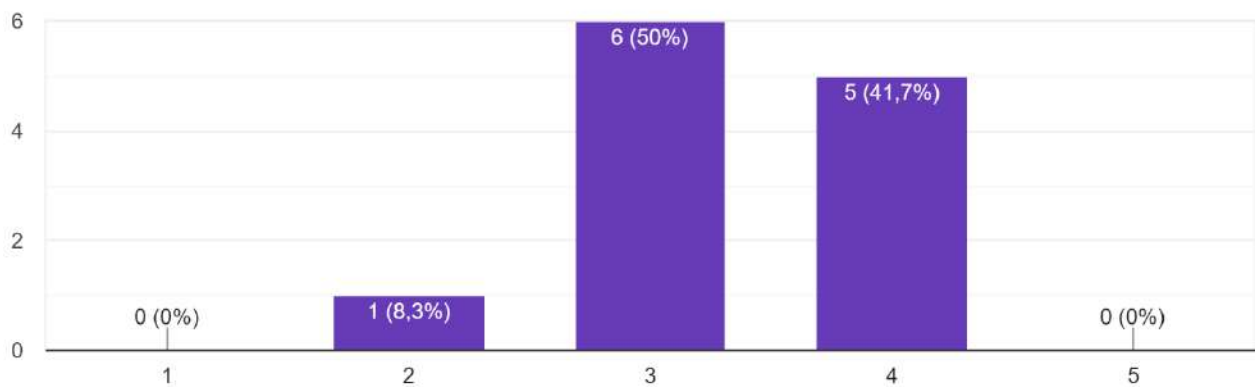
12 answers



Display Techniques

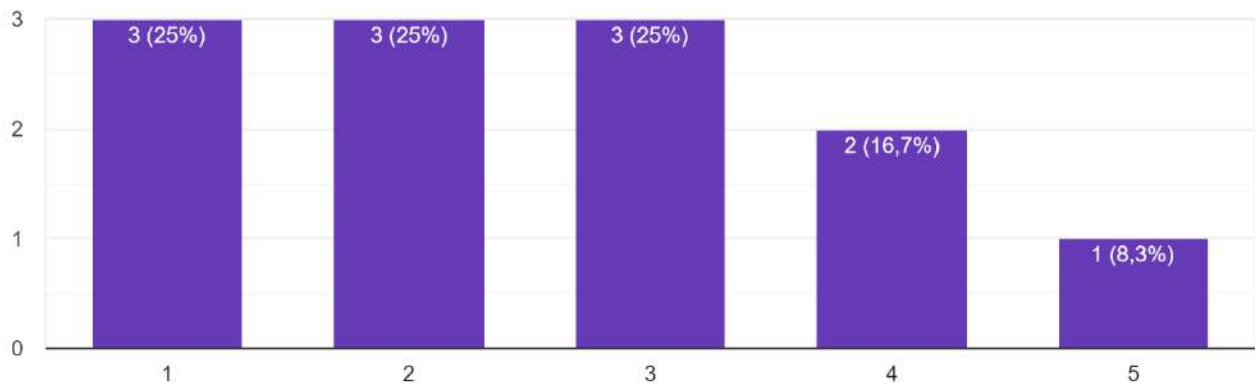
Arrows 3D

12 answers



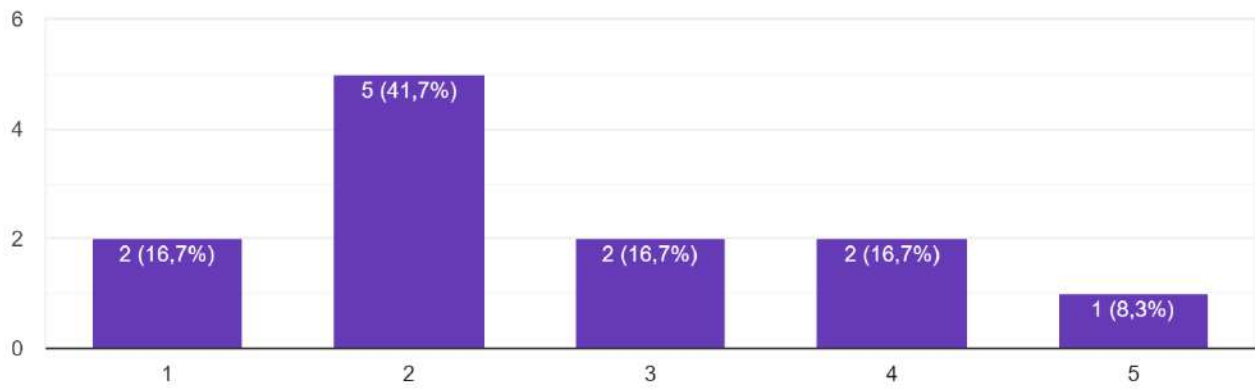
Minimap

12 answers



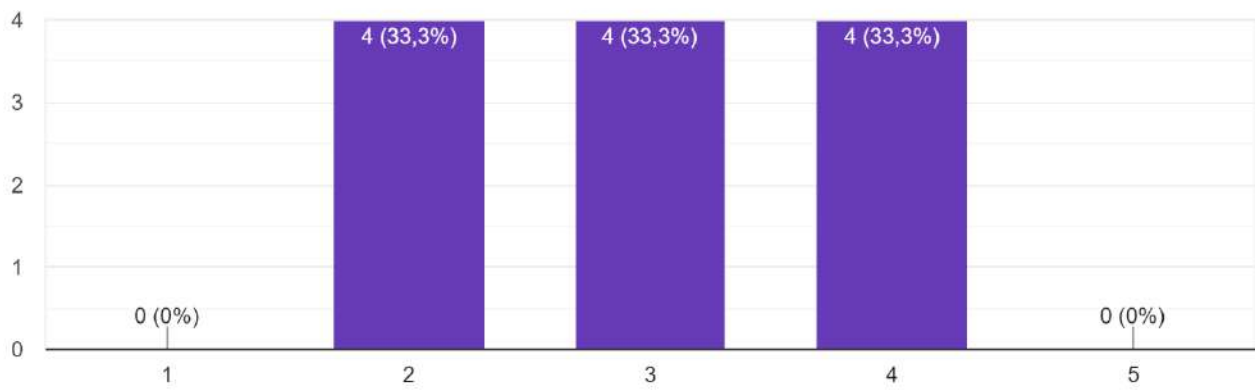
Radar

12 answers



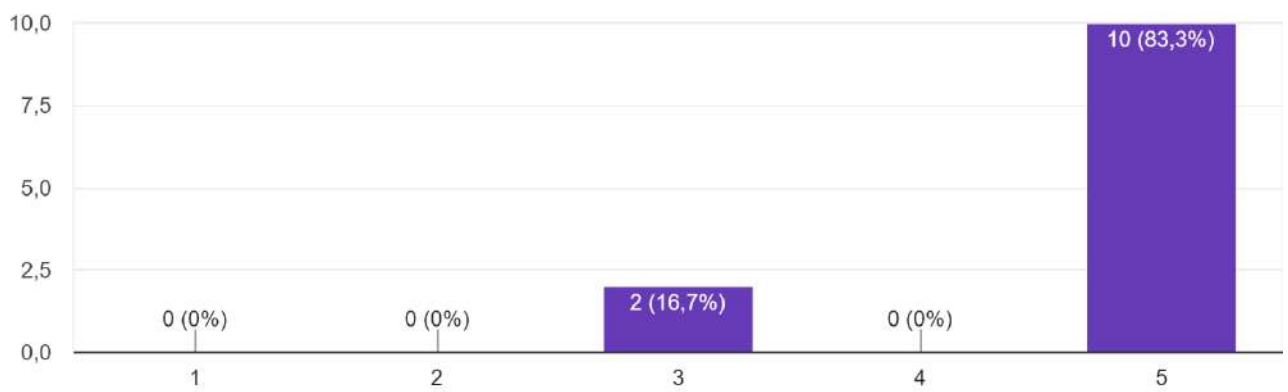
Wedge 3D

12 answers



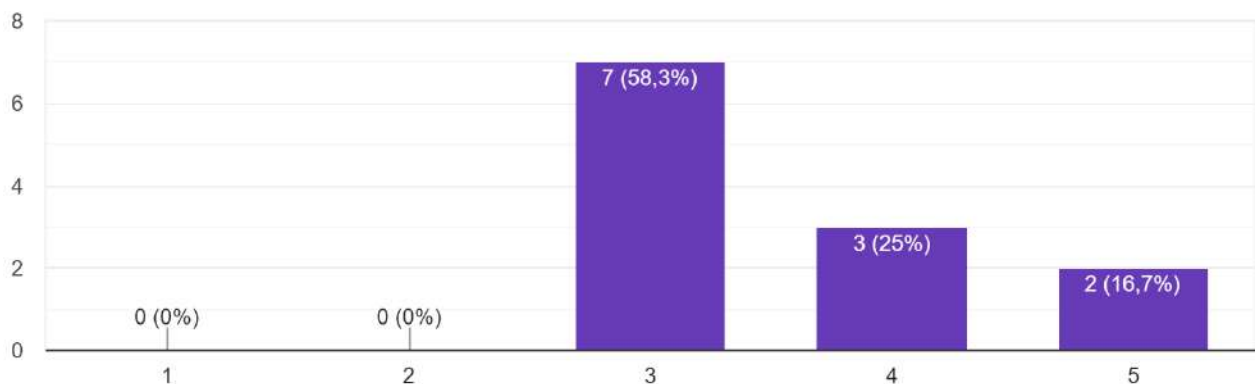
Meshes As is

12 answers



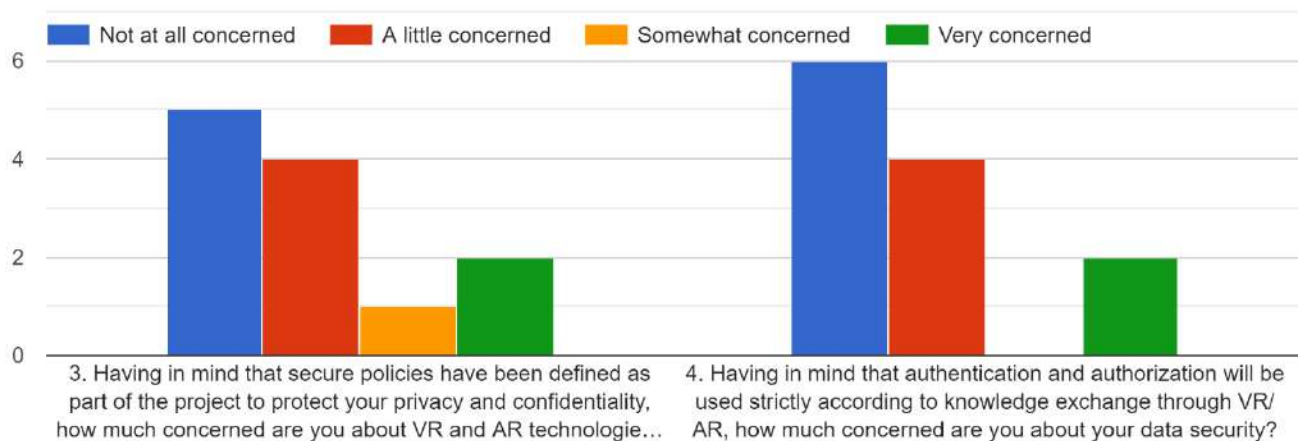
Meshes Spheres

12 answers



PRIVACY AND SECURITY QUESTIONNAIRE

Security System



To summarize, the conclusions of this analysis are presented below.

- Most of the attendees mostly use that kind of application related to social media. On the other hand, they use more infrequently applications related to safety and wellbeing.
- Most of them (75%) have used an augmented reality device, tool or application in the past, and all of them (100%) have already used a virtual reality device, tool or application.
- Most of the attendees (83.3%) agree that learning to use the driving simulator would be easy for them, and all of them agree that the interaction with the simulator was easy and they believe that could become skillful at using this tool.
- All of the attendees agree that using the proposed visualization system would enable them to drive more safely and be aware of critical upcoming events and additionally that they find this tool useful when they have to drive in a non-familiar mixed traffic environment (including vehicles, road users, bicyclists, etc). Moreover, most of the attendees are positive (83.3%) to use the proposed visualization system to make them feel less nervous about driving in an unknown area.
- All of the attendees are positive (66.7% extremely agree and 33.3% slightly agree) to use this application in order to drive more safely and securely.
- Almost all of them (11 of 12) are aware of VR/AR training tools and all of them (100%) are interested in using VR/AR technology as a training or learning tool.
- Regarding the visualization methods, attendees provide a big range of different personalized preferences.

However, the most popular method is the presentation of the occluded objects as transparent meshes displaying their silhouette as they actually are (83.3% with 5 to 5), and the less popular is the minimaps (50% with less than 3 to 5).

8 Conclusions

The work within Task 5.3 is aligned with the work in WP5 “CPSoSaware Integration and Cross-layer Optimization supporting design-operation continuum”. In this deliverable we presented a knowledge sharing platform and a XR-based toolkit for the development and the subsequent rollout of new training programs in VR. Extended reality training allows to fully experience and iterate on a virtual task before committing to the physical workplace, thus alleviating resources and ensuring a safe environment, both for trainees and equipment. Moreover, virtual interactions facilitate the granular adjustment of conditions based on personalized preferences and needs and can also become a valuable means of data gathering for ergonomic assessment and user-centered design (linked with WP2, WP3). However, the development of XR platforms that can support all different aspects of industrial training within a single system is extremely challenging. In this deliverable we presented (i) an XR-based platform for lifelong learning that can help a worker adapt to changes in the industrial environment through corresponding ICT-empowered training processes. These tools consist of components that support the creation and execution of a tutorial process in virtual environments. The software implementation was performed in the Unity3D real time development platform, so the tutorial environment can be encoded in Unity Asset Bundles which are archive files that contain platform specific assets (such as models, textures, audio clips) that can be loaded by Unity at run time. (ii) Moreover, we have developed an AR-based user training toolkit for the automotive pillar, that includes simulation of driving in a virtual city while receiving notifications of potential dangers, increasing situational awareness. A small user study allowed to assess technology acceptance and perceived usefulness exposing the efficacy and potential of the developed AR-based CPHS user training toolkit.

The XR training toolkit that has been developed is also used in D3.4 for the pose recognition analysis and the visualization techniques for the situational awareness of the operators (manufacturing pillar). Additionally, the usability of the driving simulator is discussed in more detail in D3.4 (automotive pillar).

Appendix A

SURVEYS Acceptance Study-AR/VR tools

The aim of the present survey is to assess the acceptability and usefulness of developed technological solutions for you as an operator, aiming at your health and well being, as well as promotion of your training and collaboration.

Short videos will also be presented to you to better understand the technological solutions. This survey is focused on the following tool:

-The Virtual Reality/Augmented Reality tools

* It is required

BASELINE SURVEYS

In this part of the survey we will ask you basic information about yourself, your way of relating to technology and your job position.

PERSONAL INFORMATION

1. GENDER *

- Female
 Male

2. AGE *

3. HIGHEST LEVEL OF EDUCATION YOU HAVE COMPLETED *

- Primary School
 Secondary School
 University
 Masters
 Doctorate
 None

Other _____

FREQUENCY IN TECHNOLOGY USE

4. What kind of apps do you use more frequently? *

	Never	Sometimes	Usually	Daily
Games or entertainment apps	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Social Media (e.g. Facebook, Instagram)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Learning/training apps or tutorials	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Safety and wellbeing	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

5. If you use other kind of apps, please indicate here

6. Have you ever used Augmented Reality devices or tools? *

- Never
- Sometimes
- Usually

7. Have you ever used Virtual Reality devices or tools? *

- Never
- Sometimes
- Usually

Situational awareness for cooperative driving applications in AR/VR

The driving simulator is able to create and provide different realistic traffic scenarios and different displaying techniques for the visualization of the virtual information in AR displaying devices for situational awareness of the driver.

Set up of the wheeling chair and the AR display device



TECHNOLOGY ACCEPTANCE

Please indicate your level of agreement with the following statements.

8. Perceived Ease of use Questions (PEU) *

	Extremely agree	Slightly agree	Neither	Slightly disagree	Extremely disagree
1. Learning to use the driving simulator tool would be easy for you	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2. Your interaction with the driving simulator tool would be easy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3. It would be easy for you to become skilful at using this tool	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

9. Perceived Usefulness Questions (PU) *

	Extremely agree	Slightly agree	Neither	Slightly disagree	Extremely disagree
4. Using the proposed visualization system would enable you to drive more safe and being aware about critical upcoming events	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5. Using the proposed visualization system makes you feel less nervous about driving in an unknown area	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6. You would find this tool useful when you have to drive in a non familiar mixed traffic environment (including vehicle, road users, bicyclists, etc))	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

10. Behavioural Intention to Use (BI) *

	Extremely agree	Slightly agree	Neither	Slightly disagree	Extremely disagree
7. If you could use this tool to drive more safe and secure, you would appreciate use it	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

11. Are you aware of VR/AR training tools? *

- Yes, I know
- Have heard before
- No

12. You would be interested in using VR/AR as a training/learning tool? *

- No
- Yes

13. Would you think that the VR/AR tools could help you to drive more safe and being well aware about the surrounding traffic environment? *

- I will definitely be engaged in such tools
- They sound appealing to me
- I prefer the traditional training methods
- I would not like to try them

Display Techniques

Please evaluate your preference to the following display techniques (1 non satisfying - 5 very satisfying)

14. Arrows 3D *



1 2 3 4 5

15. Minimap *



1 2 3 4 5

16. Radar *



1 2 3 4 5

17. Wedge 3D *



1 2 3 4 5

18. Meshes As is *



1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

19. Meshes Spheres *



1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

PRIVACY AND SECURITY QUESTIONNAIRE

Der Fragebogen zu Datenschutz und Sicherheit basiert auf einer gründlichen Untersuchung relevanter Fragebögen/ Umfragen: Chignell, M., Gwizdka, J., & Quan-Haase, A. (2003). Der Fragebogen zur Einstellung zum Datenschutz (PAQ) und die Internetnutzung. In AoIR 2003 Annual Conference (S. 16-19). https://www.cc.gatech.edu/gvu/user_surveys/survey-1998-10/questions/privacy.html

20. Security System *

	Not at all concerned	A little concerned	Somewhat concerned	Very concerned
3. Having in mind that secure policies have been defined as part of the project to protect your privacy and confidentiality, how much concerned are you about VR and AR technologies use for?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4. Having in mind that authentication and authorization will be used strictly according to knowledge exchange through VR/AR, how much concerned are you about your data security?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>