

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Spatiotemporal Task Planning Visualization in Evolving Large 3D Models using Temporal Focus + Context

João Filipe Carvalhais dos Santos de Matos



Mestrado em Engenharia Informática e Computação

Supervisor: Prof. Daniel Castro Silva

Second Supervisor: Prof. Alexandre Valle de Carvalho

Supervisor at CERN: Fernando Baltasar dos Santos Pedrosa

July 31, 2025

Spatiotemporal Task Planning Visualization in Evolving Large 3D Models using Temporal Focus + Context

João Filipe Carvalhais dos Santos de Matos

Mestrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

President: Prof. António Augusto de Sousa

External Examiner: Prof. Fernando Penousal Machado

Supervisor: Prof. Daniel Castro Silva

July 31, 2025

Resumo

A manutenção e evolução de instalações científicas complexas, como as do CERN, exigem um planeamento, coordenação e análise meticolosos. A necessidade de utilizar ferramentas de visualização avançadas que possam ilustrar dinamicamente alterações na estrutura e no agendamento de tarefas é cada vez mais crítica. Os sistemas de visualização 3D estáticos ou espacialmente fixos atualmente em uso no CERN são inestimáveis para compreender a disposição das máquinas e as tarefas de manutenção em curso, mas revelam-se insuficientes quando se trata de representar a evolução temporal.

A presente dissertação propõe e valida uma nova metodologia para aprimorar uma ferramenta para a visualização 3D na web, no CERN, permitindo a representação tanto dos estados atuais como históricos (e também planeados para o futuro) das instalações científicas. Central para esta solução é a adaptação e integração de uma árvore quadridimensional como estrutura de indexação espaciotemporal, que permite a gestão e recuperação de versões de modelos de máquinas e dos seus componentes ao longo do tempo. Esta estrutura associa cada componente do modelo a intervalos de tempo explícitos, garantindo reconstruções precisas das configurações das instalações para qualquer data selecionada dentro do horizonte de planeamento de manutenção. Além disso, um mecanismo de Grau de Interesse Temporal é incorporado no processo por etapas de visualização para ajustar elementos visuais, como cor e opacidade, com base na relevância temporal das atividades de manutenção: instalações, remoções e tarefas em curso. Esta abordagem foi concebida para comunicar o contexto temporal de forma intuitiva, ajudando os utilizadores a distinguir entre intervenções passadas, atuais e futuras com um esforço cognitivo mínimo.

Um sistema protótipo foi desenvolvido e implementado utilizando modelos reais e dados de teste de tarefas de manutenção das instalações TCC2 no CERN. Otimizações para modelos grandes e complexos, incluindo simplificação geométrica e seleção interativa, foram implementadas para manter a usabilidade e a fidelidade visual, mesmo com milhões de primitivas geométricas. O protótipo foi avaliado através de testes com utilizadores do CERN, incluindo engenheiros e planeadores, que realizaram tarefas de análise com a ferramenta. Os resultados, obtidos a partir de uma combinação de avaliações baseadas em tarefas, questionários de usabilidade e feedback qualitativo, demonstraram a eficácia da ferramenta em clarificar informações temporais e apoiar fluxos de trabalho de manutenção complexos. De forma crucial, os utilizadores relataram uma melhoria na compreensão das alterações e a possibilidade de facilitar as atividades de planeamento para futuras intervenções. Este trabalho avança o estado da arte na visualização 3D industrial, combinando estruturas de dados espaciotemporais com técnicas de visualização baseadas na perceção.

A abordagem desenvolvida oferece uma solução escalável, moderadamente eficiente e centrada no utilizador, adaptável a futuras expansões no CERN e a ambientes técnicos de grande escala e em evolução semelhantes.

Palavras-chave: Visualização de Informação, Modelo 3D Evolutivo, Dados Espaciotemporais, Foco + Contexto Temporais, Sequenciamento de Construção

Abstract

The maintenance and evolution of complex scientific facilities, such as those at CERN, demand meticulous planning, coordination, and analysis. The need for advanced visualization tools that can dynamically illustrate changes in structure and task scheduling is increasingly critical. Traditional static or spatially-fixed 3D visualization systems currently in use at CERN are invaluable for understanding machine layouts and ongoing maintenance tasks, but they fall short when tasked with representing temporal evolution.

This dissertation proposes and validates a novel methodology for enhancing a web-based 3D visualization system at CERN, enabling the depiction of both current and historical (as well as planned future) states of scientific facilities. Central to this solution is the adaptation and integration of a four-dimensional KD-Tree as a spatiotemporal indexing structure, allowing the management and retrieval of machine model versions and their components over time. This structure associates each model component with explicit time intervals, ensuring accurate reconstructions of facility configurations for any selected date within the maintenance planning horizon. In addition, a Temporal Degree of Interest mechanism is incorporated into the visualization pipeline to adjust visual encodings, such as color and opacity, based on the temporal relevance of maintenance activities: installations, removals, and ongoing tasks. This approach is designed to communicate temporal context intuitively, helping users distinguish between past, current, and upcoming interventions with minimal cognitive load.

A prototype system was developed and deployed using real models and test maintenance task data from the TCC2 facility at CERN. Optimizations for large and complex models, including geometry simplification and interactive selection, were implemented to maintain usability and visual fidelity even with millions of geometric primitives. The prototype was evaluated through user testing with CERN staff, including engineers and planners, who performed analysis tasks with the tool. Results from a combination of task-based evaluations, usability questionnaires (such as the System Usability Scale), and qualitative feedback demonstrated the tool's effectiveness in clarifying temporal information and supporting complex maintenance workflows. Crucially, users reported improved comprehension of changes and the possibility of easing planning activities for future interventions. This work advances the state of the art in industrial 3D visualization by combining spatiotemporal data structures with perception-based visualization techniques.

The approach developed herein offers a scalable, modestly efficient, and user-centered solution adaptable to future expansions at CERN and to similar large-scale, evolving technical environments.

Keywords: Information Visualization, Evolving 3D Model, Spatiotemporal Data, Temporal Focus + Context, Construction Sequencing

Acknowledgments

I am very grateful to my supervisors Prof. Daniel Castro Silva and Prof. Alexandre Valle for their guidance, their patience and for the time they spent with me. Without their expertise the end result would not have been as successful and the whole path would have been much harder.

My gratitude is extended to Fernando Pedrosa for his kindness and supervision before and during my stay at CERN. Likewise, many thanks to Afonso Caiado for his readiness to help and for welcoming me in this new environment. I also sincerely appreciate all of the people who I had the pleasure of spending time together with during this experience. It was great!

To my family and friends, your kindness is always cherished.

João Carvalhais de Matos

“Time is a sort of river of passing events, and strong is its current; no sooner is a thing brought to sight than it is swept by and another takes its place, and this too will be swept away.”

Marcus Aurelius

Contents

Resumo	i
Abstract	ii
Acknowledgments	iii
1 Introduction	1
1.1 Context	1
1.2 Motivation	2
1.3 Problem	3
1.4 Objectives	4
1.5 Document Structure	5
2 Background	6
2.1 CERN's Task Planning and Visualization Platform	6
2.1.1 Work at CERN	6
2.1.2 Task Planning	7
2.1.3 3D Web Visualization Tool	8
2.2 Fundamentals and Evolution of 3D Visualization Technologies	9
2.2.1 Evolution of 3D Data Visualizations	11
2.2.2 Overview of Computer Graphics Concepts	13
2.2.3 Overview of CAD and BIM	14
3 Related Work	17
3.1 Systematic Review Process	17
3.1.1 Identification Phase	17
3.1.2 Screening Phase	19
3.1.3 Review and Analysis	22
3.2 State of the Art	23
3.2.1 Visualization and Interaction with 3D Models	24
3.2.2 4D Models Structure and Representation	29
3.2.3 Visualization techniques for time-evolving data	37
4 Problem Approach	49
4.1 Detailed Problem Definition	49
4.2 Proposed Solution	49
4.3 Risk Analysis	52

5	Implementation	54
5.1	Model Preprocessing	54
5.2	Optimizations	57
5.3	Temporal Models	58
5.4	Temporal Degree of Interest	60
5.5	Interactivity and Graphical User Interface	67
5.6	Challenges	67
5.7	Summary	68
6	Experimental Protocol	70
6.1	Test Methodology	70
6.1.1	Overview	70
6.1.2	Experiment Procedure	72
6.2	Assessment Methods	73
6.2.1	Pre-test Questionnaire	73
6.2.2	Task-specific Questionnaire	73
6.2.3	Post-test Questionnaire	75
6.3	Pilot Test	75
6.4	Summary	76
7	Results	77
7.1	Sample Characterization	77
7.2	Task Performance Results	77
7.2.1	Individual Task Analysis	77
7.2.2	Distribution of Correct Answers	85
7.2.3	User Preferences	85
7.3	SUS Questionnaire Results	86
7.4	Correlation With Demographics	86
7.5	Feedback	89
7.6	Summary	91
8	Conclusions	92
8.1	Final Remarks	92
8.2	Future Work	94
	Bibliography	97
A	User Testing Script	106
A.1	Introduction	106
A.2	Informed Consent Form	106
A.2.1	Introduction:	106
A.2.2	Voluntary Participation:	106
A.2.3	Experiment Overview:	107
A.2.4	Data Collection:	107
A.2.5	Data Usage and Confidentiality:	107
A.2.6	Contact Information:	107
A.2.7	Consent :	107
A.3	Pre-test Questionnaire	107
A.4	Task-specific Questionnaire	108

A.4.1	Task 2.1	108
A.4.2	Task 2.2	109
A.4.3	Task 2.3	110
A.4.4	Task 2.4: Options Feedback	110
A.5	Post-test Questionnaire	111
A.5.1	System Usability Scale (SUS)	111
A.5.2	Open Feedback	111
B	Questionnaire Answers	112

List of Figures

2.1	Sample of colors representing different groups that perform maintenance tasks at CERN.	8
2.2	Visualization of task on the PS machine using the 2D visualization tool.	9
2.3	Global task view for the PS machine in the 3D tool.	10
2.4	Detailed section task view for the first octant of the PS machine in the 3D tool. The model contains 107,248 triangles.	10
2.5	Detailed task information for the PS machine in the 3D tool.	11
3.1	Flowchart according to PRISMA methodology.	18
3.2	Publication year distribution of selected records.	23
3.3	A Gimlens magnifying the interior of a car engine.	25
3.4	Peeling techniques.	25
3.5	Combined 2D/3D view with brushes.	26
3.6	The four spatio-temporal visualizations compared in study: (a) Embedded View, (b) Linked View, (c) Temporal Juxtaposition, and (d) Spatial Juxtaposition. . . .	27
3.7	Naïve and SAH approaches to the construction of kD-trees with depth 3 for a 2 dimensional dataset.	31
3.8	Representations of the STPM. (a) exemplifies the visual result while (b) focuses on the internal representation.	32
3.9	Gen4DCity: Generic conceptual model for delivering time-evolving 3D city models on the web based on standard formats. The entities of Gen3DGeo are depicted in white. The proposed extension of the temporal dimension is depicted in gray. .	33
3.10	Temporal navigation comparison between 3D Tiles and 3D Tiles with temporal extension.	34
3.11	Charles Minard’s 1869 chart showing the number of men in Napoleon’s 1812 Russian campaign army, their movements, as well as the temperature they encountered on the return path.	38
3.12	Application of TDOI to political map of Europe.	39
3.13	Overview of spatial 3D comparison approaches, using just two data instances (a–d) and N instances (e–h).	40
3.14	The colored time flattening operation.	42
3.15	An example of the STC integration with the use case of La Doua district (Lyon/France). All transactions types are displayed in the 3D view.	43
3.16	With a view frustum spanning a time difference, pop-up artifacts are moderated by fade in the objects.	44
4.1	General architecture of the proposed web visualization tool.	51
4.2	Information visualization pipeline of the proposed system.	52

5.1	New preprocessing pipeline.	54
5.2	TCC2 models before and after simplification.	55
5.3	TCC2 exterior simplified with a 10% target using GLTFPack.	56
5.4	Steps performed to combine model parts (exterior and interior) with different simplification levels.	57
5.5	New timeline, vertically expanded to avoid overlaps.	61
5.6	Developed Temporal Degree of Interest mechanisms.	64
5.7	Summary of visual status of equipment according to the Temporal Degree of Interest.	66
5.8	Rendering loop.	66
5.9	Visualization pipeline summary.	69
7.1	Histogram of the ages of the participants.	78
7.2	Level of experience using software to manipulate or visualize 3D models.	78
7.3	Number of correct answers for each task.	79
7.4	Tasks 1 to 3 — Highlight of task type markers on the timeline tasks.	79
7.5	Task 4 — Ongoing maintenance task.	80
7.6	Task 5 — After maintenance and before another maintenance activity.	80
7.7	Task 6 — Ongoing removal task.	81
7.8	Task 7 — Before removal task.	81
7.9	Task 8 — After removal task.	82
7.10	Task 9 — Before removal task.	82
7.11	Task 10 — Before installation task.	83
7.12	Tasks 11 to 13 — In order from left to right, after, ongoing and before installation tasks.	83
7.14	Task 15 — 3 installation tasks starting in the selected period.	84
7.13	Task 14 — After removal and before reinstallation.	84
7.15	Distribution of the number of correct answers in a total of 15 tasks.	85
7.16	Distribution of SUS Scores	87
7.17	Average number of correct answers by age bracket.	88
7.18	Average number of correct answers by experience level with 3D interfaces.	88
7.19	Average number of correct answers by gender.	89
8.1	Visualization tool with improvements: ability to hide walls, legend.	94
8.2	Summary of the selected tasks on the timeline and timeline items outlines.	95
8.3	Help menu with tabs, sections and media (placeholder).	95

List of Tables

3.1	Number of results per digital library for the query string.	19
3.2	Inclusion criteria used for selecting records collected from the digital libraries. . .	20
3.3	Overview of selected records.	20
3.4	Number of reviewed publications by type.	22
3.5	Reviewed records related to 3D visualization and interaction.	28
3.6	Overview of spatio-temporal modeling approaches.	35
3.7	Reviewed records related to the structure and representation of 3D/4D models. . .	36
3.8	Reviewed records related to 4D visualization techniques.	46
3.9	4D visualization techniques from the selected records.	47
3.10	Summary of visualization techniques for time-evolving data	48
4.1	Summary of likelihood and impact of each identified risk.	53
B.1	Questionnaire answers excluding open-ended questions (Part 1: Demographic In- formation).	113
B.2	Questionnaire answers excluding open-ended questions (Part 2: Tasks 1 to 3). . .	114
B.3	Questionnaire answers excluding open-ended questions (Part 3: Tasks 4 to 9). . .	115
B.4	Questionnaire answers excluding open-ended questions (Part 4: Tasks 10 to 15). .	116
B.5	Questionnaire answers excluding open-ended questions (Part 5: Visual Preferences).117	
B.6	Questionnaire answers excluding open-ended questions (Part 6: Temporal Degree of Interest preference and initial settings).	118
B.7	Questionnaire answers excluding open-ended questions (Part 7: System Usability Scale).	119

Abbreviations and Symbols

2D	Two-dimensional
3D	Three-dimensional
4D	Four-dimensional
ACE	Accelerator Coordination and Engineering
ANOVA	Analysis of Variance
ATS	Accelerators and Technology Sector
API	Advanced Programming Interface
AR	Augmented Reality
BIM	Building Information Modeling
CAD	Computer-Aided Design
CERN	Conseil Européen pour la Recherche Nucléaire
CG	Computer Graphics
CPU	Central Processing Unit
CSG	Constructive Solid Geometry
DS	Data Structure
EN	Engineering (Department)
FPS	Frames-Per-Second
GIS	Geographic Information System
glb	Binary Graphics Language Transmission Format
gltf	Graphics Language Transmission Format
GPU	Graphics Processing Unit
GUI	Graphical User Interface
JSON	JavaScript Object Notation
LOD	Level of Detail
LHC	Large Hadron Collider
LINAC3	Linear Accelerator 3
LINAC4	Linear Accelerator 4
LS	Long Shutdown
LS3	Long Shutdown 3
Mbps	megabits per second
MS	Microsoft
OSS	Organisation, Scheduling and Support
PCA	Principal Component Analysis
PS	Proton Synchrotron
RAM	Random Access Memory
SAH	Surface Area Heuristic
SPS	Super Proton Synchrotron
STEP	Standard for the Exchange of Product Data
SUS	System Usability Scale
TDOI	Temporal Degree of Interest
VR	Virtual Reality
YETS	Year-End Technical Stop

Chapter 1

Introduction

Three-dimensional visualizations play a crucial role in science and engineering, enabling the exploration and analysis of complex systems in an intuitive manner (Teplá et al., 2022). These visual representations help convey structural, functional, and temporal aspects of models, making them valuable tools for research, design, and maintenance.

As technology advances, the need for more sophisticated visualization techniques increases, particularly in environments where systems evolve over time. Representing changes dynamically, rather than as static snapshots, can provide deeper insights, improve decision-making processes, and offer a more comprehensive understanding of complex systems.

This dissertation aims to study and propose an innovative approach to enhancing a web-based 3D visualization system at CERN, aiming to improve how evolving structures are represented and understood. The following sections outline the motivation behind this study and the key questions it seeks to address. At the end, the overall structure of this document is presented.

1.1 Context

CERN¹, the European Organization for Nuclear Research, is a scientific laboratory that sits astride the Franco-Swiss border near Geneva. There, physicists and engineers create and use the world's largest and most complex scientific instruments to study the basic constituents of matter by accelerating particles and making them collide at close to the speed of light. Other activities are performed there, such as experiments in nuclear to high-energy physics, and studies ranging from antimatter to the possible effects of cosmic rays on clouds. This is only possible with the use of very specialized facilities, some of which are hundreds or thousands of meters long, such as the Proton Synchrotron (PS)² and the Large Hadron Collider (LHC)³.

The planning tasks associated with the maintenance activities required to ensure the continued operation of these machines are highly complex. Given the intricate nature of the machinery, it is essential to accurately extract and interpret the associated information, particularly details about

¹<https://home.cern>

²<https://home.cern/science/accelerators/proton-synchrotron>

³<https://home.cern/science/accelerators/large-hadron-collider>

ongoing and upcoming maintenance activities. Usually, these operations take days or weeks to finish and span different machine sections, and more than one group can be involved in maintenance tasks in a section at the same time (Nunes, 2023). However, periodically, these machines and facilities are subjected to "year-end technical stops" (YETS) for repairs and upgrades that run for a few months. After years of operation, they are also made to undergo more extended periods of maintenance known as "Long Shutdowns", which can last for years.

Visualizing this data can help teams better understand and manage the complexity of these operations, highlighting the importance of data visualization (Friendly, 2008). Maintenance projects can involve multiple teams that must be well coordinated to ensure smooth execution. In this context, a visualization tool that represents components and activities over time can be invaluable for organizing tasks, improving efficiency, and enhancing decision-making.

1.2 Motivation

Prior to 2023, CERN used a web-based solution called "Scheduling Tools" designed to, among other matters, help visualize scheduled maintenance tasks for machine components. The website offers some viewing options, including: Gantt View, Calendar View and 2D Top View, which present a 2D layout of the machine, where sections of the machine are color-coded based on the maintenance groups scheduled to work on them for a selected day.

To overcome some shortcomings with the 2D tool, in 2023, a 3D visualization system was created and integrated into the Scheduling Tools website to be able to view and distinguish the sections better as well as acquire the wanted information faster (Nunes, 2023). The 3D tool was prototyped on the PS accelerator and was successfully extended to work with the Linear Accelerator 4 facility (LINAC4)⁴. There is interest in integrating other facilities in the system, like the LHC, which is the biggest. Not only that, but the machines (and their respective models, consequently) can suffer modifications throughout time. An improved version of the system should allow the visualization of that evolution, while also exploring ways to upgrade the current visualization of tasks and machine information.

Currently, even if there are activities planned for a future date in a facility, it may not be easy to perceive the configuration of the facility after the activities, as components may have been changed. In fact, even to comprehend the current state of some machine, the available options are to maintain a manually updated integration model following specific activities or to conduct an on-site visit to verify the conditions.

The working time of CERN researchers is limited and precious, so having to focus on trivial assignment tasks when planning future experiments reduces their availability to analyze previous experiments or perform new ones. Poorly or incorrectly planned operations can also lead to delays and wasted time. More efficient planning activities can lead to a potentially greater scientific output. To that end, it is valuable to consolidate all relevant information about a machine and its tasks in one place, eliminating the need to cross-reference data from multiple sources.

⁴<https://home.cern/science/accelerators/linear-accelerator-4>

In addition to this short to medium-term vision of task planning and execution, a long-term vision created by the analysis of the evolution of a machine and its related tasks can be useful to spot patterns, relationships, and trends. Visualizing the process of dismantling or adding machine parts as they are occurring can also prove helpful to plan experiments and maintenance tasks during such periods.

1.3 Problem

While the current 3D system already works for three facilities, it is expected that the inclusion of bigger ones like the LHC, which has a perimeter more than 40 times larger than that of the PS, would pose other challenges as its model is much bigger and more complex. The solution to a systematic approach for visualizing models and their information over time must ensure that the information to be displayed matches the structure of the model at a given time.

This is not a trivial challenge, as sections of the machine may have been organized differently in the past: some could have been removed, replaced, or merged into other sections in the present-day model (Fouché et al., 2023). For instance, a structural component that once housed multiple subsystems might have been split into separate units over time. In addition, trade-offs must be carefully evaluated between different alternatives. Latency plays a crucial role, as a system that updates visual data in real time provides immediate feedback but may require significant computational resources. Another key aspect is the ease of manipulation; while a highly detailed model offers precision, it can also become challenging to navigate or modify, requiring a balance between detail and usability. Spatial synchronization presents another challenge, as different components may be modified independently, requiring careful alignment to maintain coherence in the overall layout. Finally, business rules must be considered, as they define constraints and dependencies within the system. For example, certain maintenance procedures may require specific sequences of operations, preventing certain modifications from being visualized before prerequisite tasks are completed.

Modifications to a facility can take anywhere from days to years to complete. To maintain accuracy during these transitions, the model must reflect changes as they happen, recognizing that some parts may gradually disappear while others are added. This becomes especially complex when both removals and additions occur within the same section. For example, if a section is being replaced in phases, the visualization must simultaneously show the decommissioned and newly installed segments, ensuring a coherent representation throughout the transition.

This is, in fact, simply a matter of visualizing the evolution of large 3D models and not only at CERN. However, this work must meet the needs and requirements of CERN which should not limit its adaptation to similar contexts.

1.4 Objectives

The main goal of this work is to propose and validate a new approach for the web-based visualization tool used at CERN to include the temporal evolution of the machines and integrated components. Thus, the model displayed when seeing tasks for a selected date reflects a specific configuration of the model and its components, which may be different for another date. The visualization of the tasks, which themselves are associated with sections or equipment of a facility, must accompany the evolution of the model of the facility. This objective raises the following research questions:

- **RQ1:** How can 3D models representing parts of a facility at a specific time, be identified and indexed to ensure accurate and efficient reconstruction of the complete machine from its components?
- **RQ2:** What are the most effective techniques for retrieving and assembling large 3D models with varying configurations and levels of detail from a model storage system to construct a scene, ensuring low latency?
- **RQ3:** How can the visualization of partial 3D models, which may overlap and evolve over time, be perceptible and convey information about ongoing maintenance tasks?

These objectives assume large 3D models to represent structures as large as the facilities at CERN and that, having already been simplified for usage on the web, can contain hundreds of thousands or a few million vertices and dozens or even hundreds of megabytes in file size. As for low latency and information perceptibility, a response time of less than 1 second is considered acceptable, as this threshold ensures that the user's thought flow remains uninterrupted, according to [Nielsen \(1994\)](#). This is achievable with a stable Internet connection averaging 100 Mbps download speed. Additionally, the user should not feel uncertain about the meaning of the visual elements or the status of the machines during interaction. The model storage system mentioned in the second question directly refers to the one devised in the first question.

To fulfill these requirements and objectives, this work proposes a novel approach to the visualization of the temporal evolution of complex machines, specifically using a four-dimensional KD-Tree, adapted to manage the temporal evolution of the models ([Jaillet, 2020](#)), combined with a Temporal Degree of Interest (TDOI) ([Carvalho et al., 2008](#)) to display changes over time. The methodology is based on the analysis of models of the TCC2 facility before and after its third period of long shutdown (LS3) to ensure an accurate representation of the changes in the facility as the maintenance tasks are planned and executed. The solution is evaluated through user testing at CERN, where engineers and maintenance teams interact with the system to assess its effectiveness in supporting their tasks. The validation process considers factors such as usability, clarity of temporal representations, and overall benefits in planning and decision-making.

This dissertation contributes to the field of 3D visualization by integrating a TDOI with a 4D-Tree approach to efficiently manage and represent model modifications over time in an industrial

setting. The combination of these techniques provides a structured way to organize both static and evolving components, offering a scalable solution for complex environments like the one at CERN. By improving the accessibility and comprehension of historical, current, and future system configurations, this work aims to enhance maintenance workflows and facilitate better decision-making for engineers working with large-scale scientific machinery.

1.5 Document Structure

Beyond the introduction, this document contains 7 additional chapters. In Chapter 2, more context regarding the operation of CERN is given as well as some notions on basic Computer Graphics topics such as the visualization and interaction with 3D models. Chapter 3 provides a literature review with a brief discussion of related works, along with a more in-depth examination of the state-of-the-art, focusing on the most relevant studies. Chapter 4 outlines a proposed approach to address the objectives of the dissertation, including a timeline and an assessment of associated risks. Chapter 5 describes the main functionalities that were developed as part of a prototype that can be used at CERN. Chapter 6 delineates the protocol used to test the prototype in individual testing sessions. In chapter 7, the results of the tests are explored and discussed. In chapter 8, an overall summary of the work developed and the main takeaways are presented, along with conclusions drawn from the findings and implications for future research or applications.

Chapter 2

Background

To better understand the context and work of this dissertation, it is essential to explore some details regarding both the environment where the work is conducted and the technical foundations that support the proposed solution. This chapter provides an overview of these aspects, offering background information that will aid in comprehending the challenges and methodologies discussed in later chapters.

The first section focuses on the activities at CERN and its organization, detailing how maintenance tasks are planned and executed. It also presents the existing 2D and 3D tools used for visualizing maintenance tasks. The second section provides a historical perspective on 3D interaction and visualization, introducing key concepts from Computer Graphics, such as Levels of Detail (LODs), and discussing the relevance of fields like CAD and BIM modeling.

2.1 CERN’s Task Planning and Visualization Platform

CERN is a unique organization where its structure and workflow play a crucial role in ensuring efficiency in the planning and execution of scientific activities, especially those that require the use of complex machinery. Maintenance tasks also require careful planning and coordination among multiple teams, relying on detailed schedules and structured procedures. To support these operations, various tools are employed, such as traditional 2D schematics and 3D models. These tools help engineers and technicians analyze components, track ongoing and future maintenance activities, and improve overall decision-making.

2.1.1 Work at CERN

CERN employs about 2500 people¹. The organization is partitioned into Sectors, each divided into Departments². The Accelerators and Technology Sector, for example, consists of four Departments: Beams, Engineering, Accelerator Systems, and Technology. Each Department is further divided into various Groups and, finally, into Sections. This dissertation is an initiative proposed

¹<https://home.cern/about/who-we-re>

²<https://directory.web.cern.ch/>

by the Organisation, Scheduling and Support (OSS) section of the Accelerator Coordination and Engineering (ACE) group of the Engineering Department (EN) in the Accelerators and Technology (AT) Sector. The OSS section is responsible for creating and managing the planning of activities, which includes coordinating logistics, overseeing co-activities, and implementing mitigation measures to prevent delays while prioritizing safety (Vergara Fernandez et al., 2023).

CERN is, perhaps, mostly known to conduct research on the field of particle physics by accelerating and colliding particles at close to the speed of light inside the Large Hadron Collider. Nevertheless, many other activities are performed there. Some examples are the hunt for hypothetical particles that have been hypothesized but never detected, of which magnetic monopoles are the main illustration; climate experiments that mimic the role cosmic rays play in cloud formation by firing their own "artificial" cosmic rays (created in the Proton Synchrotron) into an artificial atmosphere; or the study of antimatter (also created in the Proton Synchrotron) by trying to stabilize antimatter particles created by collisions for time enough to study it (Space.com, 2024). These activities can originate scientific and technological advancements with a positive impact on different industries and on society. The best-known example of technology birthed at CERN is the World Wide Web. Other relevant example is that, nowadays, most particle accelerators and detectors are not used for research but in biomedical machines for diagnosis and therapy, and in computer chip manufacturing. The tools and knowledge developed at CERN accelerate progress in many areas of science and industry³.

These advancements highlight CERN's broader impact beyond fundamental physics, showcasing how its cutting-edge machines and accelerators drive innovation across multiple disciplines. CERN's accelerators undergo multiple experiments annually, and they require regular maintenance and calibration to ensure optimal performance and accuracy. Thus, it is important to plan maintenance tasks effectively, so as to not delay any experiments.

2.1.2 Task Planning

At CERN, the Scheduling Tools platform contains a 2D task visualization tool for 5 facilities. To see the planned tasks for each one, CERN employees can use the visualization menu that provides a 2D top-view image of the selected facility. There are other visualizations such as Gantt and calendar views that include every facility. Each maintenance task is performed by at least one group, therefore, the sections of the machines are painted with the color of the group performing the task, according to a specific color scheme. Figure 2.1 exhibits some examples of colors and associated group code, including the color used for multiple groups. A special color indicates that multiple groups are performing tasks over the same section. Figure 2.2 displays the 2D top-view for the PS accelerator with some color-coded tasks. It must be noted that the tasks are scheduled separately using MS Project⁴, which is periodically synchronized (unidirectionally) with the Scheduling Tools database. The 2D view provides a useful global overview of task positions, but

³<https://home.web.cern.ch/about/what-we-do/our-impact>

⁴<https://www.microsoft.com/en-us/microsoft-365/project/project-management-software>

BE-ABP #ffc000		BE-ASR #52de1d		BE-BI #bbb600		BE-EA #ffc000	
BE-CO #ffc000		BE-GM #ffc000		Multiple #3db9d3		None #ffffff	

Figure 2.1: Sample of colors representing different groups that perform maintenance tasks at CERN.

it lacks detailed information and fine-grained control over the visualization. CERN sought to enhance the 2D view by integrating a third dimension, allowing for improved task visualization and increased user interaction (Nunes, 2023).

Meticulous planning and well-defined scopes are crucial due to the highly complex and large-scale nature of the facilities. The longer maintenance periods, known as Long Shutdowns (LS), are not frequent but typically last for several years. These shutdowns are necessary for comprehensive repairs, upgrades, and replacements of critical components that are exposed to extreme environmental conditions, such as radiation and sunlight, which can degrade materials over time. LS3 is expected to start in July 2026 and last 47 months; however, not all facilities require operations that extend across the entire period. To prevent failures and ensure safe operation, extensive replacements and inspections must be carried out during these planned shutdowns. Furthermore, certain equipment, such as cryogenics systems, requires significant overhauls at regular intervals, which further justifies the need for lengthy maintenance windows. Given the duration and complexity of these maintenance tasks, proper coordination is paramount. Each task must be carefully scheduled to avoid conflicts and ensure that resources are optimally allocated.

2.1.3 3D Web Visualization Tool

As the work developed in his Master's dissertation, Nunes implemented a 3D visualization tool for the maintenance tasks, that is already in use for the PS, LINAC3 and LINAC4 machines (Nunes, 2023). The original CAD models of the machines were provided by CERN. Objects that were unnecessary to provide task information were removed and the models were converted to the *glb*⁵ format. The converted models were simplified by a process of triangle decimation and removal of invisible triangles. Subsequently a web visualization tool was developed using the Javascript graphics library ThreeJS⁶ to display the converted models and interactively provide task information. The technology stack used in the Scheduling Tools website is a Spring⁷ backend, a React⁸ frontend and an Oracle Database⁹. As there were some dependency conflicts, the 3D tool was

⁵<https://www.khronos.org/Gltf>

⁶<https://threejs.org/>

⁷<https://spring.io/>

⁸<https://react.dev/>

⁹<https://www.oracle.com/database/>

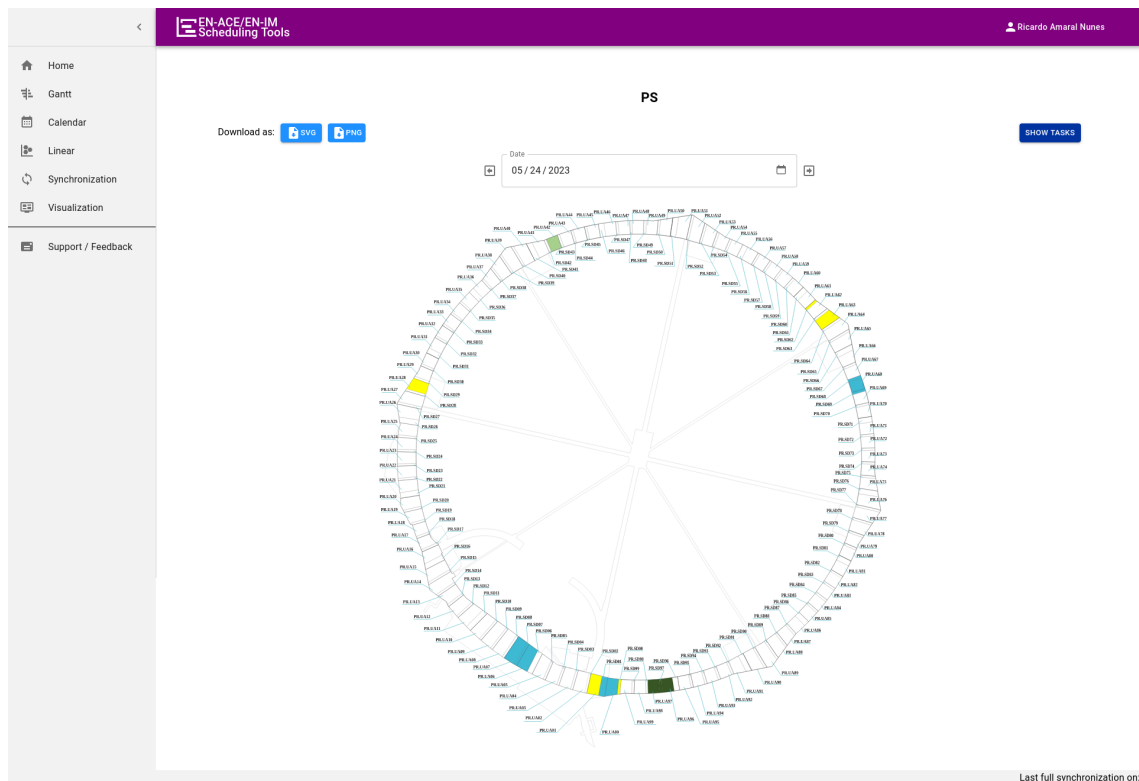


Figure 2.2: Visualization of task on the PS machine using the 2D visualization tool.

integrated as an HTML 'iframe' with the disadvantage that the visualization tool cannot interact directly or utilize information from the rest of the page.

Figures 2.3 and 2.5 show the appearance of the PS machine in the 3D tool featuring a global view and one of its sections, with and without task details. Again, parts of the machine are colored according to the groups that are performing tasks. To provide a clearer sense of the model's complexity, the global PS machine model (center plus all 8 octants) has a file size of 2.26 megabytes and contains less than 20,000 triangles. In contrast, the detailed model of the first octant (Figure 2.4) has a file size of 7.6 megabytes and comprises 107,248 triangles. During the loading process of the detailed model, a low level-of-detail version—approximately one-third of the original file size—is temporarily displayed to improve responsiveness. Performance measurements indicate that the global model achieves an average frame rate of 30 frames per second (FPS). When loading a detailed octant model, the frame rate initially drops to 11 FPS before stabilizing at approximately 25 FPS, as reported in Nunes (2023).

2.2 Fundamentals and Evolution of 3D Visualization Technologies

The development of 3D visualization and interaction techniques has a long history, evolving alongside advancements in Computer Graphics. Fundamental concepts such as Levels of Detail (LODs) allow large and complex models to be rendered efficiently, balancing visual fidelity with performance. In industrial applications, CAD and BIM modeling have become essential for designing,

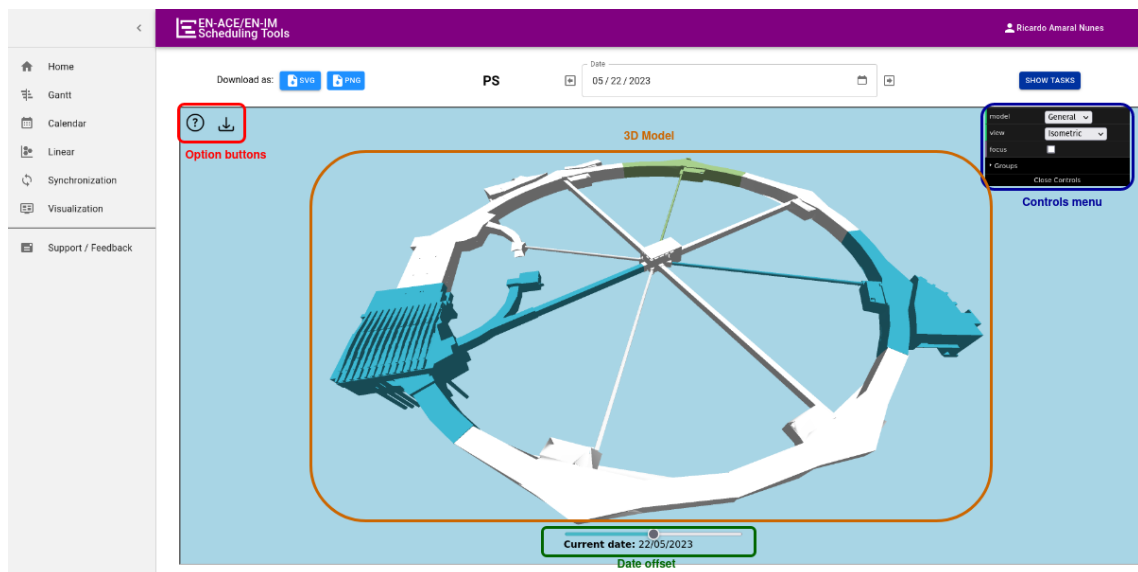


Figure 2.3: Global task view for the PS machine in the 3D tool.

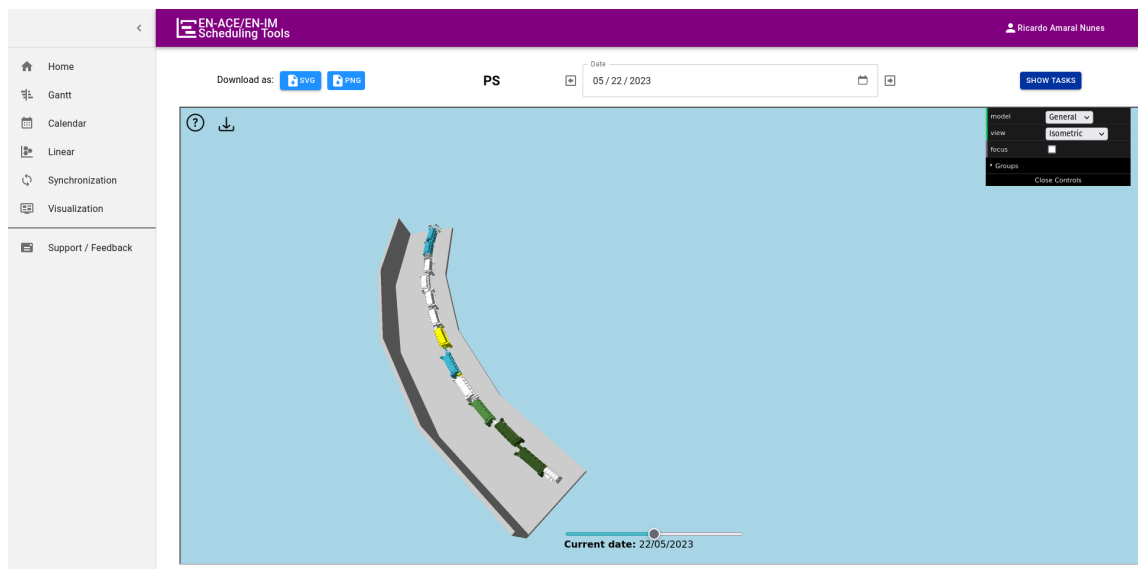


Figure 2.4: Detailed section task view for the first octant of the PS machine in the 3D tool. The model contains 107,248 triangles.

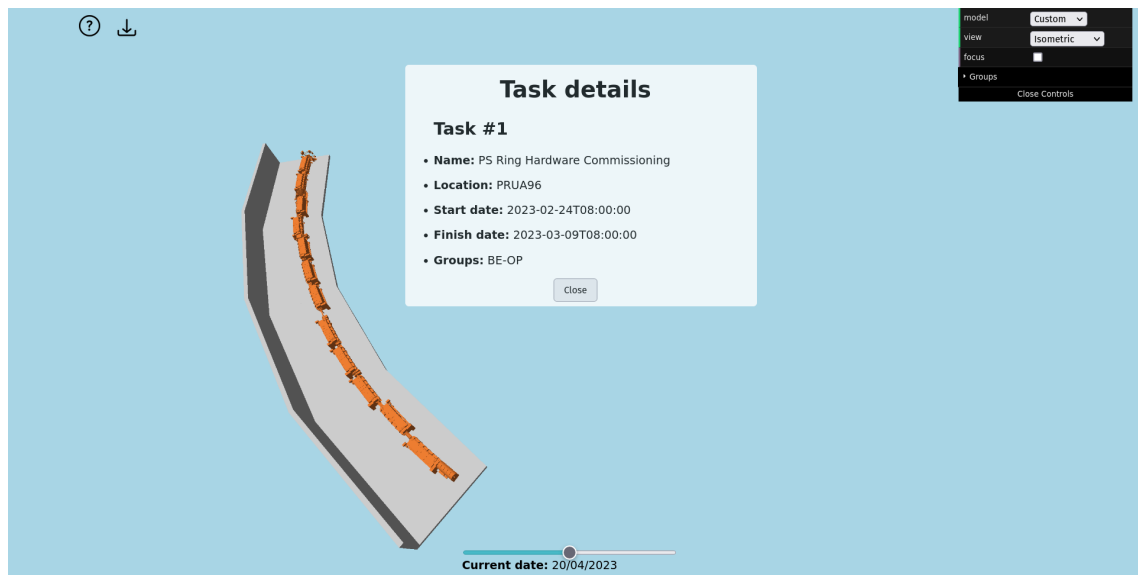


Figure 2.5: Detailed task information for the PS machine in the 3D tool.

managing, and maintaining intricate systems, providing detailed digital representations that support engineering workflows. Understanding these principles and their applications offers valuable insights into the visualization challenges and solutions explored in this dissertation.

2.2.1 Evolution of 3D Data Visualizations

The field of 3D visualization has evolved over the past few decades, having transformed from static representations of data into immersive and interactive experiences. This journey mirrors the broader evolution of data visualization, where growing computational power and new visualization techniques have opened the doors to multidimensional explorations of complex data (Müller and Zeckzer, 2015). The path of 3D visualizations began with basic 3D representations and continues to advance today with sophisticated modeling and the use of augmented and virtual realities.

When it first emerged in the 1960s, 3D visualization was a mainly static field, where graphical representations were used to model objects, environments, and abstract data (University of Washington, 2024). These static models were mostly constrained by the rendering power of early computers and provided no additional interaction beyond the original concept. During this phase, the visualization community focused on accurately depicting complex objects or structures in three dimensions, such as architectural models, molecular structures, and geographic data. Common cases included CAD models, which allowed engineers to view 3D renders of parts and assemblies, though yet without featuring real-time interaction. Such visualizations marked a foundational phase, where data was enhanced by a third dimension but confined to static rendering. There were not many interactions with 3D objects at this point, therefore, the medium was largely passive, meaning that users would observe but not manipulate or modify the visualizations. Some limited navigation was possible such as changing the viewpoint or zoom level, but it was not until the advent of interactive 3D that deeper engagement with data became feasible.

As computational power and the accompanying software advanced, 3D visualizations became more interactive. By the late 1980s and early 1990s, software such as Autodesk's 3D Studio (CGPress, 2010) began offering users tools to interact with their models in real time. This introduced fundamental interactions including panning, zooming, and rotating, which allowed users to explore 3D models from multiple perspectives. Such tools became essential in fields such as mechanical engineering, architecture, and Geographic Information Systems (GIS¹⁰), where interacting with spatial data is critical. New use cases began emerging, including navigation (zooming and panning) across vast terrains, such as digital maps, which greatly enhanced spatial understanding and analysis. Another important use case involved selection and modification, where users could select parts of a model in view, inspect details, and make modifications—a major advantage for iterative design processes. This marked a significant shift from passive visualization (viewing models without interacting) to active interaction, where users could manipulate and adjust models in real time.

The next significant evolution in 3D visualization involved the introduction of animations (Jang, 2000). Early software such as Alias Maya, one of the first widely used tools for 3D animation (York University Computing Museum, 2024). By animating data, 3D visualization could communicate temporal changes and dynamic processes, which help comprehend 3D models (Sando et al., 2009) and became invaluable in fields like meteorology, fluid dynamics, and medical imaging. In meteorology alone, for instance, animated 3D models allowed for the study of environmental changes over time, such as tracking weather patterns, mapping ocean currents, and simulating natural disasters. Animations opened the door to temporal exploration, a crucial step for simulations, turning visualizations into predictive and exploratory tools. 3D visualization has introduced multidimensional data visualization, incorporating fourth and fifth dimensions (sometimes even more). 4D visualization often includes time as the extra dimension, while 5D visualizations incorporate additional data dimensions, such as qualitative attributes or multiple time scales. A 3D model with a time variable can be used to view changes over time, which has seen prominent applications in physics, urban modeling, and simulations.

By the 2010s, developments in Virtual Reality (VR) and Augmented Reality (AR) transformed 3D visualization into immersive experiences (Deloitte, 2018), blending real and virtual environments. In VR, users could enter fully realized virtual spaces, while AR overlaid digital elements onto the physical world. These technologies allowed for a deeper, more intuitive interaction with data at the same time that they can be interacting with the real world (in AR), enabling users to manipulate objects in ways that static and even interactive visualizations usually cannot, such as hand-based manipulation or dissecting models. In industries such as urban planning and manufacturing (Eswaran and Bahubalendruni, 2022), digital twins became a core application of VR/AR, allowing users to explore and interact with real-time data models that mirrored their physical counterparts. For example, a factory's digital twin could show the current status of machinery and predict potential failures.

¹⁰<https://www.usgs.gov/faqs/what-a-geographic-information-system-gis>

As 3D visualization has evolved from static representations to immersive experiences, its impact on data interpretation has grown significantly. Where early visualizations were limited by technical constraints and linear perspectives, modern 3D visualizations incorporate rich interactivity and multidimensional analysis, providing a comprehensive and often intuitive view of data. Through each stage of its development — static, interactive, animated, and multidimensional — 3D visualization has become a crucial tool across fields. The journey has seen the field of 3D visualization expand from passive observation to active engagement, from simple navigation to complex manipulation, and from 3D representations to dynamic, multidimensional explorations.

2.2.2 Overview of Computer Graphics Concepts

Computer Graphics (CG) is the field of Computer Science that encompasses the creation, manipulation and rendering of visual content. Its technologies and tools are used in various sectors such as entertainment (digital photography, film, video games), education, scientific visualization, virtual reality, etc (Foley and Hughes, 2014). A key aspect of CG is 3D rendering, which generates images from 3D representations. These representations can take different forms, with two common approaches being polygonal meshes and parametric models—where meshes are discrete and parametric models are continuous. These methods come with their own set of characteristics, influencing how models are visualized, manipulated, and rendered, particularly in real-time environments like web-based systems.

A polygonal mesh is the most common method of representing 3D models in CG. This method breaks down an object's continuous surface into a collection of polygons (typically triangles) that form the 3D shape. Each polygon is defined by vertices, which contain the 3D coordinates that specify the object's shape. Triangular meshes are a specific type of polygonal mesh where the surfaces are composed of triangles. This method is highly popular because triangles are the simplest polygon and can efficiently represent curved surfaces by approximating them with multiple flat triangles. Polygonal meshes are widely used in applications such as game design, animation, and 3D rendering because they allow flexible and detailed representations of complex shapes. However, as the number of polygons (and therefore vertices) increases, so does the complexity of the model.

In contrast to polygonal meshes, parametric modeling relies on mathematical formulas to define shapes. Instead of approximating a surface with polygons, parametric models use equations and parameters to represent geometry. For example, a parametric model of a sphere might be defined by its radius and center point, rather than by hundreds or thousands of triangles. Parametric models are commonly used in CAD (Computer-Aided Design) applications and in contexts where precise control and flexibility are needed, such as engineering design. While parametric models can be more efficient in terms of data storage and modification, they often need to be converted into polygonal meshes when used in real-time 3D environments.

In real-time environments, such as web-based 3D applications powered by libraries like ThreeJS, the number of polygons and vertices in a model has a significant impact on performance. ThreeJS is a popular JavaScript library for rendering 3D graphics on the web, and it relies on the user's

hardware to render and display models in real-time. Models with a large number of polygons (also known as high-poly models) can dramatically increase the computational load during rendering. Each vertex and polygon requires processing (for lighting, shading, and transformations), which can strain both the CPU and GPU, especially in real-time applications. When a model contains a very large number of polygons, this complexity can result in lower frame rates, longer load times and in increased memory usage. These issues require optimization techniques to ensure smooth performance, especially when dealing with real-time interaction in web-based systems.

One of the most effective optimization strategies in 3D graphics is the use of Levels of Detail (LODs). LODs involve creating multiple versions of a 3D model at varying levels of complexity, each with a different polygon count (Heok and Daman, 2004). Typically, a model will have a high-resolution version (with many polygons) for close-up views and progressively lower-resolution versions (with fewer polygons) for views from farther distances. LODs are used to manage computational load and improve performance by ensuring that the most complex version of a model is only displayed when necessary. When an object is far from the camera, the fine details represented by many polygons are not visible, so rendering the high-polygon version is wasteful. Instead, a lower-polygon version can be used, which reduces the processing required without affecting the perceived visual quality.

This approach is essential in real-time 3D applications, such as games, simulations, and web-based systems, where performance is crucial. By switching between different LODs based on the camera's distance from the object, systems can significantly reduce the number of polygons that need to be rendered at any given moment, thus improving frame rates and maintaining smoother interaction. From the user's perspective, the use of LODs often goes unnoticed if implemented correctly. The switch between different levels of detail should happen smoothly, with the system dynamically changing the model based on the viewing distance. To this effect, it is crucial to avoid popping, a visual artifact where the model suddenly changes in appearance as the LOD switches. Advanced LOD techniques use blending and transitions to ensure that the change between detail levels is gradual and seamless (Akenine-Möller et al., 2018).

2.2.3 Overview of CAD and BIM

In the realm of design and engineering, technological advancements have radically transformed how professionals conceptualize, create, and manage projects, over the past decades. Two key digital tools that have revolutionized this landscape are Computer-Aided Design (CAD) and Building Information Modeling (BIM). While both technologies aim to improve design accuracy, collaboration, and efficiency, they differ in their approach, technical functionality, and the fields in which they are most commonly applied. Understanding these differences, as well as the software ecosystems that support them, is crucial for anyone involved in architecture, engineering, or construction.

Computer-Aided Design (CAD) refers to the use of software to assist in the creation and documentation of design files. CAD systems are used extensively in engineering and manufacturing for designing a wide range of products, from simple mechanical components to complex industrial systems (Tornincasa and F, 2010). The primary goal of CAD is to create a detailed geometric

representation of physical objects. It allows designers and engineers to create 2D or 3D models that can be analyzed, modified, and fine-tuned before physical production. CAD helps to reduce errors, saves time, and enables the automation of some repetitive tasks, such as part specifications and material calculations. Common fields of application for CAD include mechanical engineering, automotive design, aerospace, and electrical systems. Some of the primary software tools used in CAD design include AutoCAD¹¹, SolidWorks¹², CATIA¹³, and Siemens NX¹⁴.

While CAD focuses on the design and geometry of objects, BIM takes a broader approach by integrating design, construction, and operational aspects into unified models in unified databases. BIM is a process-oriented technology that creates a 3D model representing the physical and functional characteristics of a building or infrastructure project, such as whether the functions of the elements are integrated in heating, ventilation, air conditioning, electrical systems, plumbing, decorations, etc. (Borkowski, 2023) In BIM, an element in the model may even contain temporal information (4D BIM), details regarding cost (5D BIM) or other types of data. BIM is widely used in architecture, structural engineering, and construction because it provides a comprehensive view of the project and facilitates collaboration. BIM tools are designed not only for visualization but also for the simulation of building performance, cost estimation, scheduling, and risk analysis. Some of the most popular BIM software includes Autodesk Revit¹⁵, Archicad¹⁶, and Tekla Structures¹⁷.

Although CAD and BIM can be used together in a design workflow, they differ in several fundamental ways: CAD systems are usually used by individual designers or teams with limited interaction between disciplines (such as architects, engineers, and contractors). BIM incentivizes the collaboration across different teams and stakeholders by offering a shared digital platform. This multidisciplinary approach should facilitate coordination, help reducing errors and improving outcomes. Another important distinction is that CAD tools are generally used during the design and manufacturing stages but are not typically employed beyond that, while BIM covers the entire lifecycle of a building, from design and construction to operation and maintenance, long after the construction phase is completed.

CERN also relies heavily on digital models to support the design, maintenance, and operation of its complex infrastructure. The models used for both accelerators and facilities are primarily CAD-based, with CATIA being the software of choice. However, due to the high level of detail and the sheer size of some accelerator models, working with them can become challenging—at times, even impractical—without partitioning the data. CERN has, as of now, not adopted BIM tools and frameworks. Although some commercial solutions integrate construction sequencing with time-based model progression aligned with project tasks, modifying existing workflows and tools poses organizational challenges. Conversely, developing custom solutions offers greater

¹¹<https://www.autodesk.com/products/autocad/overview?term=1-YEAR&tab=subscription>

¹²<https://www.solidworks.com/>

¹³<https://www.3ds.com/products/catia>

¹⁴<https://plm.sw.siemens.com/en-US/nx/>

¹⁵<https://www.autodesk.com/products/revit/overview?term=1-YEAR&tab=subscription>

¹⁶<https://graphisoft.com/solutions/archicad/>

¹⁷<https://www.tekla.com/products/tekla-structures>

flexibility, accommodates additional requirements, and is more cost-effective. Additionally, many CERN facilities date back several decades, leading to inconsistencies in modeling standards and gaps in documentation. This dissertation specifically focuses on the TCC2 facility in the North Area of CERN.

Chapter 3

Related Work

To address the research questions outlined in Section 1.4 and contribute meaningfully to the field of data visualization, it is essential to first examine existing knowledge, techniques, and technologies currently applied to similar objectives. This chapter presents a review of the state of the art in visualizing and interacting with time-evolving 3D models. The chapter is structured in two main parts: first, detailing the systematic review process that guided the literature search and analysis, and second, presenting the findings organized into key areas.

3.1 Systematic Review Process

Data visualization features many different branches. One of them is the visualization and interaction with 3D models that evolve over time, which are also known as 4D models. In particular, models that change composition, with the addition and removal of components as time passes are the focus of this review. The main objective of this literature review is to identify the visualization techniques and tools proposed by others that can be adopted to display the information present in these models to users in a clear way. To that effect, a systematic literature review is performed, which first encompasses the identification of records after searching databases and then the selection of relevant publications based on some defined criteria. Figure 3.1 presents the flowchart of the followed methodology.

3.1.1 Identification Phase

To find relevant records about the specified domain, four digital online libraries were utilized to collect conference papers, scientific articles, and books. These libraries are ACM Digital Library, IEEE Xplore, ScienceDirect, and Scopus. The search started with multiple queries (regarding spatial and temporal visualizations), but soon it was evident that they could be merged into a single query string:

- **Query String:** ("space-time" OR spatial OR spatiotemporal OR "spatio-temporal" OR temporal OR "time-based" OR "time-evolving" OR "time-varying" OR "time-bound" OR

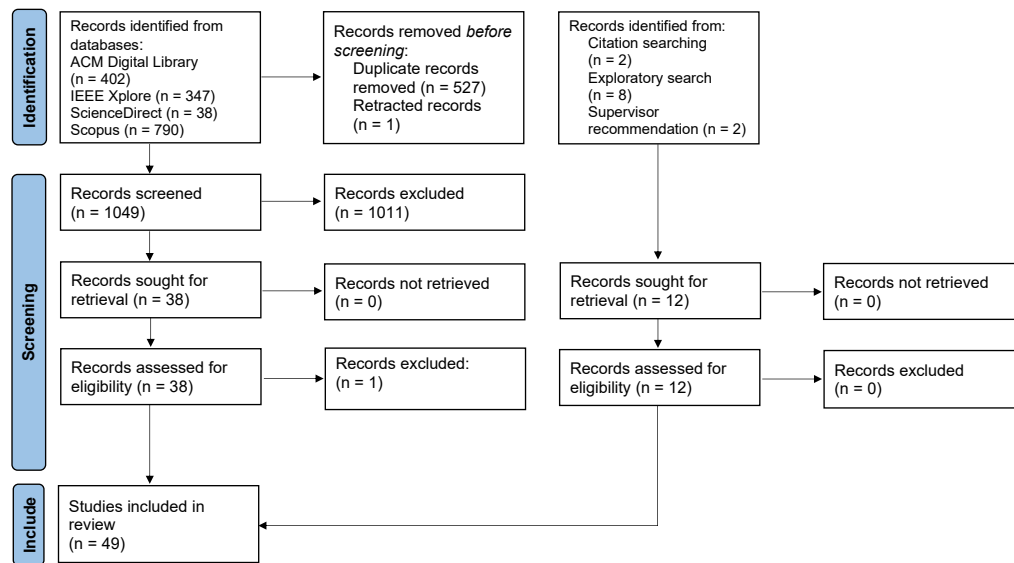


Figure 3.1: Flowchart according to PRISMA methodology.

chronological) NEAR/3 visual* AND (((3D OR "three-dimensional") NEAR/2 model) OR 4D OR CAD OR BIM)

The query string considers the more relevant type of visualizations that are being pursued, which are temporal visualizations. Naturally, some synonyms of "temporal" were included. Not only that, but spatial visualizations ("spatiotemporal" is contained in the synonyms for "temporal") were also considered, as there may be relevant information pertaining to how to display information of 3D models, where the spatiality is the most important factor, and that can be useful in designing a temporal visualization system. The wildcard "*" in "visual*" is present to match all terms that could refer to a visualization ("visual", "visualizing", "visualization", etc.) and both versions (American and British) of those words. The first "NEAR" operator is present to reduce the number of publications that mention visualization in a context separate from the temporal aspects, while the second is meant to retrieve larger descriptions such as "3D city models". Lastly, the visualizations must be related to 3D or 4D modeling or, more specifically, CAD or BIM models.

The search terms must be adapted to work with the four selected libraries while making use of some available filters:

- **ACM Digital Library:** As there is no "NEAR" operator, it is substituted with an "AND" operator. Search is only within the abstract;
- **IEEE Xplore:** Default search fields;

- **ScienceDirect:** There is a limit of 8 operators, so the query is simplified, while trying to maintain the most common expressions, to: "(spatial OR spatiotemporal OR "spatio-temporal" OR temporal OR "time-evolving") AND visualization AND (3D OR CAD OR BIM)". Search limited to the subjects of "Computer Science" and "Engineering";
- **Scopus:** The query string is as presented previously, but uses "W/" instead of "NEAR/" as it is the equivalent operator. Search limited to the subjects of "Computer Science" and "Engineering" within "Article Title, Abstract, Keywords".

Table 3.1: Number of results per digital library for the query string.

Library	QS1
ACM Digital Library	402
IEEE Xplore	347
ScienceDirect	38
Scopus	790
Total	1577

3.1.2 Screening Phase

After removing duplicates and one retracted record from the 1577 collected in Table 3.1, 1049 remained. The number of documents retrieved from ScienceDirect is significantly lower than from other databases. Although the simplified query is less restrictive than the original, it omits some terms, which may account for the reduced results. Combining multiple queries and filtering the results to better align with the full query could be a viable workaround. In the following step, 534 publications were analyzed individually and kept or rejected based on the inclusion criteria 3 and onward present in Table 3.2. The first two criteria trivially exclude any citation, reference or related paper that could have appeared alongside the search results. The remaining publications were subject to a semi-automatic verification to assess their relevance aided by a Large Language Model, which resulted in the rejection of 333 items. Having sampled 40 of the rejected materials analyzing their title, abstract and reason given from the model for the exclusion, this method was considered adequate as all of them were, in fact, not relevant. The remaining 182 records were analyzed manually just as the first 534.

These inclusion criteria ensure that, in the next phases, only the publications relevant to the research and that are highly likely to contribute to the production of this work will be further studied. After the screening, 38 records were kept and for all of them, the full-text was accessed, either via Zotero's "Find full-text" feature or manual fetching. As the number of publications kept was low (the expectation was to find at least 40), ResearchRabbit was used to select 2 more references that cite some of the previously selected records that appeared relevant according to the inclusion criteria and that contained the keyword "visualization" (McKinney and Fischer, 1998; Golparvar-Fard et al., 2009). Two publications were manually added to the library by recommendation of this

Table 3.2: Inclusion criteria used for selecting records collected form the digital libraries.

Phases	Criteria
1	Is a search engine result
2	Article, Conference Paper or Book
3	Written in English
4	Inside the field or sub-fields of Computer Science
5	Not a usability study
6	Not an education study or experiment
7	The subject of study is not exclusive to AR or VR
8	Main subject of study is related to 3D/4D visualization/modeling or information perception or modeling, or models that vary over time

dissertation's supervisors (Aigner et al., 2023; Carvalho et al., 2008) and 8 others were selected while performing an exploratory analysis using Google Scholar with the query string adapted to use the operator "AROUND" instead of "NEAR" (Andrienko et al., 2003; Bach et al., 2014; Besançon et al., 2021; Jaillot, 2020; Masoodian et al., 2015; Mota et al., 2023; Nienhaus et al., 2006; Sourina, 2007).

The resulting set of 49 publications can be seen in Table 3.3 along with a brief summary pertaining to the fields of study to be reviewed and that are relevant to this dissertation.

Table 3.3: Overview of selected records.

Reference	Summary
Aigner et al. (2023)	Extensive review of time-oriented data visualizations
Andrienko et al. (2003)	Review of spatio-temporal visualizations
Farrimond and Hetherington (2004)	Using 3D and a XML-based structure to visualize temporal data
Mao et al. (2016)	Spatiotemporal data visualization using Spark
Bach et al. (2014)	A review of temporal data visualizations based on space-time cube operations
Besançon et al. (2021)	Review of 3D spatial interaction techniques
Brazauskas et al. (2021)	BIM real-time visualization
Card and Nation (2002)	Focus+context visualization methods, based on a tree DS that stores degrees-of-interest
Carvalho et al. (2008)	Spatio-temporal visualization
Cheymol et al. (2022)	Interactions in a 3D visualization using a multi-function slider

Continued on next page

Reference	Summary
Vranic et al. (2001)	3D object retrieval via representation with harmonics and PCA
Dou et al. (2020)	Spatio-temporal data visualization for GIS and BIM
Dursun et al. (2022)	Integration of BIM with GIS
Eligehausen et al. (2013)	Integration of Blender with GIS for 4D visualization
Ye et al. (2012)	Hydrodynamic visualization with GIS
Froehlich and Bowman (2009)	Analysis of movement and trajectory
Gautier et al. (2023)	Visualization techniques using space-time cube
Golparvar-Fard et al. (2009)	4D models from photographs
Guo et al. (2014)	Indirect temporal visualization of 3D models information
Piringer et al. (2004)	3D scatter-plots with temporal data
Hertel et al. (2009)	Spatio-temporal city models in CityGML
Ivson et al. (2018)	4D visualization of BIM models in construction
Ivson et al. (2020)	Visualization tasks in BIM
Xia et al. (2014)	Tree structure with temporal annotations
Jaillot (2020)	Web visualization of time-evolving city models
Kim et al. (2017)	4D data visualization techniques
Kim et al. (2003)	Temporal Analysis in 4D GIS
Li et al. (2009)	Spatio-temporal visualization of buildings in Google SketchUp
Masoodian et al. (2015)	Comparison of focus+context techniques using multi-layered displays
McKinney and Fischer (1998)	4D CAD, modeling and performing tasks
Mota et al. (2023)	Annotations with temporal data in 3D models
Neumann et al. (2003)	Projection of temporal imagery on 3D surfaces
Ni et al. (2017)	Spatiotemporal data management system for structural health monitoring of large-scale structures
Nienhaus et al. (2006)	Illustrating spatial assembly of CSG models
Pindat et al. (2013)	Visualization of the interior of CAD models
Ponchio and Hormann (2008)	4D hypermeshes represent dynamic geometry and can be intersected by hyperplanes
Qing et al. (2011)	Modeling and visualization of moving objects
Rodríguez-Gonzálvez et al. (2017)	4D modeling and visualization of buildings/cultural heritage
Samet (2016)	GIS spatial visualization
Sando et al. (2009)	Information perception in 3D structures. Animations vs static views
Simões et al. (2015)	Spatio-temporal visualization of 3D models on mobile

Continued on next page

Reference	Summary
Sourina (2007)	Spatio-temporal querying
Tsang et al. (2004)	Temporal thumbnails: aggregation of time varying viewing data
Wang et al. (2016)	Modeling of sub-aquatic terrain temporal data
Wen and Lin (2010)	Framework for 3D city models using a managed object based method
Peng et al. (2011)	An optimal color mapping strategy based on energy minimization for time-varying data
Yang et al. (2005)	Performance optimization of 3D visualizations

3.1.3 Review and Analysis

Table 3.4: Number of reviewed publications by type.

Type	Number of Publications
Book	1
PhD Thesis	1
Article	16
Conference Paper	31

The publications collected present some variety; however, it is possible to observe in Table 3.3 that some topics are more prevalent than others. BIM appears as the most frequent modeling system, although CAD and GIS are mentioned a few times as well. The usage of urban modeling datasets also has a notorious presence in the reviewed literature, more so than models of industrial machines. The most common publication type is conference paper (Table 3.4), and it is worth noting that [Aigner et al. \(2023\)](#) is the only book selected and there is only one appearance of a PhD thesis ([Jaillot, 2020](#)). While 2009 is the year with most selected publications, overall, they are well spread out throughout the years (Fig. 3.2).

After reviewing the publications, some trend analysis can be performed. The literature trends show a clear division between 2D temporal visualizations (focusing on traditional charts and maps) and 3D/4D approaches that handle spatial-temporal data. The latter dominates this pool of publications, particularly in domains like GIS, BIM, and CAD, where understanding temporal changes in 3D models is crucial. Technology-wise, there are some mentions of web-based solutions and real-time visualization capabilities, with a focus on performance optimization for large datasets. Key application areas cluster around urban development (GIS, city modeling), construction (BIM, CAD), and cultural heritage, with visualization techniques ranging from basic temporal annotations to sophisticated space-time cube operations. Common themes across these domains include the integration of multiple data sources, interactive visualization methods, and the balance between performance and user experience, particularly in web and mobile environments.

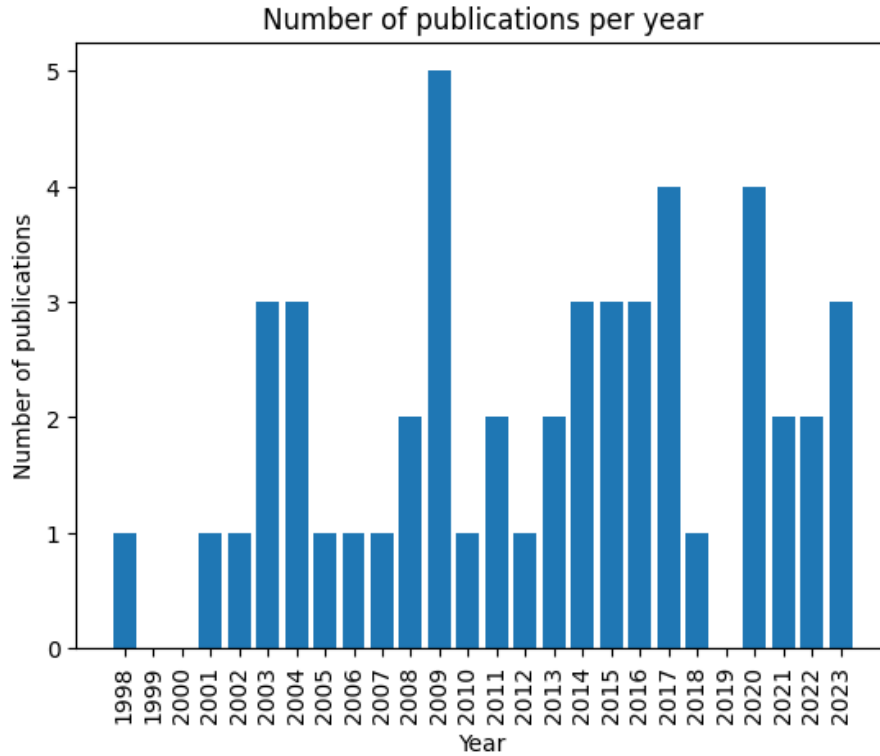


Figure 3.2: Publication year distribution of selected records.

3.2 State of the Art

Time is a fundamental dimension in data, characterized by its distinct and unique properties. Consequently, creating good temporal visualizations requires treating time-oriented data in a special way, with appropriate visual, interactive and computational methods (Aigner et al., 2023). This section provides a detailed analysis and synthesis of past and current research advancements in the fields of data visualization, interaction, and analysis of 3D and 4D data. Additionally, as CERN deals specifically with large models of complex machines, special visualization and interaction techniques which are not specific to temporal visualizations, but may nonetheless enhance their comprehension, are taken into account. The reviewed methods and applications are categorized into the following three primary topics:

- Visualization of 3D models and interaction techniques.
- Representation structures for 4D models.
- Visualization techniques for data that changes over time.

All of these knowledge fields offer valuable insights for addressing the problem defined in this dissertation. To build and visualize a model of a complex machine that changes configuration over time, it is important to use effective techniques for interacting with the base model, make the changes easy to understand, and apply methods for managing and extracting spatial features.

3.2.1 Visualization and Interaction with 3D Models

The visualization of 3D models and the interaction with them is a crucial aspect of various fields, including Computer Graphics, CAD, and scientific visualization. Several papers address different facets of this topic, ranging from fundamental 3D model retrieval tools to interactive rendering and detailed visualization within complex models.

3.2.1.1 Existing techniques

Beyond basic viewing, interactive exploration of 3D objects requires specialized techniques. [Pin-dat et al. \(2013\)](#) introduces Gimlenses, a multi-view detail-in-context technique allowing users to "drill" into complex models. By creating cone-shaped cuts and offering synchronized view-points at different scales, Gimlenses reveals internal details without losing overall context. It is a focus+context technique used to detail the view of otherwise occluded or partially hidden components of an object. It is possible to instantiate a second Gimlens inside the view created by a previous lens, though this can create a conflict as the focused object of a lens can be hidden to visualize the focused object of another lens. This predicament is solved by rendering the portion of the hidden object that lies near the focus point of the first lens with a transparency effect. Figure 3.3 depicts a Gimlens used to view a bolt inside a car engine where the main window provides the context view and a second window contains the focused object highlighted in green. This approach is useful for visualizing intricate models like car engines or anatomical structures, potentially extending to time-varying models by applying the lensing to different temporal states. It also employs automatic visibility adjustment and orientation constraints to ensure a coherent and interactive exploration experience even within complex models.

Similarly, to visualize a model with layers (e.g., the exterior and interior of a car), [Masoodian et al. \(2015\)](#) proposes five focus and context awareness techniques: *Object Isolation*, *Component Segregation*, *Peeling Focus*, *Peeling Focus and Context* and *Slicing*. The implementation was made using multilayered displays, nevertheless, the following approaches could be implemented on a single screen aided by variable opacity mechanisms. Object isolation separates an object of interest onto the front layer while displaying the rest of the model in a transparent back layer, allowing focused editing while maintaining context. Component segregation divides the components of an object based on their proximity to the viewer, placing closer components on the front layer and distant ones on the back, enabling precise manipulation without overlapping distractions. Peeling focus gradually removes portions of the front layer to expose occluded objects beneath, helping to locate hidden components while preserving the overall model context. Peeling focus and context extends this by removing portions from both layers simultaneously, dynamically revealing interior parts based on the model's orientation, making it ideal for tasks involving single-object contexts. A comparison of the two peeling techniques is presented in Fig. 3.4, where the focus and context are, respectively, the exterior and interior of the car model. The last method is slicing, which incrementally transfers sections of the model from the back layer to the front layer, using a clipping plane to selectively reveal and display solid details for in-depth inspection.

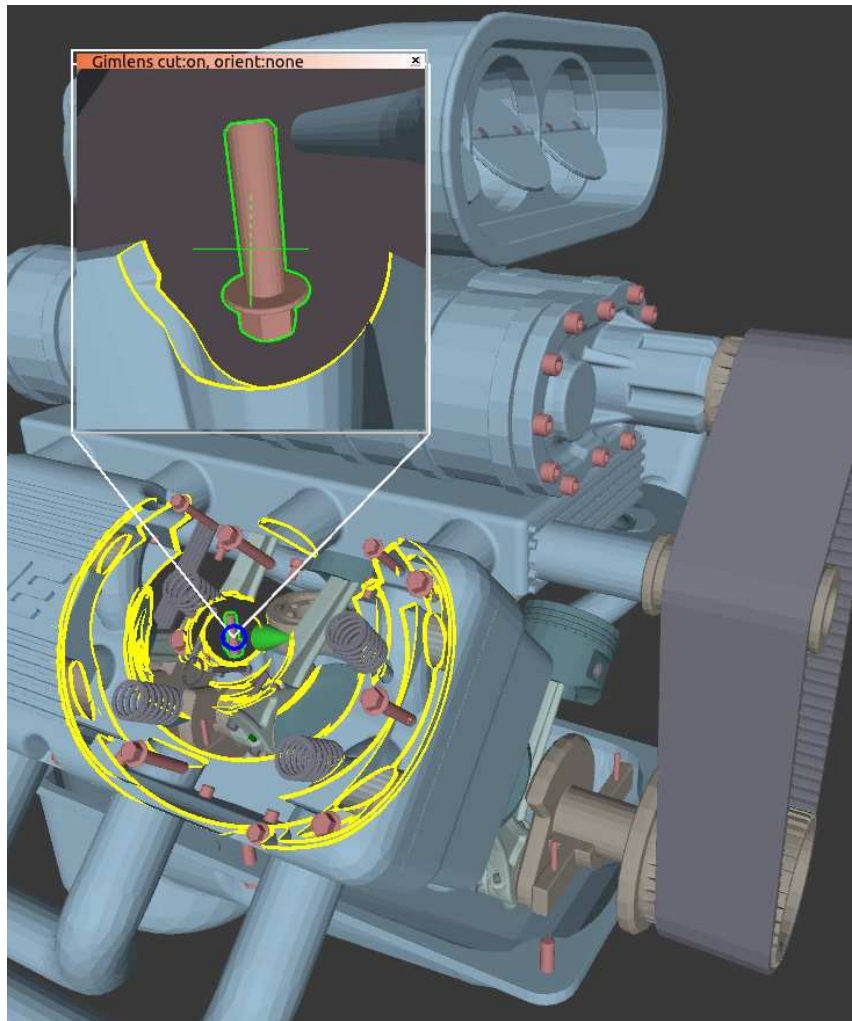
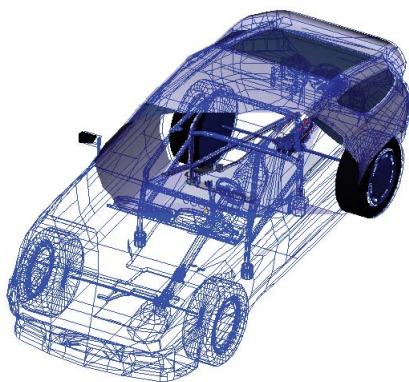


Figure 3.3: A Gimlens magnifying the interior of a car engine. Taken from [Pindat et al. \(2013\)](#).



(a) Peeling focus.



(b) Peeling focus and context.

Figure 3.4: Peeling techniques. Taken from [Masoodian et al. \(2015\)](#).

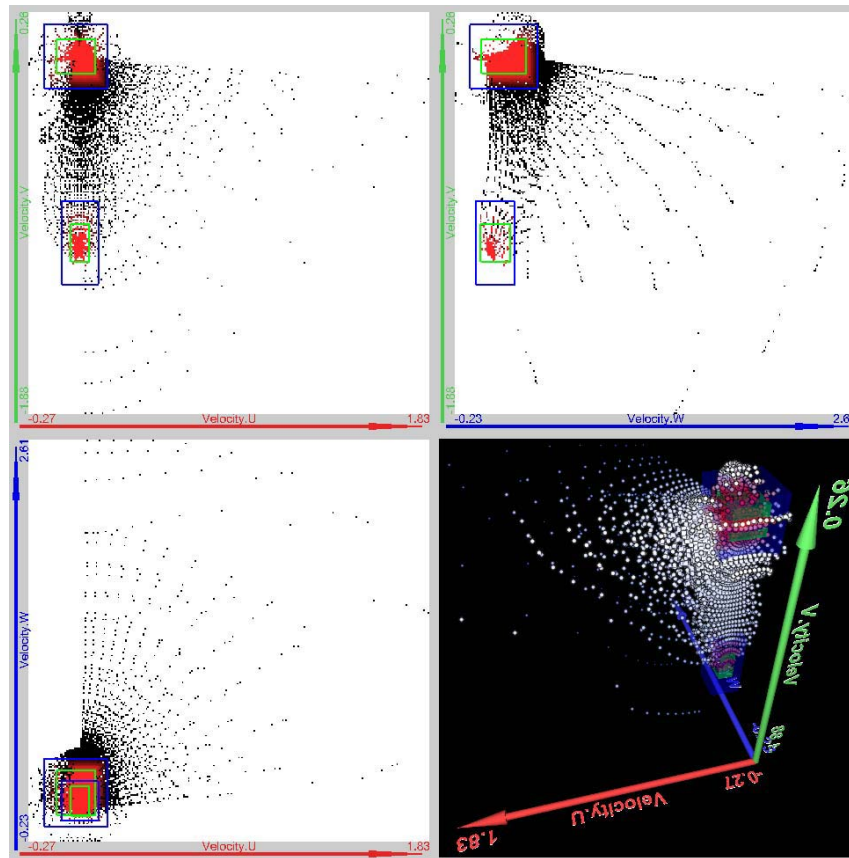


Figure 3.5: Combined 2D/3D view with brushes. Taken from [Piringer et al. \(2004\)](#).

Another aspect of interaction involves the combination of 2D and 3D views. [Piringer et al. \(2004\)](#) presents a linked 2D/3D scatterplot approach that takes advantage of the familiarity of 2D interaction while providing a richer 3D visualization. This work improves depth perception in 3D scatter-plots using depth-dependent point size, halos, and color mapping. This visualization also provides linked 2D views (e.g. views perpendicular to each axis) that can be interacted with using brushing techniques. Brushing refers to the interactive selection of data points in one view (such as a 2D plot) that are simultaneously highlighted or selected in other linked views (such as a 3D plot), helping users focus on specific subsets of data. In this example, the brushes in the 2D views are displayed as transparent boxes in the 3D view, as shown in Fig. 3.5. It was created with the SimView software and utilizes a flow simulation dataset from the automotive industry.

[Mota et al. \(2023\)](#) is another example of the usage of linked views with 3D models. These and other types of views were used to display temporal data that is not spatial, like the exposure of a building to sunlight. The authors describe four visualization techniques for combining thematic temporal data in 3D urban environments, along with their evaluation results. The techniques are divided into two categories: plot-based and color-coded designs. The first plot-based technique, Embedded View (Fig. 3.6 (a)), directly embeds plots on building surfaces (e.g., bar charts). It uses a decal-based approach where charts follow the building's geometry, with attribute values shown on the vertical axis and time stamps on the horizontal axis. The second plot-based technique,

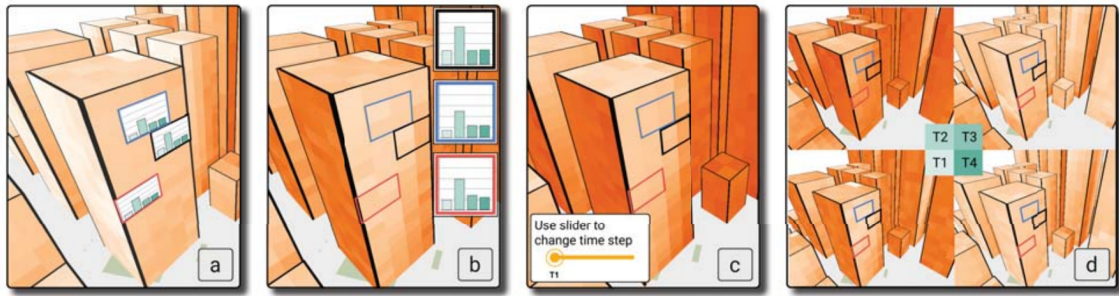


Figure 3.6: The four spatio-temporal visualizations compared in study: (a) Embedded View, (b) Linked View, (c) Temporal Juxtaposition, and (d) Spatial Juxtaposition. Taken from [Mota et al. \(2023\)](#).

Linked View (Fig. 3.6 (b)), similarly uses charts but places them on the right side of the screen instead of on the buildings. For color-coded designs, Temporal Juxtaposition (Fig. 3.6 (c)) shows one data instance at a time using a sequential colormap, with users controlling time steps through a slider. Spatial Juxtaposition (Fig. 3.6 (d)) arranges all data instances in a grid layout. The evaluation results revealed that plot-based visualizations led to higher accuracy in analytical tasks, as participants could better discriminate data when values were encoded as positions on a common scale compared to color encoding. However, color-coded visualizations allowed faster task completion. Plot-based visualizations also showed more consistent performance as task complexity increased.

Other techniques, such as animations ([Sando et al., 2009](#)) and even sliders can positively impact the way a user is able to extract relevant information from data visualizations. The rubber slider mechanism from [Cheymol et al. \(2022\)](#) is an interaction interface designed for VR applications, used to manipulate timelines and temporal data. It builds on the traditional 2D slider concept by introducing a second degree of freedom, allowing users to bend the slider's axis for fine-grained control over a secondary value. This bending adjusts the time range scale on a timeline, with visual feedback through color changes indicating the degree of curvature. Two control modes are provided: Direct Control, where the cursor maps directly to a year and bending narrows the displayed range, and Incremental Control, which shifts the year incrementally while bending modifies the precision of adjustments. Tested in a VR environment showcasing art pieces on a 3D Earth model, the rubber slider can aid the dynamic exploration of temporal and spatial data.

3.2.1.2 Additional Works

Table 3.5 lists all the publications reviewed that contain visualization and/or interaction techniques with 3D models that may be useful to understand large and complex models. While the literature search identified additional publications related to 3D visualization and interaction, some of the works, though valuable in their respective contexts, are less directly relevant to the specific focus of this dissertation than the key works discussed earlier. These publications explore visualization techniques that are not specifically oriented toward understanding model evolution over time.

Table 3.5: Reviewed records related to 3D visualization and interaction.

Reference	Visualization	Interaction	Datasets
Besaçon et al. (2021)	AR, VR	Haptics, Mid-air gestures	Varied
Dursun et al. (2022)	LoDs, Ortophoto		City models
Froehlich and Bowman (2009)	Object movement, Trajectory	Time slider, Trajectory dragging	Varied
Masoodian et al. (2015)	Multi-Layered Displays, Focus+Context	Selection, Clipping planes	Car model
McKinney and Fischer (1998)	View status, Alerts	Perform visual tasks, Annotations	CAD models of buildings
Neumann et al. (2003)	Media projection on 3D models		University campus
Nienhaus et al. (2006)	Transparency and edge enhancing		CSG models
Pindat et al. (2013)	Focus+context on occluded objects	Highlight and select	Car engine, Skull, Warehouse
Sando et al. (2009)	Animation, Static views		Primitive objects

3.2.1.3 Summary

The reviewed techniques primarily focus on three key areas: detail-in-context visualization, layered visualization, and temporal data representation. Gimlenses offers a solution for examining internal details of complex models while maintaining context, which could be valuable for inspecting specific accelerator components without losing the broader perspective. The five focus and context awareness techniques (Object Isolation, Component Segregation, Peeling Focus, Peeling Focus and Context, and Slicing) provide various approaches to handle layered visualizations, potentially useful for examining different temporal states of CERN's equipment.

From Mota et al. (2023), the comparison between plot-based (Embedded View and Linked View) and color-coded designs (Temporal and Spatial Juxtaposition) may offer insights related to maintenance tasks or machine status for CERN's visualization needs. Plot-based visualizations proved more accurate for analytical tasks, while color-coded visualizations were faster to use. This trade-off between accuracy and speed could inform the design choices for CERN's visualization tool. The rubber slider mechanism also presents an interesting interaction technique for temporal navigation, which could be adapted for precise control over time intervals and respective time-varying components in CERN's models.

For CERN's specific application, these techniques could be combined to improve the current visualization tool and the way task information is presented and how the user controls time changes, by: navigating through temporal changes in equipment, maintaining context while examining specific components, comparing different temporal states and interacting precisely with time-based data.

3.2.2 4D Models Structure and Representation

Creating a structure that effectively supports the management and representation of the contents of a 4D model—encompassing 3D spatial data and its evolution over time—is a fundamental prerequisite for enabling efficient querying, selection, and visualization of desired model states or components. To address the challenges associated with large datasets and the integration of the temporal dimension, various techniques have been developed that optimize both the organization of data and its dynamic representation. These methods enhance the ability to interact with and highlight changes over time, allowing for more efficient exploration of complex, time-evolving models.

3.2.2.1 Traditional Approaches

To include temporal annotations in 3D datasets, first it needs to be defined which concepts should be used. Per ISO (2002) there are two time primitives, *instant* and *period*, where an *instant* is a “0-dimensional geometric primitive representing position in time” and a *period* is a “one-dimensional geometric primitive representing extent in time”. The standard further define two types of time: *valid time* and *transaction time*, where the former is the “time when a fact is true in the abstracted reality” and the latter is the “time when a fact is current in a database and may be retrieved”.

These primitives and types serve as a basis for most temporal and spatio-temporal models that use spatial information. Siabato et al. (2018) provides an extensive review of spatio-temporal models for geospatial data (in GIS), categorizing models based on a typology encompassing 18 trends in spatio-temporal modeling. These models have been used in the representation of diverse real-world temporal events occurring across different spatial and temporal scales, including phenomena like building construction, wildfire progression, pedestrian movement, and dining activities. Table 3.6 contains the complete list of approaches to model spatio-temporal data; however, *snapshots method* and *graphs-based* were found to be the ones most used in the modeling of the temporal dimension of 3D cities.

The snapshot model (Armstrong, 1988) represents one of the earliest attempts to incorporate temporal aspects into spatial databases. It operates by creating complete copies of data tables at different timestamps, essentially capturing the entire state of the spatial data at specific moments in time. This approach has been adopted by the 3D community to represent geometric evolution of various spatial scales, from individual monuments to entire cities (Jaillot, 2020). While this model enables straightforward visual comparison between different temporal states, it suffers from significant limitations. The lack of direct links between spatio-temporal objects makes complex temporal queries challenging, particularly when analyzing object lifecycles. Additionally, the model is inefficient for large datasets due to data redundancy, as unchanged objects are duplicated across timestamps.

The graph-based model (Renolen, 2000) offers a more sophisticated approach to representing spatio-temporal information. This model conceptualizes objects as existing in three possible states: static, changing, or dead, with each state persisting for a defined period. A key innovation of this approach is treating change itself as a state that occurs over time rather than as an instantaneous event. Objects can transition between these states, including the possibility of "reincarnation" from a dead state. The model defines seven types of changes: creation, alteration, destruction, reincarnation, merging/annexation, splitting/deduction, and reallocation. This approach has proven particularly valuable for handling complex temporal queries and representing changes at various spatial and temporal granularity levels, for example, in the analysis of architectural sites (Jaillot, 2020).

3.2.2.2 Advanced Data Structures and Systems

Hertel et al. (2009) addresses the problem of efficiently visualizing and querying large-scale spatio-temporal urban datasets, which include 3D city models annotated with temporal information. Visualization systems traditionally do not handle temporal changes, which limits their use for dynamic urban environments. To overcome this, the authors propose a system based on a 4D kD-tree, an extension of a binary space partitioning structure, optimized with the Surface Area Heuristic (SAH). The dataset used consists of urban models in the CityGML¹ format, augmented

¹<https://www.ogc.org/publications/standard/citygml/>

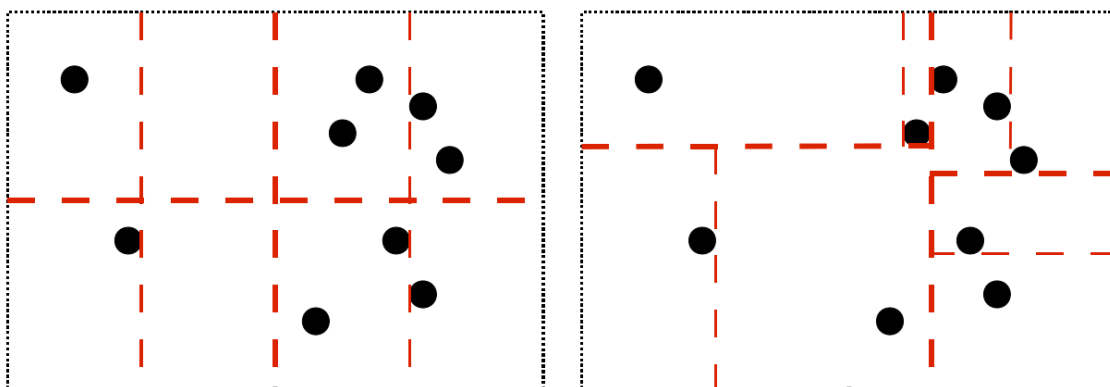


Figure 3.7: Naïve and SAH approaches to the construction of kD-trees with depth 3 for a 2 dimensional dataset. Taken from Hertel et al. (2009).

with timestamps indicating the construction and destruction dates of objects. These timestamps are treated as a fourth dimension to enable temporal exploration.

The kD-tree is employed to hierarchically partition the dataset, reducing bounding volumes and optimizing rendering and querying processes. The SAH, adapted to the 4D context, calculates partition costs based on node surface areas while considering only relevant spatial and temporal intersections. This heuristic ensures that, even for large datasets, the tree is compact and traversal efficient, with minimized bounding volumes of the buildings geometry, which skips empty space early. Figure 3.7 presents a comparison of the construction of kD-trees for a two dimensional dataset using the naïve approach and the Surface Area Heuristic. The nodes in the right tree (the divisions) appear closer to the data (black circles) compared to the tree constructed by the naïve approach. Rendering utilizes view frustum and occlusion culling, which skips the rendering of objects that are not intercepted by the camera’s view frustum or that are occluded behind other objects. The tree structure facilitates this identification of visible objects skipping occluded or irrelevant regions. This hierarchical approach ensures high frame rates (close to 100 FPS) even for complex scenes (4 GB CityGML dataset with 11 million triangles).

For large datasets, the scene is divided into equal-sized tiles. Each tile has its own kD-tree, which is stored on disk. This allows for efficient management of memory by loading only the necessary tiles into memory when they are needed for rendering. To minimize latency during rendering, tiles are loaded into memory ahead of time based on a spherical prefetch volume as it accommodates all potential camera movements, including fast rotations. Although this consumes more memory than a cone-shaped volume, the increase in data transfer is not significant and ensures smooth navigation. Spatio-temporal queries are supported through a query language inspired by the Spatio-Temporal Query Language (Erwig and Schneider, 2002), allowing users to define volumes in 4D space to retrieve objects or highlight specific features. Querying and rendering share the same traversal mechanism, ensuring seamless integration. This system addresses a very similar problem to the one presented in this dissertation; however, it is limited to appearance and disappearance events without taking into account partial changes or substitutions.

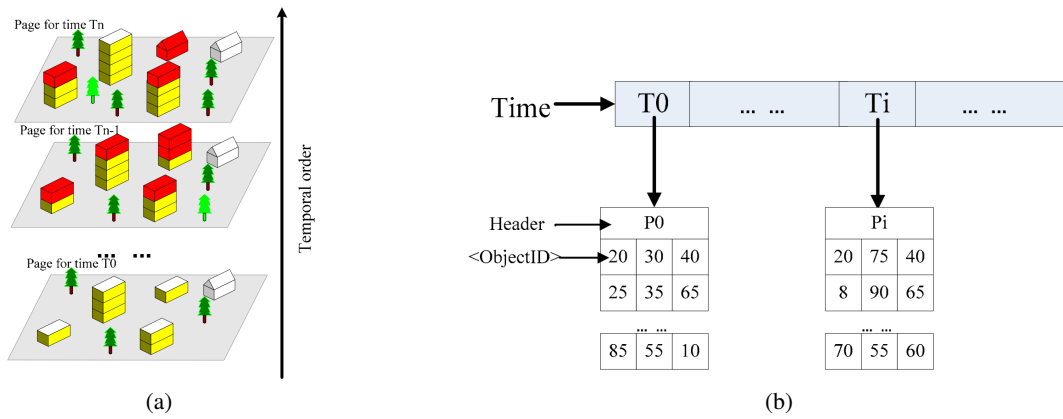


Figure 3.8: Representations of the STPM. (a) exemplifies the visual result while (b) focuses on the internal representation. Taken from Li et al. (2009).

From the same year, the Spatio-Temporal Page Model (STPM) proposed in Li et al. (2009) is a model organized around "pages" (similar to snapshots) but handles changes efficiently by using object-based storage where each entity has: unique *ObjectID*, a time attribute indicating when changes occur and a *WhatItChange* attribute tracking the type of change. The model maintains changes through an initial state (Page 0) containing all objects at time T_0 and subsequent changes are recorded by updating object attributes rather than creating complete new snapshots. This is shown in Fig. 3.8, where different pages contain different object ids. Three types of changes are tracked: NULL for new objects, DELETE for removed objects and REPLACED+ObjectID for modified objects. To reconstruct any temporal state, the system queries objects with timestamps before the desired time, filters based on *WhatItChange* status and assembles the valid objects into a new "page". This makes STPM an approach that uses snapshot-like pages for visualization, employs event-based tracking of changes, maintains object identity across time, reduces data redundancy compared to pure snapshots and provides more efficient retrieval than pure event-based systems.

Jaillot (2020) introduces two foundational models designed for the representation of 3D geospatial features, incorporating the temporal dimension to facilitate visualization of time-changing urban environments. The first model, referred to as Gen3DGeo, provides a conceptual framework that generalizes 3D geospatial standards such as 3D Tiles² and I3S³ for web delivery of geographic features like buildings or bridges. This framework organizes these features through hierarchical structures called spatial nodes, relying on spatial indexing methods like quadrees or octrees. Sets of spatial nodes are grouped into a root object called a "Set" representing the entirety of the dataset. It also accommodates multi-resolution representations by refining nodes to achieve higher levels of detail. By unifying shared concepts of 3D Tiles and I3S, Gen3DGeo ensures flexibility, interoperability, and scalability for handling large datasets efficiently while allowing extensions.

Building upon the Gen3DGeo model, the Gen4DCity model (Fig. 3.9) integrates the temporal

²<https://www.ogc.org/publications/standard/3dtiles/>

³<https://www.ogc.org/publications/standard/i3s/>

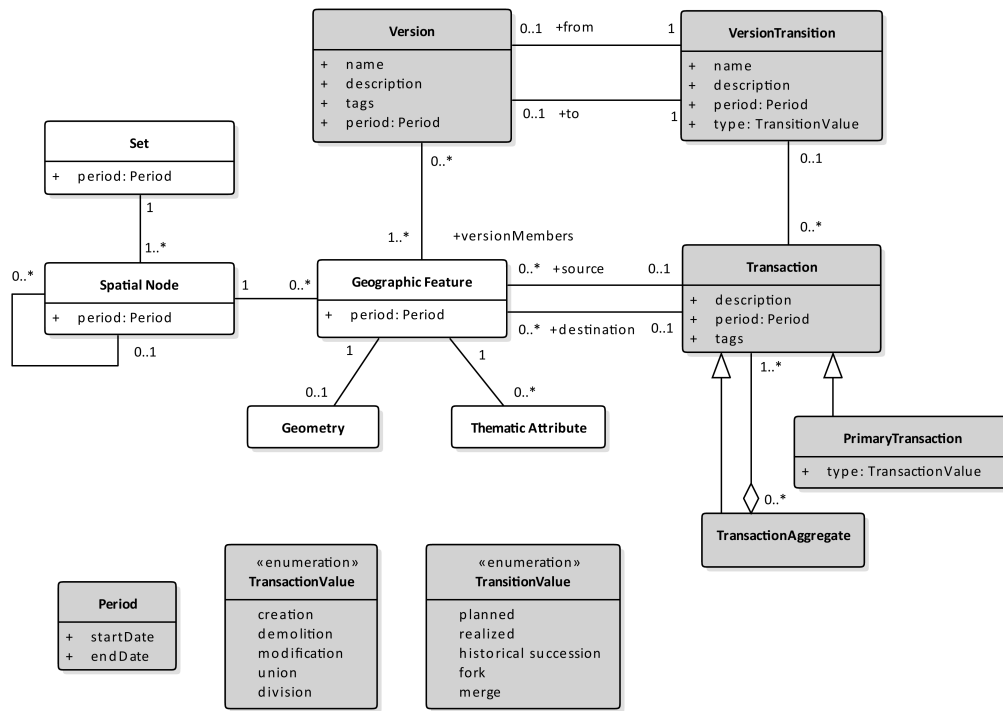


Figure 3.9: Gen4DCity: Generic conceptual model for delivering time-evolving 3D city models on the web based on standard formats. The entities of Gen3DGeo are depicted in white. The proposed extension of the temporal dimension is depicted in gray. Taken from [Jaillet \(2020\)](#).

dimension, focusing specifically on urban settings. It introduces concepts like Periods, which define the valid time intervals of geographic features (e.g., when a building is constructed or demolished), alongside spatio-temporal indexing capabilities. Additionally, the model organizes time-stamped states of spatial and thematic features into "Versions", with "VersionTransitions" used to semantically describe changes between these states, such as construction, modification, or destruction. The model is designed to support varying levels of temporal precision, from simple time-based queries to detailed temporal analysis via transactions and transitions. This extension enables the tracking of urban changes, from large-scale city-wide transformations to fine-grained modifications at the building level. The specification and implementation of the Gen4DCity model are detailed using 3D Tiles, which are extended to include temporal features. The logical model Logic3DGeo serves as a foundation, structuring datasets into tilesets linked by hierarchical spatial nodes. This model is further extended by Logic4DCity, which incorporates temporal aspects such as "Version", "VersionTransition", and "Period" within the tilesets. These entities allow for more advanced spatio-temporal indexing and can also be associated with bounding volumes to define temporal visibility for features. The integration maintains compatibility with existing 3D Tiles structures, enabling the seamless representation and management of time-evolving datasets.

The technical implementation employs the 3DTiles_temporal specification, which leverages extensions and extras defined in the core 3D Tiles standard to include temporal data without altering the base format. Dataset-wide information, including versions, transitions, and temporal

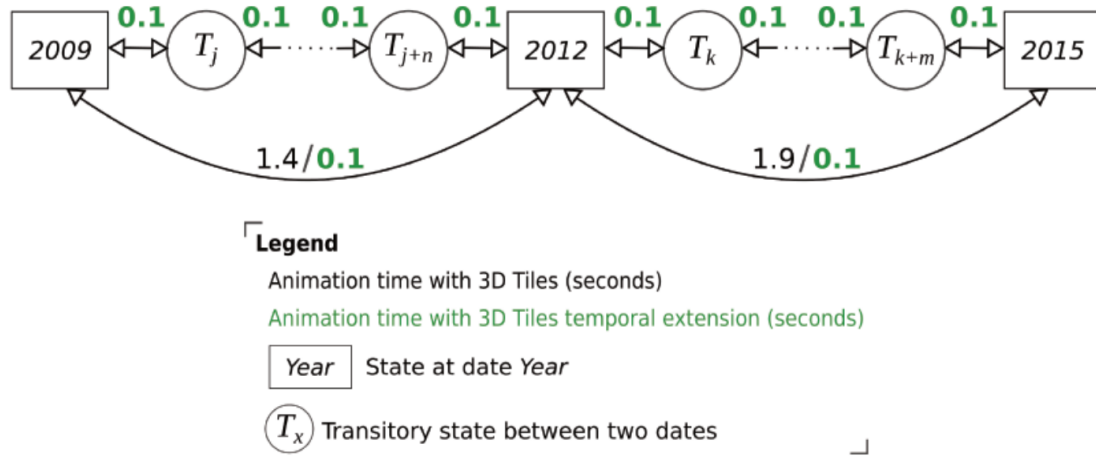


Figure 3.10: Temporal navigation comparison between 3D Tiles and 3D Tiles with temporal extension. Taken from Jaillot (2020).

metadata, is stored within tilesets. Temporal attributes like *startDate* and *endDate* are included in bounding volumes to manage spatio-temporal constraints, while thematic properties are augmented with temporal arrays via a "BatchTable". This approach minimizes data duplication by recording only the changes over time instead of repeating unchanged features, resulting in smaller, more efficient datasets that allow for interactive and scalable visualization of time-evolving cities. The temporal extensions are formalized in JSON schemas to ensure reproducibility and compatibility with existing 3D Tiles infrastructure.

A prototype system was developed to evaluate the applicability of the proposed 3DTiles_-temporal extension for visualizing time-variant 3D city models. Utilizing *py3dtiles*⁴, a tool for generating and processing 3D Tiles datasets from other formats, the system provides temporal tiling functionality to process time-stamped city models and output them in the 3DTiles_temporal format. This data is then visualized with UD-Viz⁵, an open-source web client that supports interactive 3D and temporal navigation. Experiments conducted on the Lyon dataset, consisting of buildings from 2009 to 2015, highlighted the benefits of the temporal extension. Three data representations—CityGML snapshots, basic 3D Tiles, and the 3DTiles_temporal extension—were compared, revealing substantial memory savings with the temporal approach: up to 7.3 times smaller than CityGML (3176MB and 53951 buildings in total, for the three years considered) and 1.4 times smaller than standard 3D Tiles. Additionally, it also ensured faster responsiveness during visualization (Fig. 3.10).

Gautier et al. (2023) presents a temporal city model structured as a semantic model graph that integrates different versions and changes between them. Each version contains city objects with geometry and unique identifiers. The changes (transactions) between versions are of three types: construction (new building appears), modification (building changes) and destruction (building

⁴<https://gitlab.com/py3dtiles/py3dtiles>

⁵<https://github.com/VCityTeam/UD-Viz>

Table 3.6: Overview of spatio-temporal modeling approaches. Taken from [Siabato et al. \(2018\)](#).

Modeling approach	Modeling approach	Modeling approach
Snapshot method	Semantic-based	Moving Objects
Time-stamping	Event-based	Graphs-based
Base state amendment vectors	Process-based	Lifespan-based
Space-time composite model	Ontology-based	Agents-based
Domain-based modeling	Feature-based (Entity-based)	Kinematics
Object-Oriented	Identity-based	Ontological foundations
	Conceptual modeling extensions	

disappears). These transactions are stored in an RDF⁶ knowledge graph that links versions and tracks changes over time. Implicitly, there also exists an "unchanged" transaction. The system supports filtering transactions by type and focusing on specific buildings through interactive selection. The work was implemented with the UD-Viz library, which, as previously mentioned, allows to integrate 3D data as 3D Tiles. The authors have used the temporal 3D Tiles extension from [Jaillot \(2020\)](#).

3.2.2.3 Additional Works

Table 3.7 summarizes methods used to represent 3D but mainly 4D models with respect to the management of the models and to the data structures that are used. The table contains works that assume other types of representations for objects and models other than the ones previously discussed, but none were deemed as relevant to the structuring of the models of the machines at CERN for temporal visualization. It is, however, worth mentioning that the reviewed literature does not mention the use of LODs or other techniques which enable the use of different models for the same object, which is a crucial optimization technique required to manage large models of complex machines like at CERN.

3.2.2.4 Summary

Existing temporal annotation frameworks are built upon foundational time primitives (*instant* and *period*) and types (*valid time* and *transaction time*), as defined by ISO standards, laying the groundwork for spatio-temporal modeling. Techniques such as snapshot-based approaches, graph-based models, and advanced indexing methods like 4D kD-trees emerge as key strategies for capturing temporal changes. Snapshot-based methods capture discrete temporal states but suffer from inefficiencies due to data redundancy and lack of connections between temporal instances. In contrast, graph-based models enable flexible lifecycle tracking of objects through transitions like creation, alteration, and destruction, providing more sophisticated temporal queries and improved granularity.

⁶<https://www.w3.org/RDF/>

Table 3.7: Reviewed records related to the structure and representation of 3D/4D models.

Reference	Approach
Mao et al. (2016)	Octree using geo coordinates
Card and Nation (2002)	Tree DS that uses degree-of-interest calculations
Vranic et al. (2001)	Canonical representation of model for querying
Dou et al. (2020)	Annotate geospatial data with time tags using NoSQL
Eligehausen et al. (2013)	Store objects properties in blender frames
Golparvar-Fard et al. (2009)	Overlay Time-Lapsed Photographs on 3D models and monitor construction
Hertel et al. (2009)	4D kD-tree, Spherical pre-fetching, Tiling
Ye et al. (2012)	Spatiotemporal model based on geo-objects
Xia et al. (2014)	Document annotations in a tree
Jaillot (2020)	Expansion of 3D tiles with temporal dimension
Kim et al. (2003)	4D GIS using video server
Li et al. (2009)	Spatio-Temporal Page Model in Google SketchUp
Ni et al. (2017)	Tying temporal data and construction schedule to 3D models, Nested database
Ponchio and Hormann (2008)	Represent dynamic geometry as four-dimensional hyper-mesh
Qing et al. (2011)	Spatio-Temporal data structure for moving objects
Rodríguez-González et al. (2017)	Discusses temporal evolution of models but does not implement anything
Simões et al. (2015)	Mobile framework that auto adjusts quality of rendering data transmission
Sourina (2007)	Spatio-temporal predicate for querying moving points relative to a solid
Wang et al. (2016)	Spatio-temporal snapshot slices and spatial interpolation
Wen and Lin (2010)	Framework supporting dynamic representation of Geospatial Managed Object
Yang et al. (2005)	Multi-resolution models

For managing and visualizing large-scale time-evolving 3D datasets, methods like the Spatio-Temporal Page Model (STPM) and 4D kD-trees offer efficient storage and querying. STPM minimizes redundancy by assigning each object unique identifiers and tracking changes event-based, while 4D kD-trees optimize visualization through hierarchical partitioning and efficient rendering techniques, achieving high performance even for complex datasets. Recent advancements, such as the Gen4DCity framework, build upon general 3D geospatial models like Gen3DGeo by introducing temporal constructs, enabling seamless representation of changes at urban and building scales. Integrating temporal attributes (e.g., *Periods*, *Versions*, and *VersionTransitions*) within spatial datasets enhances the scalability, interoperability, and responsiveness of temporal navigation. Notably, the development of the 3DTiles_temporal extension demonstrates substantial memory savings and efficient real-time exploration of urban transformations, making it particularly relevant for scalable web-based visualization systems. Semantic graph models further complement these approaches by associating city versions with transactional changes (construction, modification, destruction) and tracking object identities over time. Leveraging RDF-based knowledge graphs enables interactive filtering and querying, offering intuitive user interaction and analysis of urban temporal changes.

For CERN's proposed visualization tool, these techniques may be applicable. Tools like the 4D kD-tree and 3DTiles_temporal can enable interactive and high-performance visualization of evolving models, such as machine components. Graph-based and semantic models would support advanced querying and tracking of changes, allowing users to seamlessly navigate modifications over time. Combining these techniques into a unified visualization platform would allow efficient management of large-scale, dynamic datasets, trying to ensure scalability and responsiveness required for a sophisticated application like the one envisioned at CERN. The proposed approach may combine any of these techniques with the use of LOD models, which was identified as the main literature gap.

3.2.3 Visualization techniques for time-evolving data

Visualizing temporal changes in 3D models and data is crucial for understanding dynamic processes. The publications analyzed discuss various techniques, ranging from focus+context methods to 4D model representation and specific application domains like construction planning, cultural heritage and the visualization of urban datasets.

Solely 2D approaches have been excluded from this review since the target users of the visualization devised in this dissertation are highly familiarized (and many times also experienced) with 3D tools and applications, where visual fidelity and spatial understanding are paramount. The choice of a 3D environment naturally aligns with their existing workflows at CERN and cognitive models. The fundamental advantage of 3D over 2D in this context lies in its ability to directly represent the intrinsic three-dimensional nature of the models being visualized, circumventing the cognitive overhead associated with interpreting 2D projections of inherently 3D data.

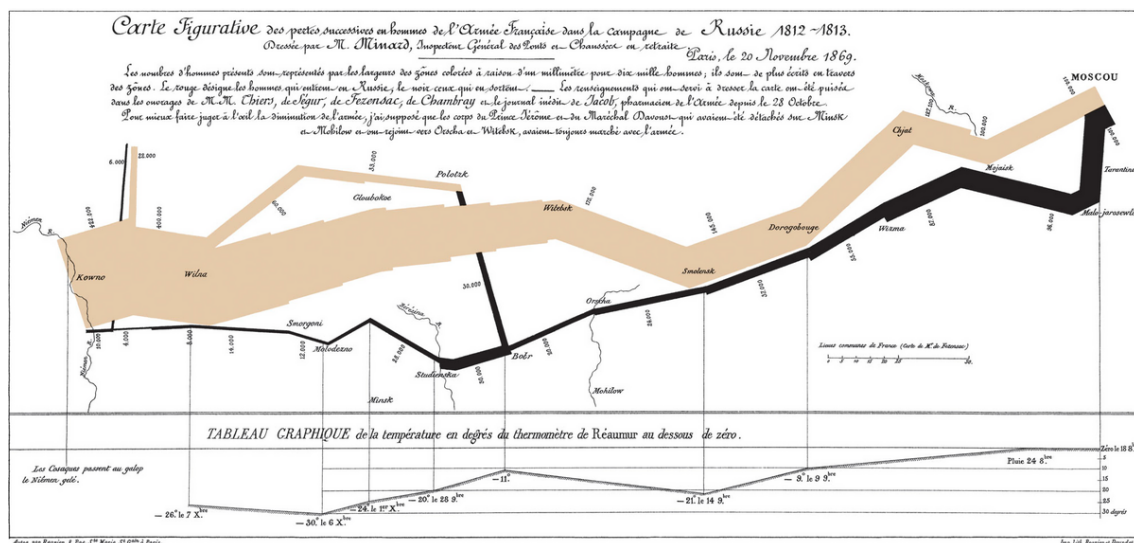


Figure 3.11: Charles Minard's 1869 chart showing the number of men in Napoleon's 1812 Russian campaign army, their movements, as well as the temperature they encountered on the return path. From <https://commons.wikimedia.org/wiki/File:Minard.png>

There are, however, classical examples of temporal evolution worth mentioning, such as Minard's map (in Fig. 3.11) of Napoleon's Russian campaign (Minard, 1869), which is a pioneering example of multivariate visualization where time is intrinsically linked to spatial progression. While such examples may not be inherently 3D, their underlying principles of depicting change over time can inform the design of effective spatiotemporal visualizations in a 3D context.

In addition, the integration of supplementary interaction and navigation aids could be considered. Approaches using multiple-view displays such as "small multiples" (Tufte, 2001) could be adapted. Specifically, a "minimap" or overview panel that provides a simplified 2D representation of the 3D scene could facilitate navigation and contextual understanding without compromising the benefits of the primary 3D view. In spite of that, the present review highlights approaches that use spatial data or that can be directly adapted to 3D.

3.2.3.1 Focus+Context

The concept of focus+context is central to visualizing temporal changes. Carvalho et al. (2008) propose a Temporal Focus + Context model using a Temporal Degree of Interest (TDOI) function to control visualization properties. The TDOI function considers record valid time and user requirements to dynamically adjust transparency, color, and other visual cues, effectively highlighting changes while preserving context. The perceived effect of displaying a 3D object applying this degree of interest is that it is possible to visualize the state of the object in a previous and posterior time states while perceiving what changes in the model are occurring.

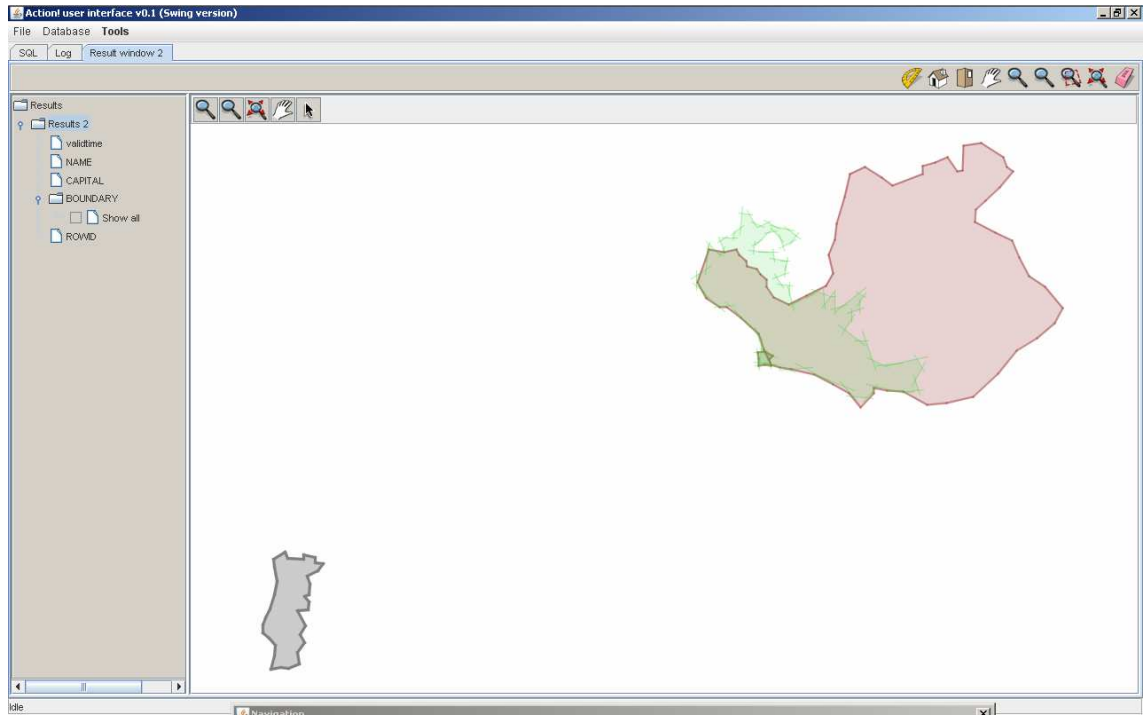


Figure 3.12: Application of TDOI to political map of Europe. Taken from [Carvalho et al. \(2008\)](#).

The TDOI for a record r (an object or component of a model) is defined as:

$$TDOI = f(I, vt, FVT, FI) \quad (3.1)$$

Where:

- I : The overall time period under user analysis $[t_{start}, t_{stop}]$.
- vt : The valid time (time span) of the record r , expressed as $[vt_{start}, vt_{stop}]$.
- FVT : A user-defined function that filters the valid time of r , returning specific temporal features.
- FI : A user-defined function describing the degree of interest across time period I .

Equation 3.1 defines a TDOI, which is a function of the temporal window in consideration, I ; is influenced by whether a record's temporal existence (vt) overlaps with the analysis window; FVT determines the specific aspects of vt that are considered in the relevance calculation; and FI weights the importance of different times within I , allowing user prioritization. Figure 3.12 depicts an example usage of a TDOI on a political map of Europe that represents only three territories. The red color indicates that, as we slide into the future, it will cease to exist, while the green territory will be created.

This study's limitations stem from its reliance on discrete data for both spatial and temporal dimensions, assuming static intervals for objects. Future work includes incorporating *transaction*

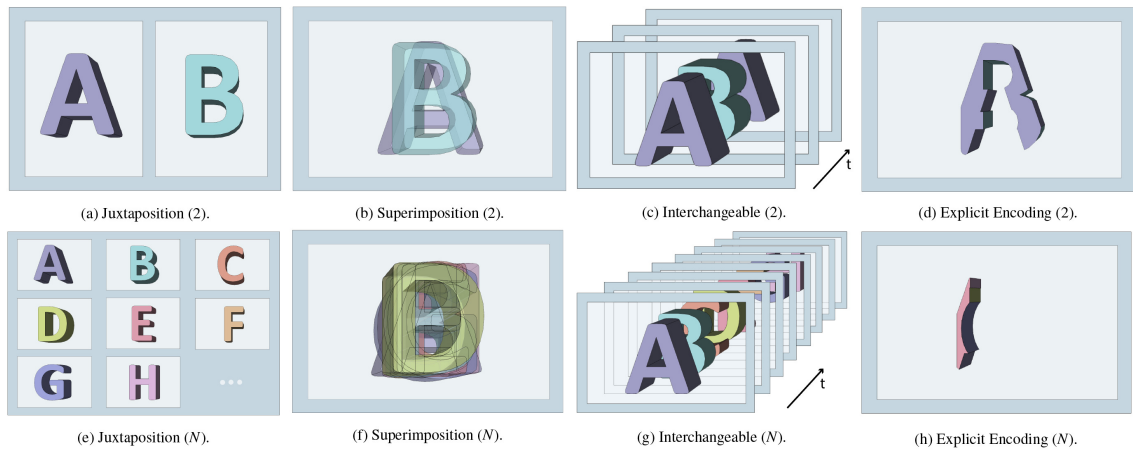


Figure 3.13: Overview of spatial 3D comparison approaches, using just two data instances (a–d) and N instances (e–h). Taken from Kim et al. (2017).

time alongside *valid time* to enhance temporal modeling and conducting a study to evaluate the model’s effectiveness for users. In spite of that, the technique presented is a valid approach to help a user at CERN correctly understand that a new component is being added to a machine, a pre-existing component is being uninstalled or even that at the same time some parts are being created while others are being removed.

3.2.3.2 Spatial 3D and 4D Comparisons

A survey on techniques that are used to compare spatial 3D and 4D data is presented in Kim et al. (2017). The visualization techniques can be categorized as one of four different types: juxtaposition, superimposition, interchangeable, and explicit encoding. Although they are less frequent, hybrid approaches that are a combination of the base approaches are also used and can even be more effective. Figure 3.13 contains simple examples of the four methods where the datasets are 3D models of letters of the alphabet.

Juxtaposition approaches consist of presenting multiple datasets side-by-side in separate coordinate spaces. Each dataset is shown independently, enabling comparison without overlap or occlusions. Although they are generally simple to interpret, they are less effective for identifying subtle differences, are challenging for large-scale datasets because of display state constraints, and force users to mentally align corresponding features between datasets. Juxtaposition can be effective for comparing snapshots of 4D datasets, but not ideal for time-varying visualizations since temporal context is not inherently represented.

Superimposition overlays multiple datasets within the same coordinate space. Features of different datasets coexist in the same view, often using transparency or color differentiation to manage occlusion. It preserves spatial context, making it easier to detect and interpret spatial variations, provides simultaneous access to all datasets in a single view and is good for detecting differences or overlaps in datasets. However, dense or complex datasets may still present problems with occlusions and overlaps. Superimposition is effective for analyzing single timesteps or

closely related timeframes in spatial 4D data, but occlusion issues make it unsuitable for long-term or highly dynamic 4D comparisons.

Interchangeable visualization presents datasets one at a time in the same coordinate space, dynamically switching between them, often using animated transitions or user-controlled switches. It allows for high-resolution, detailed inspection of each dataset and eliminates any issues with occlusion. The problems faced when using interchangeable visualizations are that they rely on user memory to compare datasets, which can lead to cognitive overload and may be ineffective in detecting subtle spatial or temporal variations. They are ideal for comparing sequential timeframes in spatial 4D datasets.

Explicit Encoding computes and visualizes derived data (e.g., differences or statistics) instead of raw datasets. Only comparative results are shown. The way different datasets are combined can be varied, being the difference between two 3D models a simple example. Differences or derived metrics (e.g., cluster trends) can be encoded in color, glyphs, or other visual metaphors. In Fig. 3.13, (d) and (h) exemplify the encoding as being the interception of the different datasets. This method has the advantage of providing a focused, simplified view of differences between datasets, highlighting subtle variations effectively, and reducing the visual complexity of the display. Nonetheless, explicit encoding omits the raw data, which might obscure underlying patterns or contexts, is heavily dependent on the meaningfulness of the derived metric and can oversimplify comparisons, especially in complex datasets. In 4D visualizations it can be particularly useful for analyzing trends; however, it does not convey the complete temporal evolution of the data.

3.2.3.3 Space-Time Cube Operations

Bach et al. (2014) makes a review of space-time cube operation used in data visualizations. The concept of space-time cubes, which combine time as one axis and two additional data dimensions as the other two, enabling complex temporal visualizations. The operations identified can be broadly grouped into time-based operations, space-based operations, and sampling with 3D visualization, each serving distinct visualization purposes. Focusing only on the time-based operations, time cutting operation allows for extracting a specific temporal snapshot, or a single time slice, from the conceptual space-time cube. This is akin to taking a still photograph of a moment in a moving scene, capturing a single instance in time. In data visualization, this is referred to as a time slice and allows users to analyze a particular moment's state. Similarly, time flattening collapses all the time slices along the temporal axis into a single two-dimensional image, much like long-exposure photography that captures motion across time into one static image. Discrete time flattening, a variation of this technique, selects specific time slices for combination rather than merging all temporal data, analogous to multiple-exposure photography. This approach has been used to summarize sequences or motion patterns over defined intervals. Another visualization operation is colored time flattening, where different time slices are assigned distinct colors, allowing viewers to distinguish trends, changes, or patterns across time. This has applications in dynamic graphs or visualizing relationships over time, such as distinguishing old connections from new ones or highlighting stages in a process. Similarly, time juxtaposing involves placing

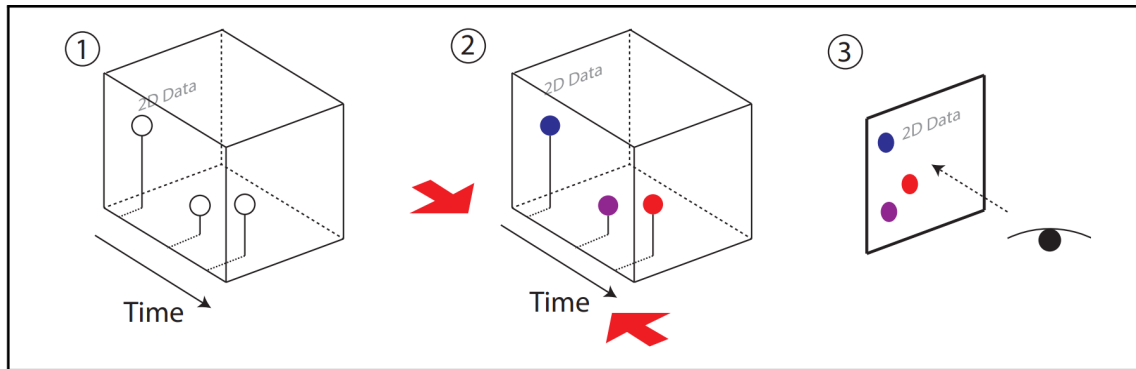


Figure 3.14: The colored time flattening operation. Taken from Bach et al. (2014).

multiple time slices side-by-side for comparison. Figure 3.14 exemplifies the colored time flattening operation and the use of the space-time cube that, in this case, contains one axis representing the temporal dimension while the other two make up a 2D referential.

These operations collectively support advanced visualization approaches, enabling the analysis and visualization of machines that change over time in the same spatial context. For instance, these visualizations could track how machine components interact, simulate assembly or disassembly processes, or show changes in system design over time. By applying operations such as colored time flattening and time juxtaposing, it becomes easier to model the temporal evolution of machines and their component states. This allows for visualizations that can predict design patterns, simulate interactions, or analyze maintenance trends.

Gautier et al. (2023) also uses a space-time cube to represent the evolution of urban geometry, where different versions of 3D city models are stacked along the z-axis representing time. Each temporal level shows buildings at that time period and transactions between levels are visualized through: color coding (green for construction, red for destruction, yellow for modification), physical links between building centroids across levels and ground projection of changes for better geographic context. Interactive features include: a double-entry time slider to select specific time ranges, the ability to toggle transaction types, building selection to focus on individual evolution, and free navigation in the 3D space. Both visualization and interaction features are visible in Fig. 3.15, where many building modifications are present, some demolitions and very few constructions.

3.2.3.4 BIM Visualizations

Ivson et al. (2020) provides a systematic review of 140 articles in visualization in Building Information Modeling and discusses new approaches to support users with visualizing time series 3D data. It classifies BIM visualization techniques according to phase in the life cycle of the project and to its use case. The five phases and the main use case for each one are:

- *Design Review*: Clash detection.
- *Work Planning*: Task Scheduling.

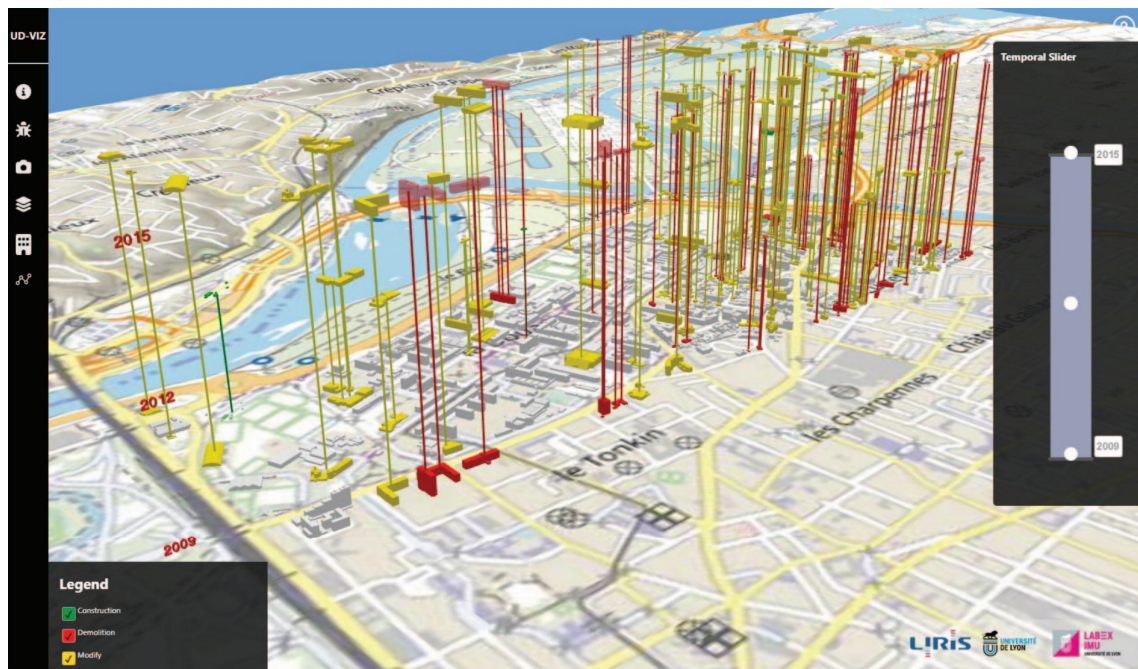


Figure 3.15: An example of the STC integration with the use case of La Doua district (Lyon/France). All transactions types are displayed in the 3D view. Taken from [Gautier et al. \(2023\)](#).

- *Work Execution*: Progress Tracking.
- *Sustainability Analysis*: Energy Analysis.
- *Facility Management*: Maintenance Management.

taking only into account the main use cases, the set of articles is reduced to 70, from which 25 different visualizations were distinguished. However, only 7 regard 2D/3D CAD views. The first two techniques, Annotation and Component Highlight, utilize visual effects to draw the user's attention. Annotations typically overlay 2D geometries and text onto a model, while Component Highlight modifies the color, outline and/or transparency of existing geometries. Techniques such as Highlight, Visibility, and Cascade Animations similarly alter visual attributes of CAD objects but are specifically designed to convey temporal information. Visibility Animation displays construction progress by making components visible only when their construction begins which portrays task progression. Highlight Animation uses changes in color or transparency of 3D geometries to indicate task status, and Cascade Animation maps time onto a vertical spatial dimension, similar to a space-time cube, and highlights tasks in sequence to better understand temporal task relationships while preserving spatial context. The last two visualization techniques extend the 3D CAD model by incorporating additional geometries. These include Physical Entities and Spatial Regions, which are used to represent locations and affected areas related to heavy machinery, temporary structures, or general fieldwork activities.

Some remarks and suggestions about 3D CAD visualizations are also discussed, mainly that highlights should use adequate color schemes, 4D animations should focus objects of interest to

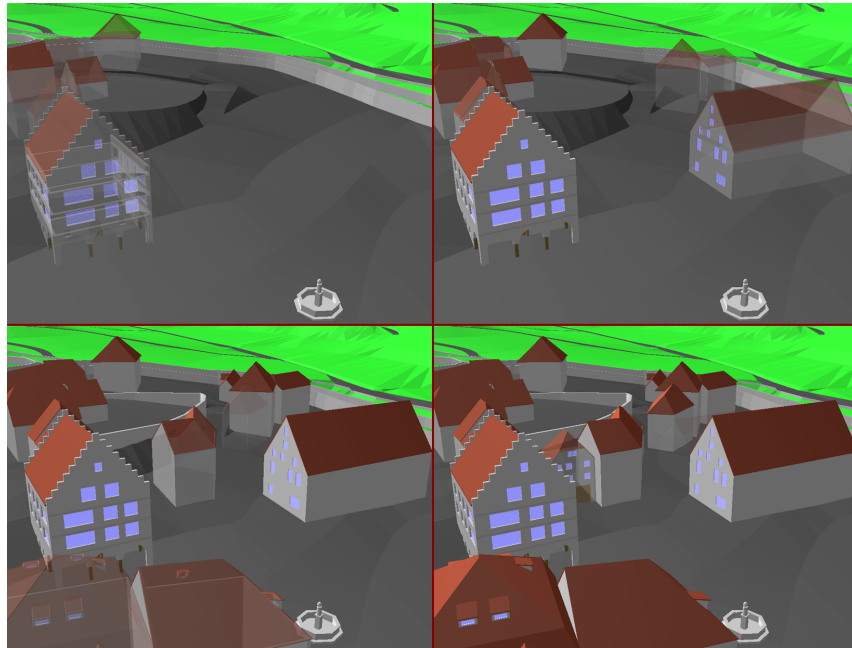


Figure 3.16: With a view frustum spanning a time difference, pop-up artifacts are moderated by fade in the objects. Taken from Hertel et al. (2009).

convey important temporal information, and there already exist techniques to manage occlusion and reduce clutter which should be used more often. The model visualization tool at CERN currently uses Annotation and Component Highlight and any of the animations presented may be used to display the changes in the machines over time.

3.2.3.5 Additional Works

While the following works are less directly relevant to the state-of-the-art, they are still worth noting for their contributions to the broader context. The previously mentioned Hertel et al. (2009) enables the visualization of urban change over time in an interactive system, using fading (Fig. 3.16) to represent temporal dynamics when buildings are appearing or disappearing. Peng et al. (2011) presents an optimized color mapping strategy for visualizing time-varying data, using bilateral filtering and energy minimization. This improves feature visualization and preserves consistency across time steps. These approaches showcase the importance of adapting visualization to the features of interest and utilizing dynamic visualization techniques. Ponchio and Hormann (2008) presents a framework for interactive rendering of dynamic geometry by representing it as 4D tetrahedral meshes. This method allows continuous interpolation between frames regardless of topological changes and leverages GPU acceleration for interactive performance. Specific applications often demand tailored visualization techniques. In construction, Golparvar-Fard et al. (2009) combines 4D as-planned models with time-lapsed photographs using augmented reality, providing a powerful tool for monitoring progress. The color-coding scheme based on earned value analysis adds an analytical dimension to the visualization. Ivson et al. (2018) proposes CasCADE, which

maps time as a spatial dimension creating a "cascading" view of construction schedules. This innovative approach facilitates schedule overview, conflict detection, and improved communication.

Table 3.8 classifies the analyzed references about the visualization of 4D and spatio-temporal models according to some of the attributes identified in Aigner et al. (2023): time primitives which can be "intervals" or "instants"; time reference which can be either "spatial" or "abstract"; dimensionality of the variables, "Multi-D" or "1D" meaning, respectively "multidimensional" and one-dimensional; mapping of time which is either "linear" or "cyclic". To expand on the visualization techniques, Table 3.9 enumerates the main mechanisms used to visualize the temporal dimension of the data. The gaps that can be identified are that none of the approaches incorporate a mechanism like TDOI (Carvalho et al., 2008) in the visualization system and so, support for visualizing simultaneous removal and addition of two models, on the same location, was not found.

3.2.3.6 Summary

Visualizing temporal changes in 3D models is critical for understanding dynamic processes across various application domains, including construction planning, and the evolution of urban datasets. The reviewed techniques reflect a diverse set of methodologies aimed at effectively capturing and presenting the spatial and temporal evolution of data, many of which are directly applicable for CERN's visualization tool to analyze and interact with complex machines that change over time.

Table 3.10 summarizes the key visualization techniques, their underlying methodologies, the supported transaction types, time primitives, and notable features. By incorporating one or a combination of these visualizations, while trying to fill the gaps of utilizing TDOI in a full-fledged visualization system and supporting overlapping models, the work of this dissertation may be able to effectively support the understanding and analysis of dynamically evolving models, ensuring clear and interactive insight into key spatio-temporal changes.

Table 3.8: Reviewed records related to 4D visualization techniques.

Reference	Primitives	Reference	Variables	Mapping
Andrienko et al. (2003)	Intervals	Spatial	Multi-D	Mixed
Farrimond and Hetherington (2004)	Instants	Spatial	Multi-D	Linear
Bach et al. (2014)	Instants	Both	Both	Linear
Brazauskas et al. (2021)	Instants	Spatial	1D	Linear
Carvalho et al. (2008)	Intervals	Spatial	Multi-D	Linear
Cheymol et al. (2022)	Instants	Abstract	1D	Linear
Gautier et al. (2023)	Intervals	Spatial	Multi-D	Linear
Guo et al. (2014)	Instants	Abstract	Both	Linear
Piringer et al. (2004)	Instants	Abstract	Multi-D	Linear
Hertel et al. (2009)	Intervals	Spatial	Multi-D	Linear
Ivson et al. (2018)	Intervals	Spatial	Multi-D	Linear
Ivson et al. (2020)	Intervals	Spatial	Multi-D	Linear
Jaillot (2020)	Intervals	Spatial	Multi-D	Linear
Kim et al. (2017)	Instants	Abstract	Multi-D	Linear
Li et al. (2009)	Instants	Spatial	Multi-D	Linear
Mota et al. (2023)	Instants	Spatial	1D	Linear
Ponchio and Hormann (2008)	Instants	Abstract	Multi-D	Linear
Tsang et al. (2004)	Intervals	Abstract	1D	Linear

Table 3.9: 4D visualization techniques from the selected records.

Reference	Techniques
Aigner et al. (2023)	<i>Literature review</i>
Andrienko et al. (2003)	Both
Farrimond and Hetherington (2004)	Changing color, size and position
Bach et al. (2014)	Space-time Cube operations
Brazauskas et al. (2021)	Annotations and color changing
Carvalho et al. (2008)	Transparency and color changing, superimposition
Cheymol et al. (2022)	Multi-function slider
Gautier et al. (2023)	Space-time Cube operations
Guo et al. (2014)	Mainly 2D
Piringer et al. (2004)	Size, color and outline changing
Hertel et al. (2009)	Fading
Ivson et al. (2018)	Cascading views
Ivson et al. (2020)	Highlights and annotations
Jaillot (2020)	Transparency and color changing
Kim et al. (2017)	Juxtaposition, superimposition, interchanging, encoding
Li et al. (2009)	Colored labels
Mota et al. (2023)	Embedded and linked views, juxtaposition.
Ponchio and Hormann (2008)	Animation
Tsang et al. (2004)	Aggregation/encoding

Table 3.10: Summary of visualization techniques for time-evolving data

Reference	Visualization Technique	Transactions	Primitives	Highlighted Features
Carvalho et al. (2008)	Temporal Focus + Context (TDOI)	Creation, modification, destruction	Valid time	Uses a Temporal Degree of Interest (TDOI) to dynamically adjust transparency and color to highlight changes while preserving spatial-temporal context.
Kim et al. (2017)	Juxtaposition, Superimposition, Interchangeable, Explicit Encoding	Creation, modification, destruction	Valid time	Presents four spatial comparison techniques, enabling users to analyze differences across versions or temporal evolution of 3D data.
Ivson et al. (2020)	Highlight, Visibility, Cascade Animations, Annotations	Task creation and status tracking	Valid time	Focuses on BIM and introduces animations that effectively convey temporal information for progress tracking and task scheduling.
Bach et al. (2014)	Space-Time Cube Operations: Colored Time Flattening and Juxtaposing	Creation, modification, destruction	Instant, period	Combines multiple temporal snapshots into visualizations such as slices, overlays, or collapses for easy detection of patterns and differences.
Gautier et al. (2023)	Space-Time Cube with transaction visualization	Creation, modification, destruction	Valid time	Represents evolution in 3D city models by stacking temporal layers with visual links and color coding for various transaction types.

Chapter 4

Problem Approach

Having explored the necessary knowledge and insights, this chapter presents the methods employed to address the research problem of integrating a temporal visualization into CERN’s web view system for models of complex machines. The approaches taken to design, plan, and implement the solution are outlined here. Appearing first is the definition of the problem, in the context of CERN and the current visualization tool, and identifying the key challenges. The designed solution is then detailed, focusing on the architectural and technical choices made to incorporate the temporal visualization. Finally, the chapter concludes with a risk analysis, identifying potential challenges and proposing mitigation strategies to address technical and organizational uncertainties. Together, these elements provide a roadmap for achieving the research objectives.

4.1 Detailed Problem Definition

At CERN, scientists plan and execute experiments on very large and complex facilities. To ensure that the machinery works correctly, maintenance tasks are performed frequently on sections of the machines. Currently, a 3D web-based tool is used to visualize which sections are under maintenance and which groups are performing those operations (as depicted in Subsection 2.1.3). An important aspect not being taken into account by the available tool is that there are tasks that add new components and machines or that remove them, which implies that the corresponding 3D model should also be updated to reflect those changes. These tasks usually occur during long shutdown periods. Being able to visualize, for any selected day, which transformations are being performed on the machines is particularly useful to the workers who are following the maintenance planning. To achieve this, the previous 3D visualization tool must be expanded to display the machine’s components that are valid for the selected time.

4.2 Proposed Solution

The proposed solution leverages the four-dimensional KD-Tree, as already experimented in Hertel et al. (2009), to create a system for managing temporal models. A KD-Tree is a ‘space’ partitioning

data structure where the space is actually 'K' dimensional. Using this structure, it is possible to create a tree with different versions of a model and then associate each node with a time interval so that it can later be traversed to find which nodes are within a queried time interval. In this case, the tree leaves contain the 3D objects that compose the original 3D models.

A KD-Tree is a straightforward data structure to implement and adapt, with widespread examples available. It is created by receiving at least two different versions of a model. Since each version has an associated date and each 3D object that will be part of the tree comes from one of the versions, it is possible to set the start and end dates of each object in the tree. At the moment of the tree's creation, the objects should already have associated any tasks that they can be a part of, so that the start and end dates of the components can be further altered. All objects are loaded in memory and are selectively shown or hidden according to the date the user has selected.

To visualize changes in the machines, a Temporal Degree of Interest (TDOI) is employed, as described in [Carvalho et al. \(2008\)](#). This feature intends to ensure clarity and make it easy to distinguish between significant and minor changes in the objects. This mechanism shall include a time window of interest that the user can adjust. This affects the opacity and color of the models displayed. LODs can be used and incorporated, and directly controlled by the KD-Tree or managed by the rendering system. Both options are explored.

With this approach, it is expected that workers at CERN can visualize and easily interpret changes in the constitution of facilities and machines, with minimal visual lag and a stable frame rate. That is expected to prove helpful in reducing the time spent analyzing and planning maintenance tasks and construction operations. This work utilizes state-of-the-art methods to manage and visualize changes in the models. It adds a TDOI to the visualization of machine components, whereas none of the approaches used to model evolving 3D models used a similar technique. It also applies an approach first used with urban modeling on large-scale machines, even though the work in [Ivson et al. \(2018\)](#) also focused on industrial applications and machines. Another relevant aspect is that the planning may undergo changes, so the trees may not simply be stored for later use.

Figure 4.1 represents the flow of information of the proposed solution. While similar to the corresponding diagram presented in [Nunes \(2023\)](#), where task information is created by a Facility Coordinator in Microsoft Project, and is synchronized with the Scheduling Tools database daily to display the tasks in the interactive 3D tool; it now takes into account that there must be someone (they can also be a Facility Coordinator) making the correspondence between the names of the equipment present in the tasks and the references of the equipment in the 3D models. The correspondence is made in a table that is associated with MS Project, but is editable on the Scheduling Tools website. This requirement is new because the models of the facilities that are being used now are structured differently than the ones used previously by Nunes.

The information visualization pipeline for the temporal evolution of machines and display of maintenance task data consists of four main stages, as illustrated in Fig. 4.2. Starting with raw maintenance records from the planning as well as the entire set of components for a given machine or facility, the data analysis stage transforms this information into structured data in the

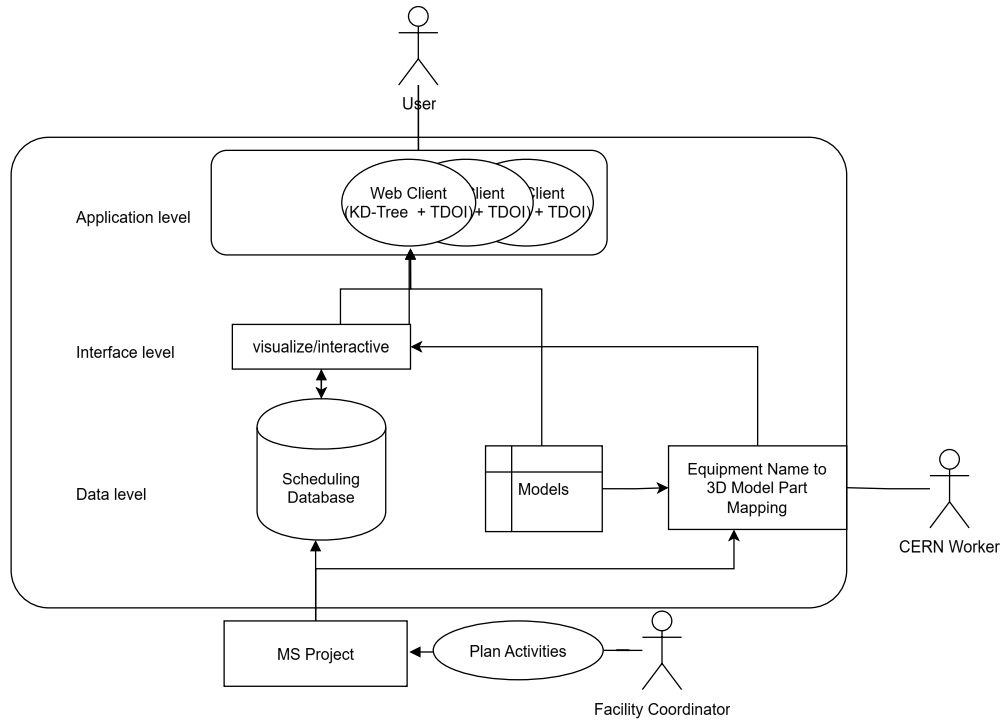


Figure 4.1: General architecture of the proposed web visualization tool. Adapted from Nunes (2023).

form of a 4D tree. During this initial stage, the system organizes components into a hierarchical structure, associates maintenance task data with specific components and timestamps, and preserves maintenance group information for later visual encoding. The filtering stage implements temporal and spatial selection mechanisms. Users can specify time intervals of interest and select specific components, either individually or in groups. A key aspect of this stage is the calculation of Temporal Degree of Interest, which considers the selected time interval and may contemplate the hierarchical relationships between components. This TDOI value becomes crucial for subsequent visual representation decisions. In the mapping stage, the system translates filtered data into visual properties. Maintenance groups are assigned specific color schemes (previously defined by CERN) for clear differentiation. The previously calculated TDOI values influence both transparency levels and color. The final rendering stage implements these visual mappings through ThreeJS and custom shaders. The shader parameters are configured to handle color assignments and transparency values. This approach, extended for temporal models, ensures efficient rendering while maintaining visual continuity as users explore different temporal and spatial aspects of the maintenance data. This pipeline enables interactive exploration of machine maintenance data while aiming to maintain temporal context through appropriate visual encodings.

In summary, this solution offers, if implemented correctly, a scalable, efficient, and user-focused system for CERN workers that manages temporal models and displays their evolution. This is expected while ensuring compatibility with existing standards and addressing some perfor-

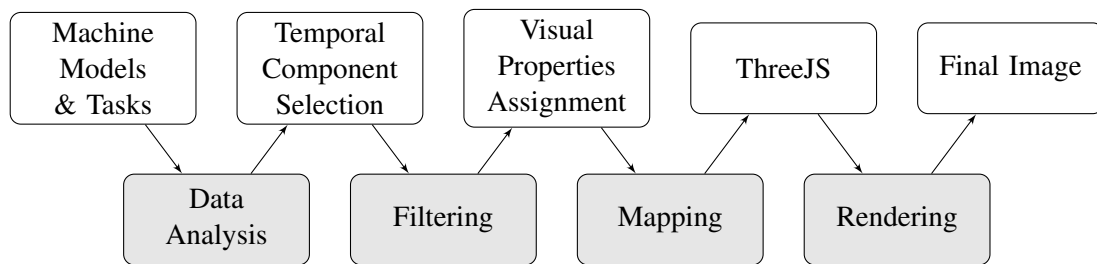


Figure 4.2: Information visualization pipeline of the proposed system.

mance issues and usability challenges. At the same time, it aims to expand on the work of previous authors by combining different techniques in the same system.

4.3 Risk Analysis

The main identified risks are:

- **R1: Using unfamiliar libraries/tools** – This risk is likely to affect development and could cause delays. To mitigate this, it is essential to read the available documentation, prototype early, and use a methodical approach when troubleshooting and debugging.
- **R2: Unavailability of test data** – Although less likely, the final prototype may offer reduced utility to its intended users if some scenarios are not taken into account. To address this, the approach involves developing minimal mocks to facilitate testing and validation when data is unavailable.

Although less probable, the final prototype may offer limited utility to its intended users. To mitigate this, the approach involves developing minimal mock-ups to facilitate testing and validation when data is unavailable.

- **R3: Insufficient testing** – This can lead to a poor evaluation of the project’s contributions. The mitigation plan includes defining tests early and arranging testing sessions in advance to ensure thorough validation.
- **R4: Scalability/performance issues** – This risk may make the developed system difficult to use. To mitigate this, frequent experimentation with high volumes of data, particularly models representing large machines with multiple modifications, is recommended.
- **R5: Underestimating model organization efforts** – This could delay the development of visualization techniques. The mitigation approach involves defining exact requirements and planning ahead, with a backup solution of implementing simpler versioning at the cost of performance.
- **R6: Integration delays** – This is not unlikely and could result in development delays. To minimize this risk, studying the current implementation and ensuring a modular development approach is crucial.

Table 4.1 presents a summary of these risks, together with their likelihood and possible impacts both on the schedule and on the end result. Working in person at CERN for the bigger part of the active development period was also identified as possibly bringing some issues, such as communication with other teams and the availability of certain data and APIs, however, this was not an issue.

Table 4.1: Summary of likelihood and impact of each identified risk.

Risk	Likelihood	Impact
R1	High	Delays
R2	Low	Reduced usefulness to final users
R3	Medium	Poor evaluation of the contribution
R4	Medium	The developed system is not easy to use
R5	Low	Late development of visualization techniques
R6	Medium	Delays

At the end, the resulting developed program must be deployed on CERN's website. At this point, it is critical to have solved most if not all performance issues, otherwise users will find the system not useful and it may even make their workflow worse as people were already used to the previous system. As the risk likelihoods are re-evaluated over time, so must be the impacts and mitigation strategies. For example, gathering users for testing is a problem that was supposed to have been solved by the middle or end of April, since some development must have occurred first and it is not possible to start inviting users for testing early on in the development phase. In reality, participants only started being invited in May, but that did not compromise in any way the number of responses as the target of having 25 participants was still reached.

Chapter 5

Implementation

This chapter documents the development of a prototype that aims to tackle the previously defined problem. Initially, the pipeline created in Nunes (2023) suffers some updates, then the performance is taken into account and some optimizations have to be introduced, and finally, temporal models are created using a 4D tree and maintenance tasks are visualized with a time window mechanism.

5.1 Model Preprocessing

The starting point for the development of the work planned is the already existing code, pipeline, and documentation devised in Nunes (2023). The original preprocessing pipeline utilized Unreal Engine to convert the CAD models to glb and Simplygon to simplify the models.

However, the pipeline had to be altered (by changing the name of a meta-property), as shown in Fig. 5.1. The provided script in Unreal Engine had to be updated to be correct in version 5.5, otherwise the execution resulted in an error. Even then, Unreal Engine performs poorly on the used computer with frequent crashes or freezes (mainly due to the rendering of the output preview, which does not seem optimized for older graphics cards without ray-tracing capabilities) that would prevent or interrupt the conversion process. Therefore, CAD Assistant¹ was chosen to replace Unreal Engine. It is free, but as highlighted in Nunes (2023), the conversion in CAD

¹<https://www.opencascade.com/products/cad-assistant/>

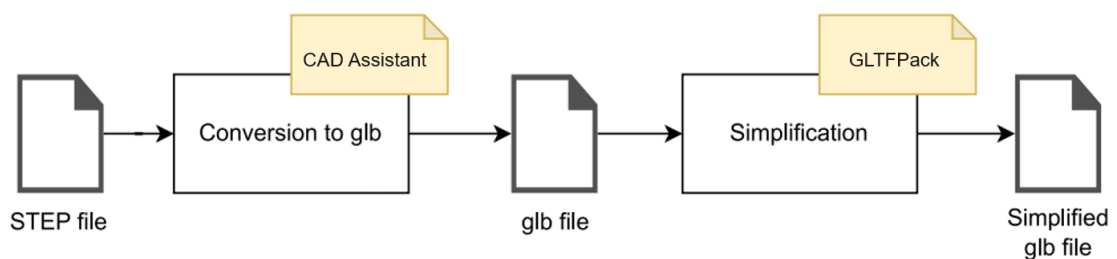


Figure 5.1: New preprocessing pipeline. Adapted from Nunes (2023).

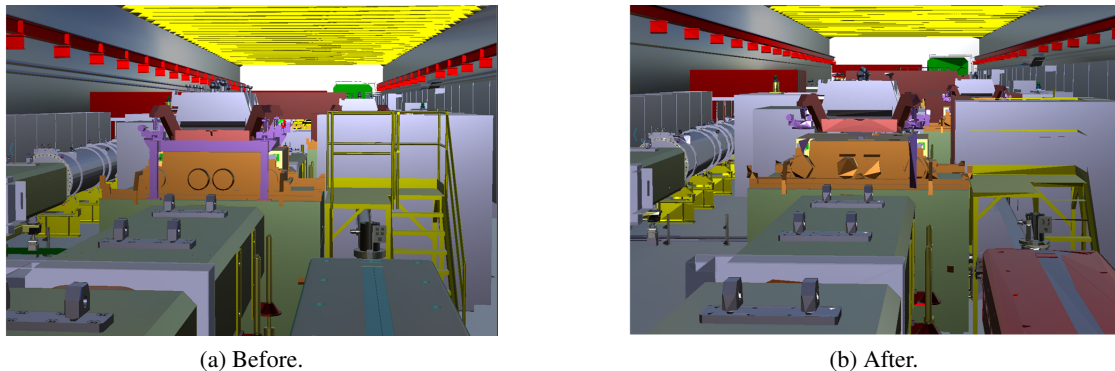


Figure 5.2: TCC2 models before and after simplification using a 10% target.

Assistant «is a manual process and doesn't allow scripting. Also, it doesn't provide any options to optimize the tessellation process.». Even if it does not allow scripting, the full conversion process comprises only a few steps and a few clicks. The lack of a customizable tessellation process (the conversion of a mathematically defined surface to a polygonal surface) has also not presented itself as an inconvenience. In Unreal Engine it is possible to set the maximum tolerance values and maximum lengths of the cords in the resulting geometry. These tessellation parameters control the quantity of polygons generated for the surfaces. For the same TCC2 model (an old version), using CAD Assistant results in files with more than three times the number of polygons than a file converted using Unreal Engine. This is, however, before simplification. After the simplification (for the same reduction target), the model originated from CAD Assistant retains only 37% more polygons than the one converted in Unreal Engine. Therefore, if possible, it is best to use Unreal Engine to convert the models from step to glb.

The simplification software Simplygon also needed to be replaced, as there is no longer a free license to use the program. At first, the search for an alternative to simplify glb/gltf files lead to gltf-transforms². Its scripting packages were tested to simplify glb models, but the level of simplification was underwhelming, and the file size of the simplified version was still too big at more than 50 MB. The file used is a model of the TCC2 building of the North Area facility at CERN. The original file size (after conversion to glb using CAD Assistant) is 178 MB and contains 6.8 million triangles. Later, since it appears to be the alternative adopted by a portion of the online community that stopped using Simplygon, the trial version of Polygon Cruncher³ was tried and the results obtained were very satisfying; however, after the trial version expired, the ability to export the models was lost. Finally, after revisiting gltf-transforms, the underlying library that is used to perform the simplification was identified: the "meshoptimizer" library. Using Gltfpack⁴, an open-source command-line interface program (based on the "meshoptimizer" library⁵), the desired simplification targets (less than 25 MB) were achieved with good visual quality.

²<https://gltf-transform.dev/>

³<https://www.mootools.com/en/software/polygon-cruncher>

⁴<https://github.com/zeux/meshoptimizer/tree/master/gltf>

⁵<https://github.com/zeux/meshoptimizer>

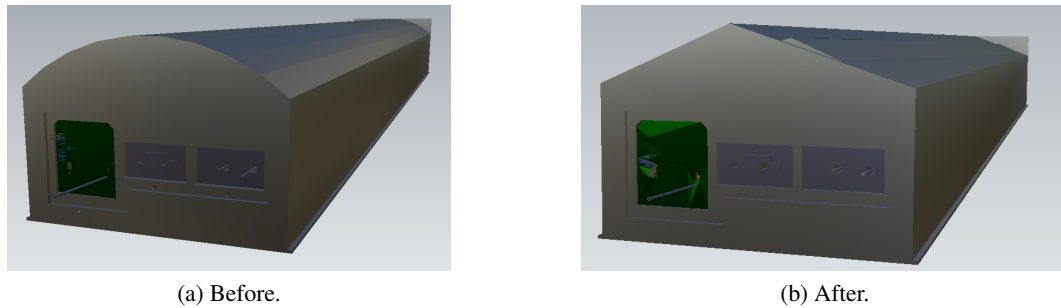


Figure 5.3: TCC2 exterior simplified with a 10% target using GLTFPack.

Figures 5.2a and 5.2b display a model of TCC2 before and after simplification, respectively. A target of 10% was used for the whole model, the original model is approximately 178 MB with 6.8 million polygons, while the simplified model is approximately 19 MB with 700 thousand polygons. Some objects like the yellow stairs lose significant detail, others become less round and more jagged, but for the most part, it is still possible to distinguish sub-components with a satisfactory level of visual fidelity. It is worth noting that the program was run with the options to keep named nodes and meshes, to use aggressive simplification, and to not lock the borders of the meshes. Mesh quantization, the process of converting vertex attributes to 8 or 16-bit values instead of floating-point values, can be disabled to reduce the file size slightly. This option was utilized until the resulting output was deemed satisfactory, as CAD Assistant does not support quantized meshes, whereas ThreeJS does.

A drawback of Gltfpack, when used in the command line, is that it is not possible to programmatically define different simplification targets which can result in under or oversimplified geometry of some meshes to achieve a satisfying simplification of the rest of the meshes. Figure 5.2, where the stairs are oversimplified, and Fig. 5.3, where the exterior mesh loses too much detail, are examples of that. For cases like the oversimplified exterior, it is easy in CAD Assistant to export the model with parts of it removed (by toggling the visibility) and so it is possible to combine, using 3D editing software like Blender⁶, the interior of a simplified model with the exterior of a not so simplified model. The only caveat is that Blender may export the combined model with a file size much larger than the sum of the combined parts, but this can be easily fixed by running Gltfpack again with a simplification target of 100% or 99%. This process is summarized in Fig. 5.4. Another possible alternative is to explore the meshoptimizer library and create a script that takes into account the dimensions of a mesh before simplifying it so that very large objects are not simplified too much. In addition, it is possible to specify different target errors, that is, the maximum error, in percentage, that is acceptable for the simplification of a mesh.

⁶<https://www.blender.org/>

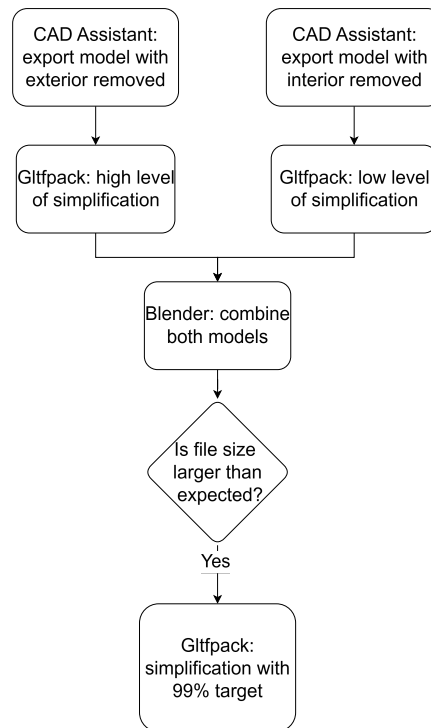


Figure 5.4: Steps performed to combine model parts (exterior and interior) with different simplification levels.

5.2 Optimizations

Even simplified, the detailed model of TCC2 is larger than any model tested in Nunes (2023). This made the frame rate stay most of the time between 9 and 15 frames per second. The direct cause of this poor performance was found to be the number of draw calls per frame, usually between 10000 and 20000. The number of draw calls is proportional to the number of different visible objects and meshes in the scene and, as ThreeJS loads the glb files, it creates one new object per node in the hierarchy of the model and one new mesh per leaf. To combat this issue, an optimization technique implemented in ThreeJS called "Batched Mesh" is used. A batched mesh combines every mesh of the original loaded object into a single one with a single geometry buffer that is ordered according to the addition of the meshes. Even then, it is possible to correspond each original mesh with its created geometry as a new ID is created with each addition to the batched mesh. This ID can be used to edit properties of the corresponding geometry, in this case, the visibility, color and transformations matrix. The main drawback of the batched mesh is that it only allows the use of a single material for the whole object, but this is not a problem in this case, since even if it is expected to have the need to instantiate objects with different materials (for transparency and outline effects), they can themselves be batched or merged meshes which will not increase the draw call meaningfully (always less than 10 in the scenarios tested). To maintain the full correspondence between the original hierarchy of the glb model and the batched mesh, a dictionary where the keys are the unique meshes and the values are the geometry ids in the batched

mesh is kept. With this, it's easy to select any of the original objects in the batched mesh even if they are made up of many meshes.

Further optimizations include disabling the sorting of the objects each frame according to their depth in the batched mesh (the batched mesh itself is static, so this does not have any drawback) and the addition of more Levels-Of-Detail using a community-made extension to ThreeJS (batched-mesh-extensions⁷ by "agargaro") which also uses the meshoptimizer library to automatically generate simplified LOD models based on the given simplification targets. Combining both increased the frame-rate approximately 30% . All of these optimizations increase the frame-rate on the PS (non temporal model) from between 50 and 60 FPS to between 120 and 144 FPS, compared with the previous version of the tool. The drawback is the increased load time of the models. The model of the first octant of the PS now takes 4.76 seconds, on average, to load, up 52% from 3.14 seconds. In both cases, the model is being loaded from cache in a scenario without tasks. The rise is due to the time taken to create the KD-tree. Loading the program and displaying TCC2 takes, on average, approximately 16 seconds. Only 1.5 seconds are spent creating the KD-Tree, while more than 13 seconds are needed to load into memory two versions of TCC2 (totaling 38.8 MB). The rest of the time is spent making API calls and traversing the scene to associate objects with tasks and dates.

5.3 Temporal Models

The first attempt at creating a structure that could manage 3D models and associate their components with different active time intervals was using 3D tiles, inspired by the work in [Jaillot \(2020\)](#), previously analyzed in Section 3.2. However, it was not clear how to programmatically divide a 3D model into several regions (spheres or boxes) because there would always be overlaps and some components span a big portion of the whole length, width or height of each model. Then, the idea was to associate each tile with an interval of time and then swap them according to the desired time. This would require the pre-creation of multiple versions of each tile, one per each change that occurred within the tile. As the planning at CERN is evolutive and extensive activity plannings for the facility tested do not exist yet, this approach is unmanageable and non-viable.

The chosen alternative is a tree structure, in particular a four-dimensional KD-Tree, such as the one in [Hertel et al. \(2009\)](#). A tree is a much more flexible data structure and, in this case, the tree needs to only be constructed at runtime, given information about the planning activities. As for the models, for any pair of versions that are to be compared, only the two full models are needed. Tree nodes contain both spatial and temporal bounds, and leaves contain also a list of meshes. Each mesh is previously associated with installation and removal dates which depend on which version of the model they are a part of and on if they are associated with any equipment installation or removal task. The tasks are fetched from an API endpoint and consist of lists of key value pairs, in json format, indicating the start and end dates, location or equipment, responsible groups, activity type (installation, removal or maintenance), and other details. With the equipment name

⁷<https://github.com/agargaro/batched-mesh-extensions>

from a task, another API endpoint (mentioned in Section 4.2) is queried to get the corresponding reference in the 3D model. The API selects the correct version of the model using the date. Having the correct reference, the scene is traversed in order to find the correct object. This process is performed once, when the models load, so all tasks that appear in the timeline are associated with the corresponding equipment.

In the mesh user data is also stored a list of unique task ids. As a task is associated with an equipment which is a 3D object, every sub-component and mesh of the object also stores the same task id. The tree is constructed naively so as to be built faster, splitting consecutively by the x, y, z and start time (installation date) dimensions. The decision to consider only the installation time was made because including the removal time did not yield any noticeable improvement, so the simpler approach was chosen.

For each facility with temporal versions, there is a configuration file that specifies the filepath and start date of each version. In this case, for a given visualization date, the closest previous version is loaded together with the closest future version. This configuration file must be manually updated if there are new versions and the models themselves are the simplified glb files obtained during the simplification step. There is no limit for the number of versions specified and stored. In spite of that, only 2 models can be compared at a time, which simplifies the program logic, and is enough considering that when the visualization loads it displays a timeline starting 2 months in the past and ending 2 months in the future. This time span is short enough to guarantee that there are not 3 (or more) distinct versions of any facility existing within the interval. This value is not adjustable by the user, so as to ensure that the user does not select an interval where 3 different versions of the facility coexist. The selection of a 4 months period was made based on the anticipated range of interest for the typical user, considering the tasks they most frequently need or wish to perform.

The KD-Tree is also essential to speed up the raycasting process used to select equipment. When a ray is created from the position of the camera directed to the position of the cursor, it is tested for collisions against the spatial partitions defined by the KD-Tree. The ray is tested against a node, and if it intersects the node's bounding volume, it is recursively tested against its children. Typically, only one child (left or right) contains the ray, allowing the traversal to quickly reach a leaf node. At the leaf node, the ray is tested against the actual meshes stored there, which are few in number, making the collision detection efficient.

The tree is constructed with two adjustable parameters: the maximum number of triangles per node, set as 120, and the minimum node size (measured as the amplitude in any of the spatial axes), set as 0.1. These values influence how many meshes end up in each leaf of the tree and, by consequence, their depth. When the tree is constructed with two versions of the TCC2 model, the average depth of the leaves is 21.7, and 76 thousand meshes are distributed over 31 thousand leaves, which gives almost 2.5 meshes per leaf, on average. If this number is significantly larger, the frame-rate is noticeably affected, as the raycasting operation is performed on every mouse movement and each mesh of the intersected leaves is tested against the rays.

5.4 Temporal Degree of Interest

A Temporal Degree of Interest (TDOI) mechanism was implemented, so as to have the possibility of viewing information about not only present but also past and future tasks, according to a configurable time window. The equipment that is never installed or removed has an active time interval from minus to plus infinity (or to the start date of a new version of the facility), installed components have a defined start time and removals set an end time for the components.

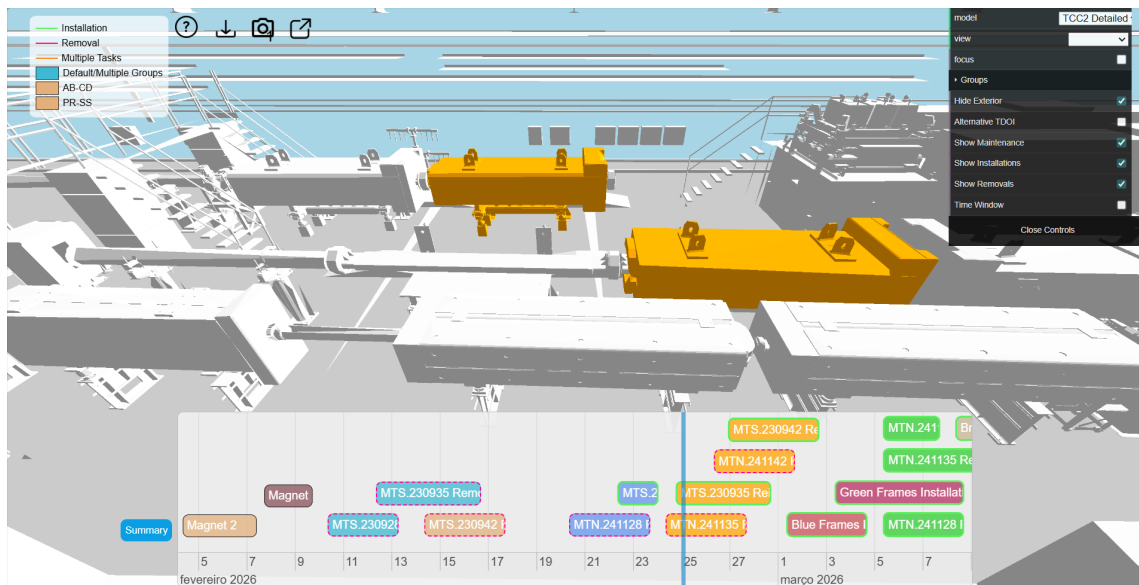
A first idea has been to use the TDOI mechanism to only convey information regarding installation and removal dates; however, in this case, installation and removal tasks have themselves a duration and it is important to be able to tell if a task is ongoing. Then, it is decided that the TDOI mechanism takes into account the type of task (installation, removal or maintenance), the interval when the task occurs and it also uses the color of the task. It was previously defined in Nunes (2023) that when any equipment contains an ongoing task it is painted with the color of the group that is responsible for the task, otherwise, the equipment appears white. The previous work defined that while a task is ongoing the color of the object does not change nor does it have any transparency effect, being desirable to keep this convention.

The TDOI is used to display visual information regarding the progress of the status of the task: before the start, ongoing, and after the end. To this effect, the opacity of the objects is changed to indicate recency or temporal proximity, like a progressive fade-in or fade-out effect. Likewise, the hue of the color always stays the same, but by changing its saturation and making it closer to white, it is possible convey information regarding temporal proximity too. The initial implementation calculates a value between 0 and 1, which determines either the object's opacity or the degree of color interpolation between the task's color and white, depending on the type of task and whether it is starting or ending.

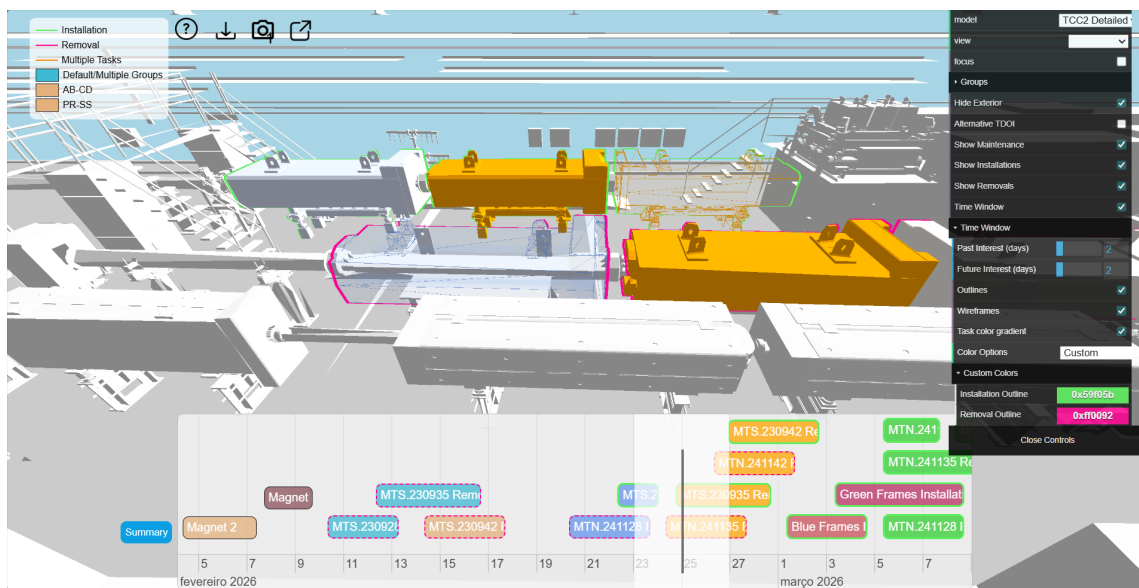
The previous "date slider" was replaced by a timeline, which now displays the tasks according to their start and end times. On the timeline there can be a time marker representing only an instant (Fig. 5.5a) or a temporal window (Fig. 5.5b). The presence of the window is what activates the feature. When the time marker is used, only the tasks that intersect it are shown on the equipment, whereas using the temporal window shows all tasks that intersect the interval. It is, however, possible to show all the tasks that occur during that whole day. That functionality has been kept that way for facilities without temporal models that do not use the TDOI window, such as the PS.

The TDOI function takes the following arguments:

- *isInstallation*: Boolean indicating if the task is an installation.
- *taskStartTime* (t_S): Start time of the task.
- *taskEndTime* (t_E): End time of the task.
- *centerTime*: Center time, position of the vertical time marker (not used in the first implementation).
- *timeInterval.start* (I_S): Start of the time interval.



(a) Timeline with time marker.



(b) Timeline with temporal window.

Figure 5.5: New timeline, vertically expanded to avoid overlaps.

- *timeInterval.end* (I_E): End of the time interval.
- σ : Standard deviation for the normal distribution, default value is 0.3.

The task duration and interval duration are computed as:

$$taskDurationTime = t_E - t_S, \quad (5.1)$$

$$intervalDuration = I_E - I_S, \quad (5.2)$$

$$denominator = \min(taskDurationTime, intervalDuration). \quad (5.3)$$

The normal distribution function used is:

$$normalDistribution(x, \sigma) = \exp \left(-0.5 \cdot \left(\frac{x - 1.0}{\sigma} \right)^2 \right), \quad (5.4)$$

where x is the normalized time difference, and $\sigma = 0.3$ by default.

The output is a pair $\{tdoi, changeOpacity\}$, where $tdoi \in [0, 1]$ determines the opacity or color interpolation level, and *changeOpacity* indicates whether to adjust opacity (`true`) or diffuse color (`false`).

For installation tasks, the first implementation of the Temporal Degree of Interest mechanism works as follows:

1. As the time window and the task interval start intersecting from 0 to 100%, the opacity increases.

Beginning of the task—task starts within the interval, ends outside:

$$\text{If } t_S \leq I_E \text{ and } t_S \geq I_S \text{ and } t_E > I_E, \text{ then} \quad (5.5)$$

$$tdoi = normalDistribution \left(\frac{I_E - t_S}{denominator}, \sigma \right), \quad (5.6)$$

$$changeOpacity = isInstallation. \quad (5.7)$$

2. While the intersection is total, the color of the equipment is the color of the task.

Default case—the task is fully contained within the interval or the interval is fully contained within the task:

$$tdoi = 1.0, \quad (5.8)$$

$$changeOpacity = \text{false}. \quad (5.9)$$

3. As the degree of the intersection drops, the color of the equipment progressive gets more white/gray.

End of the task—task starts outside, ends within the interval:

$$\text{If } t_E \leq I_E \text{ and } t_E \geq I_S \text{ and } t_S < I_S, \text{ then} \quad (5.10)$$

$$tdoi = normalDistribution\left(\frac{t_E - I_S}{denominator}, \sigma\right), \quad (5.11)$$

$$changeOpacity = \neg isInstallation. \quad (5.12)$$

Figure 5.6a exemplifies this. The task marked with "1" is fully inside the window so the magnet shows the color of the task in full opacity. Task "2" is ongoing but is not fully opaque as the interval does not fully intersect the temporal window. Similarly, the third equipment is also transparent; however, the corresponding task has not started yet.

A second version that takes into account the time marker of the temporal window works in a different way:

1. When the interval of a task intersects the window the equipment appears to go from fully transparent to opaque using the color of that task.

Beginning of the task—task starts after center time, within the interval:

$$\text{If } t_S > centerTime \text{ and } t_S < I_E, \text{ then} \quad (5.13)$$

$$tdoi = normalDistribution\left(\frac{I_E - t_S}{I_E - centerTime}, \sigma\right), \quad (5.14)$$

$$changeOpacity = isInstallation. \quad (5.15)$$

2. While the time marker intersects the task, the color does not change.

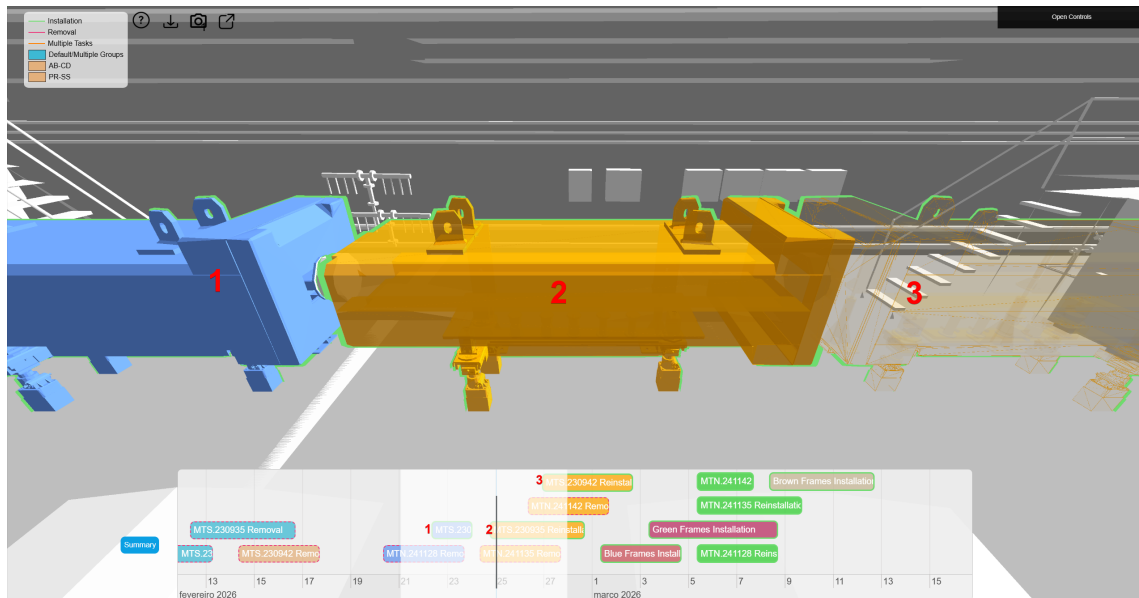
Task contains center time—if the center time is within the task duration:

$$\text{If } centerTime > t_S \text{ and } centerTime < t_E, \text{ then} \quad (5.16)$$

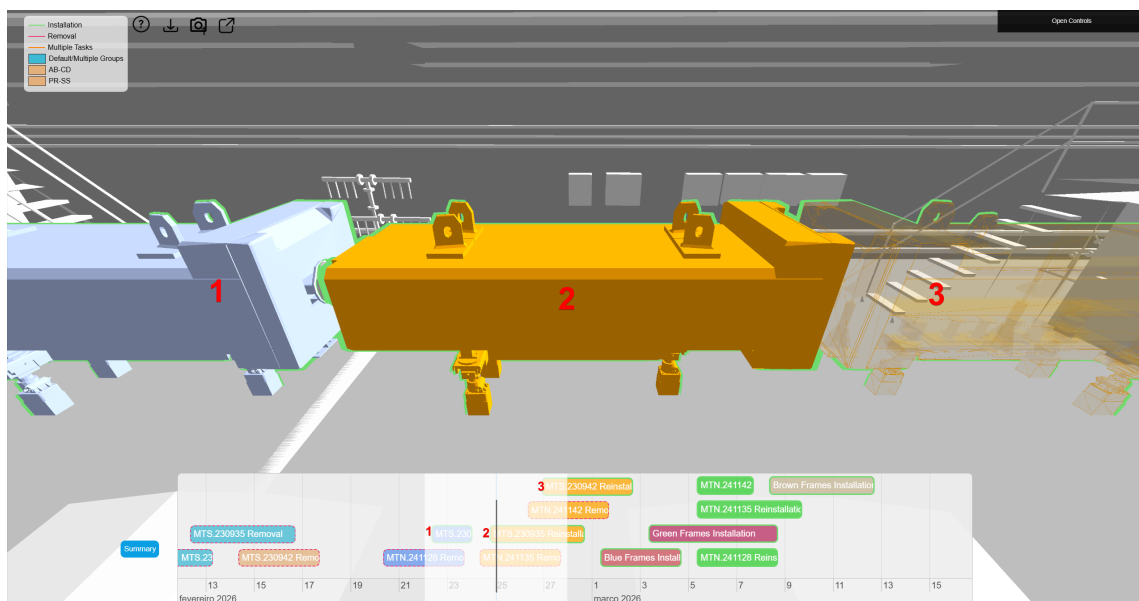
$$tdoi = 1.0, \quad (5.17)$$

$$changeOpacity = false. \quad (5.18)$$

3. When the intersection ends, and while the window still intersects the task interval, the color progressively gets more white/gray.



(a) First mechanism.



(b) Second mechanism.

Figure 5.6: Developed Temporal Degree of Interest mechanisms.

End of the task—task ends before center time, within the interval:

If $t_E < centerTime$ and $t_E > I_S$, then (5.19)

$$tdoi = normalDistribution\left(\frac{t_E - I_S}{centerTime - I_S}, \sigma\right), \quad (5.20)$$

$$changeOpacity = \neg isInstallation. \quad (5.21)$$

In Fig. 5.6b, the first installation task has not started yet, so the equipment has a color between white and the color of the task. The black vertical bar (current time marker) intersects the second task, so it is ongoing and the color of the equipment is fully opaque and not mixed with white. The last equipment is transparent, which indicates that the task will start in the future.

For removals the effects happen in reverse order and for maintenance activities the opacity of the equipment never changes, only the color both at the beginning and end of the task. The second implementation became the default option but with the possibility of toggling the first method.

An issue remained that it was not possible to know, at a glance even if it had a color or transparency effect, whether an equipment was being installed or removed, so colored outlines that distinguish between these types of activities were added. Maintenance tasks do not possess an outline. Figure 5.7 presents a summary with possible representations of the equipment in different phases and different types of tasks.

When the temporal window intersects more than one tasks on the same equipment, an orange outline is shown to serve as an alert. As it is not as simple to visually indicate everything that is going on when there are multiple tasks, the user is expected either to reduce the size of the window, turn it off completely or inspect the details of the tasks to fully understand what is happening during that period of time. The color and transparency effects of the equipment will be according to one of the tasks (with no guarantee of order) and ignore all the others so there is no visual superposition.

The outlines were created using the inverse hull technique where a slightly scaled copy of the object is shown behind the main object (by only displaying the back faces). This would usually be enough, but as the object can become transparent, the outline object should not be visible behind the object, only around it. To solve this, the stencil buffer was used so that the outline object was not drawn where the main objects were present. A postprocessing technique was previously tested and, while the outlines themselves looked better, the performance was noticeably worse as that technique implied rendering the whole scene twice. Using the stencil buffer approach present in Fig. 5.8, first everything except the outline objects are drawn, but only the objects with outlines update the stencil buffer. After that, only the outline objects are drawn, according to what is present in the stencil buffer.

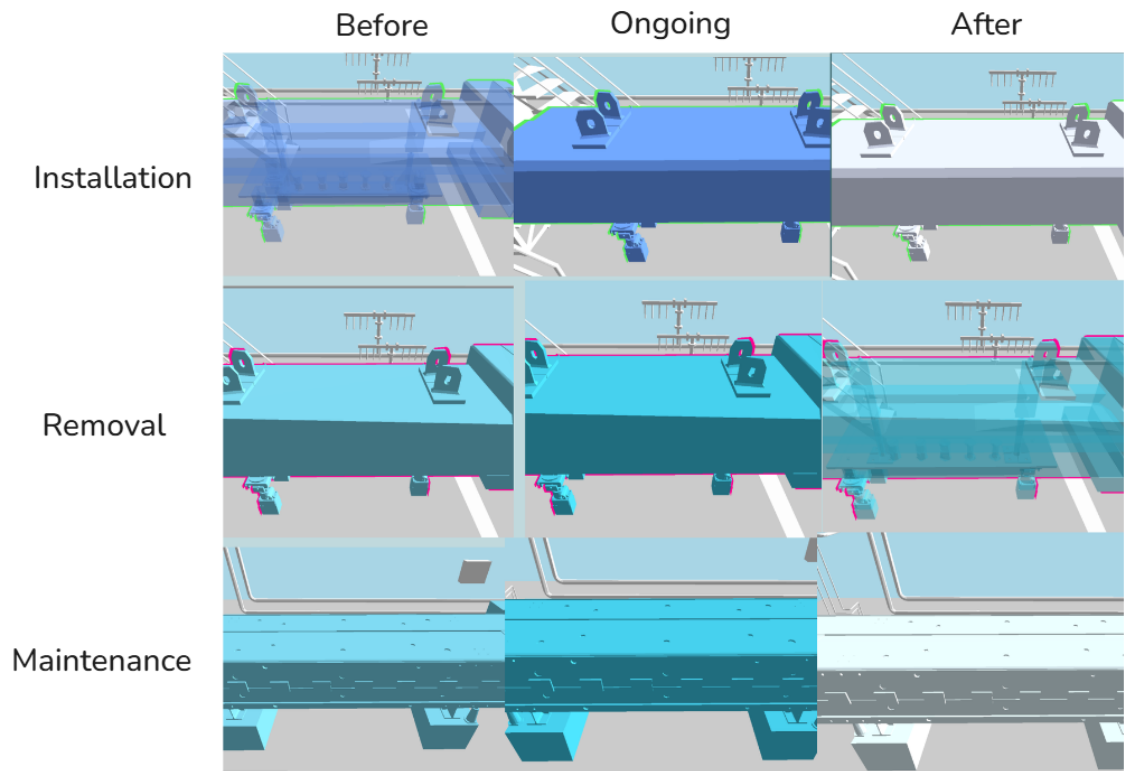


Figure 5.7: Summary of visual status of equipment according to the Temporal Degree of Interest.

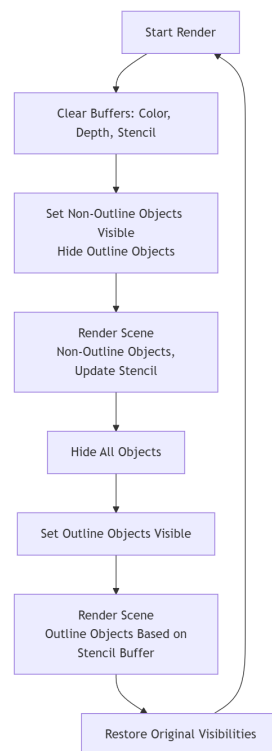


Figure 5.8: Rendering loop.

5.5 Interactivity and Graphical User Interface

To improve the navigation of the scene, new controls were added, namely the *Map Controls* and *Pointer Lock Controls* from ThreeJS. Movement with the keyboard (arrows or WASD keys) is also supported now. Regarding the Graphical User Interface (GUI), besides the already mentioned timeline, there are options to toggle the temporal window, the outlines, and the wireframes. It is possible to hide the walls and to filter tasks both by type and by group. The temporal window is adjustable in each direction (past and future) using integer values between 0 and 14 days. There are options to change the colors of the outlines (two presets and one option which allows the selection of custom colors) and the ability to duplicate the current view (facility and date) into a new tab is now present. This can be useful to open the tool in fullscreen as it is meant to run inside an 'iframe' in the Scheduling Tools website.

Old functionality and interactions remain, namely the selection of models and cameras through the GUI, and the ability of double clicking components to load detailed views (when on a global view) or to open a pop-up with task details (when already on a detailed view).

5.6 Challenges

There were some challenges faced when trying to adapt the implementation to CERN's specific case. The main one is that there is not a comprehensive correspondence between the names of the elements in the 3D models and the names of the equipments that are referred in the tasks. Another one is that some equipment is duplicated in the 3D model, which may cause issues in the future, and some components do not have a unique name, which appears to only apply to small and "irrelevant" components so there is not a problem with that, for now. The solution to the first issue was reached by other members of the section and consists in populating a table (available as an API endpoint) with the mapping between the names of the equipment and their corresponding references in the 3D model. This table is not filled automatically and it is also necessary to specify the version of the model, as different versions may label the same equipment with different references. It is, then, important to note that to be able to select any component of the 3D model, it is crucial to be provided with their names or any other unique identifier.

This implementation is expected to be able to scale for even larger models and, to test this, a small section of the LHC was exported from CATIA as two different models, as it was not possible to export them as a single model. Together, they exceeded 700 MB before simplification and 40 MB after simplification. From them, a simplified global view was created, and it is possible to select each of the 2 models and to load the corresponding detailed view. This is, however, what had been done already in Nunes (2023). Extending the usage of this tool to very large models may still be a challenge as there may be even larger files or models that do not simplify very well, but, as of now, the current approach is likely to continue working.

5.7 Summary

The preprocessing pipeline for 3D model conversion and simplification, initially built in Nunes (2023), required some updates that didn't compromise the results. Loading optimizations tackled the performance bottlenecks of the detailed TCC2 model, which was larger than those used in the previous version of the tool. The high number of draw calls per frame dragged the framerate to 9–15 FPS. By taking advantage of ThreeJS's Batched Mesh technique, which consolidates multiple meshes into a single geometry buffer, draw calls were drastically reduced, boosting performance. Additional enhancements, like disabling depth sorting and integrating a community-made extension for automatic Level-of-Detail (LOD) generation, further improved frame rates approximately 30%, though, overall, initial loading times have increased at least 50% due to the creation time of the KD-Tree. These optimizations maintained visual fidelity while ensuring smoother rendering.

Temporal model management uses a four-dimensional KD-Tree constructed at runtime with nodes defined by spatial and temporal bounds. This tree structure handles two model versions at a time, aligning with the visualization's four-month timeline window. Each mesh stores installation and removal dates tied to specific tasks, enabling dynamic updates without pre-creating multiple model variants. This approach proved simple and adaptable, but can still be optimized to achieve better performance or replaced. As the tree is built at runtime according to the tasks present in the selected 4 month period, it is not possible to store them server-side. Since the tree stores references to the loaded 3D models, it is also not straightforward to cache it locally. The Temporal Degree of Interest mechanism enhances visualization by highlighting past, present, and future tasks within a configurable time window. Replacing a basic date slider with a timeline, TDOI adjusts object opacity and color based on task type (installation, removal, or maintenance) and their temporal overlap with the window. A refined second implementation, which became the default, uses a time marker for smoother transitions, with colored outlines distinguishing task types and alerting users to overlapping tasks. While the stencil buffer technique for outlines ensured performance over postprocessing alternatives, challenges remain in visually clarifying complex task overlaps, requiring users to adjust the time window or inspect task details for full clarity. Figure 5.9 depicts a visual summary of the visualization pipeline and its three phases. The models are preprocessed, then are used to build a tree structure according to information regarding maintenance tasks and, finally the information is visualized, with or without the aid of the TDOI mechanism.

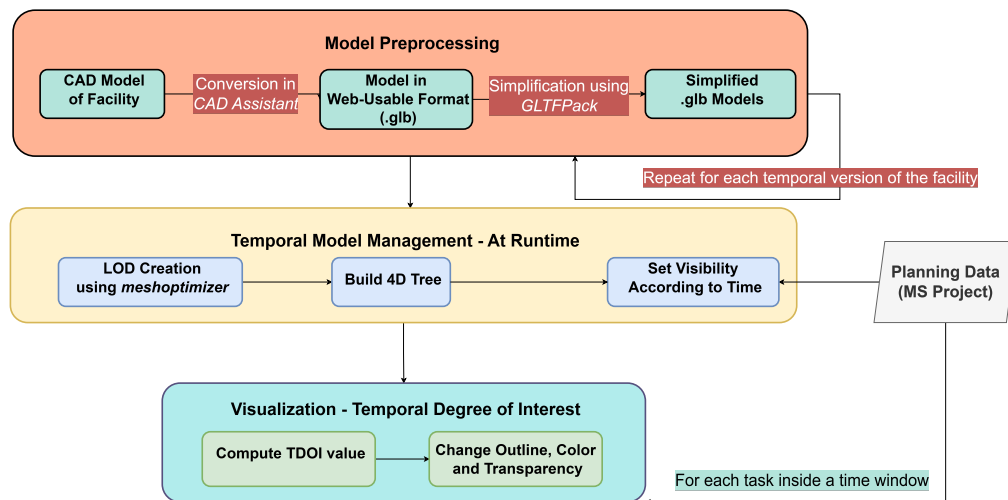


Figure 5.9: Visualization pipeline summary.

Chapter 6

Experimental Protocol

This chapter outlines the experimental protocol designed to evaluate the usability and effectiveness of the developed prototype of a temporal visualization tool developed to represent the evolution of 3D models for machines and facilities undergoing maintenance tasks. The primary objective is to assess whether users can intuitively interact with the tool to comprehend the status and temporal progression of ongoing tasks. The experiment involves CERN members, both experienced and novice users, engaging with a prototype to navigate, manipulate a timeline, and extract task-related information, with findings validated through structured assessments.

The protocol encompasses a detailed methodology, including test procedures, participant recruitment, and evaluation methods such as task-based questionnaires and the System Usability Scale (SUS), established in Brooke (1996). The experiment leverages a web-based platform to test the tool's compatibility with user objectives, focusing on its intuitiveness and functionality. The full testing script is present in Appendix A.

6.1 Test Methodology

The main goal of the experiment is to test the usability and understandability of the developed temporal visualization tool. The findings will be very important to answer **Research Question no. 3**, which pertained to the clear and unambiguous perception of the evolution of 3D models representing machines with ongoing maintenance tasks. To this effect, users should be able to analyze the models, move around and change the time using a timeline in an effective way.

6.1.1 Overview

The experiment is performed to learn how users engage with the tool and how compatible it is with their objectives by obtaining answers to the questions:

- **Question 1:** - Can users understand the status of the ongoing tasks?
- **Question 2:** - Can users perceive the evolution of the machines and facilities with regards to what changed and over what period of time?

- **Question 3:** - Is it easy for users to interact with the visualization and achieve the results they want?

It is worth noting that, as a new iteration of an already existing visualization tool, this experiment does not intend to retest unchanged features from the base visualization.

The prototype is validated by CERN members. Both by people that use or have used the previous version of the tool and by people that are using it for the first time. Testing with people unfamiliar with the tool is an opportunity to measure its intuitiveness, as little to no knowledge about the facilities or the visualization itself is needed beforehand. Nevertheless, testing with the end users is crucial to refine the system. The test is performed with the model of the building TCC2 using task data created specifically for this test.

The following outlines the fundamental aspects of the experiment, encompassing its accessibility, requirements, and evaluation approach:

- **Where and When:** in person, at CERN, during the week from May 6th to May 28nd.
- **Equipment and Software:** a computer with a web browser and access to the Internet. The program is hosted on the development version of the relevant CERN webpage. The browsers *Safari* and *Firefox* are not recommended because of incompatibility and/or performance issues. Firefox performs worse since it does not support the "WEBGL_multi_draw" extension¹, which is used by the batched mesh feature, so ThreeJS relies on a slower fallback. Safari, however, may run the program without issues or may not load it at all, depending on the model of computer used. This unreliability was observed but not thoroughly tested or confirmed using other sources.
- **Coordinator, observer, and facilitator:** the researcher, with help from Afonso Caiado from CERN, is the one conducting and facilitating the majority of the usability testing sessions, with additional responsibility for scheduling.
- **Participants:** only CERN members were considered, as the tests need to be conducted in person. Participant recruitment is supported by the supervisor Fernando Pedrosa, who contacted members of the EN-ACE-OSS section, workers in buildings 112 and 30, and some other CERN workers via email so that testing sessions could then be directly scheduled.
- **Assessment Methods:** a formative assessment through a questionnaire and a SUS form. The former is meant to evaluate whether the visualization is clear enough to answer questions correctly regarding the status and evolution of the facilities or machines. The latter measures the overall satisfaction with the usability experience. The SUS questionnaire provides a moderate amount of questions and it is varied and generic, which is suited for this experiment. Other alternatives may be too short or too specific, for example focusing on the utility of the program to satisfy all the work needs and use cases of specific groups of people, which is not the focus of neither the test nor of the tool.

¹https://developer.mozilla.org/en-US/docs/Web/API/WEBGL_multi_draw

- **Help Resources:** an initial contextualization of the tool is performed and the user is encouraged to experiment with the navigation controls and with the timeline. It is explicitly explained what the outlines and the respective colors mean, as well as the fact that the color of the tasks and equipment represent the group that is performing the task. If a participant is not able to advance on a task, any previously explained concept can be clarified. However, not being able to answer after help is provided counts as a wrong answer.

6.1.2 Experiment Procedure

Participation in the experiment is voluntary and conducted individually, with direct interaction between the experiment facilitator and participants. The steps each participant must follow are detailed below:

1. The participants are initially asked to read and sign an informed consent form.
2. The first set of questions focuses on demographics: age, gender, experience with the tool and familiarity with 3D software.
3. The participants are directed to the facility selection webpage where they are asked to select the "North Area" facility and building "TCC2". The initial visualization date is also selected as February 1st 2026 to fall within a time window that contains the previously created test tasks.
4. Participants are let to freely use and explore the tool while being told how to move around and use the timeline. The meaning of the colored outlines is also explained at this time.
5. They are asked, sequentially, 15 questions about information that can be perceived by using the tool. To answer the visualization questions, the participants have to perform tasks to be able to obtain the necessary information.
6. Some feedback regarding user preferences about the outlines, wireframes and the TDOI mechanism is collected.
7. A standard SUS questionnaire is placed.
8. Three open-ended questions are included to gather feedback on the overall experience, the relevance of the TDOI mechanism, and any additional comments, feedback, or suggestions.

6.1.2.1 System Load

The prototype is reliant on client-side processing, so the performance is dependent on the device used, in this case, the computer of each tester. As most workers at CERN use a similar or equal laptop provided by CERN, most people are expected to be affected by load times and any issues with the frame-rate in the same way. Even then, deviations from the average are to be expected either when using slower or older computers or better performing ones like desktops. The researcher's

laptop is also available as a backup in case participants do not have a computer available or are having any problem loading the testing platform (such as access permissions).

6.1.2.2 Completion Criteria

The experiment is deemed complete once the participant signs the Informed Consent Form, answers all required sections of the questionnaire (pre-test, in-test, and post-test), and does not want to add any more comments.

6.1.2.3 Success Criteria

In the experiment questionnaire, each of the 15 in-test tasks is scored binarily: a correct answer earns 1 point, while an incorrect answer receives 0. Correct answers are predefined by the test conductors during preparation and most questions have predetermined options to choose from.

6.2 Assessment Methods

The main assessment methods consist of a task-based questionnaire and a SUS questionnaire. Before the main questionnaire, some demographic questions to characterize the testers and better understand the results are placed and after questions about the preferences of the user are asked as well as broader questions to gather all types of feedback.

The questionnaire starts with an introduction explaining its purpose, context, and terms (Informed Consent Form present in Appendix A.2), followed by a section collecting demographic details like age and experience using software to manipulate or visualize 3D models (Pre-test Questionnaire). Next, participants complete tasks to assess their interaction and understanding of the visualization (Task-specific Questionnaire,). They then evaluate the system's usability and provide open-ended feedback for deeper insights (Post-test Questionnaire). Later sections explore the questionnaire's detailed structure.

6.2.1 Pre-test Questionnaire

Participants are asked their age, gender, whether they have previously used the 3D visualization tool at *Scheduling Tools* and, on a 5-point Likert Scale, how much experience they have with 3D model manipulation and visualization through software .

6.2.2 Task-specific Questionnaire

In some of the questions, the participant is asked to select a pre-defined camera. The testers can still move around and any feedback regarding the navigation controls is welcome. However, that aspect had a larger focus during the tests performed in Nunes (2023), which validated the usefulness of a 3D interface in this context. Not only that, but the preset cameras are positioned in places where it is possible to visualize all of the equipment that is explored in the subsequent

tasks. This test focuses on the perception of the visual elements associated with the new temporal window and the TDOI mechanism, along with any supporting features, such as the timeline. To that effect, some questions concern the analysis and usage of the timeline, others emphasize the temporal window itself which can be moved and resized, and all of the task statuses (before the start, ongoing and after the conclusion) are tested at least once, for both installation and removal activities.

T1-3: *Using the “Camera 1” view, determine the number of ‘Installation’, ‘Removal’, and ‘Maintenance’ tasks starting in February.* - Tests the visualization’s ability to filter and display task types within a specific time frame.

Steps: Select the “Camera 1” view. Observe the timeline to count tasks starting in February or move the time marker along February and count the number of activities of each type.

Result: [T1] Maintenance: 3; [T2] Installation: 3; [T3] Removal: 6.

T4-14: *With “Camera 1” view, analyze equipment status within a 4-day time window for specified dates, using options: “no task,” “before/ongoing/after removal/installation/maintenance task,” or combinations.* - Tests the visualization’s ability to track equipment status and timeline interactions, and how the user perceives the color and transparency effects associated with the TDOI.

Steps: Select “Camera 1” view, hide timeline items by pressing “t,” and set a 4-day time window (2 days past, 2 days future). Move the time marker to the specified dates and assess equipment status. Optionally, seek help to locate components or move the window. Avoid double-clicking equipment to expand task details initially. Later, enable task expansion and press “t” to show timeline items. Adjust the “future interest” to 9 days for the final date.

Results:

- 02/02: [T4] MBN.23097 - ongoing maintenance.
- 03/02: [T5] MBN.23097 - before and after maintenance tasks.
- 11/02: [T6] MTS.230928 - ongoing removal; [T7] MTS.230935 - before removal.
- 14/02: [T8] MTS.230928 - after removal; [T9] MTS.230942 - before removal.
- 22/02: [T10] MTS.230928 - before installation.
- 26/02: [T11] MTS.230928 - after installation; [T12] MTS.230935 - ongoing installation; [T13] MTS.230942 - before installation.
- 17/02 (9-day future interest): [T14] MTS.230935 - before installation and after removal.

T15: *Set the timeline to 1st March, select “Camera 4” view, and determine the number of installations starting between 01/03 and 10/03.* - Tests the utility and intuitiveness of adjusting the width of the time window.

Steps: Set the timeline to 01/03, select “Camera 4” view, and hide timeline items by pressing “t.” Count installations starting between 01/03 and 10/03 without changing the date. Use the time window interval as a hint.

Result: 3 installations.

6.2.3 Post-test Questionnaire

The post-test questionnaire begins with questions regarding the preferences of the user:

1. Do you prefer the outlines turned on or off?
2. Which color option do you prefer for the outlines? Do you think having the option to set custom colors is necessary?
3. Do you prefer the wireframes turned on or off?
4. Do you prefer the default time window functionality or the alternate one (after explaining and demonstrating both)?

After, a standard SUS questionnaire composed of the following questions is placed (Appendix A.5.1). Each statement is rated from 1 to 5, where 1 means that the participant strongly disagrees and 5 that the participant strongly agrees with the sentence.

At the end, three open-ended questions are asked:

1. What were the worst parts of the experience?
2. Do you think that the temporal features of the system are useful to plan or execute maintenance activities?
3. Do you have any other comments, including recommendations for improvement or feature suggestions?

6.3 Pilot Test

A single pilot test was conducted on April 30th with a colleague from the EN-ACE-OSS section. The structure of the test was the same as the one presented previously; however, it contained 6 additional tasks. This test took approximately 45 minutes, which exceeded the maximum target of 30 minutes. The most important feedback given was regarding how hard it was to move around and point the camera to where the participant needed. With that in mind, the following improvements were made:

- Removed 6 tasks so that the duration of the test was reduced to 30 minutes.
- Changed the navigation controls from *Orbit Controls* (rotating camera around a target) to *Map Controls* (2D panning/zooming) and added the option to toggle *Pointer Lock Controls* (first-person camera rotation via mouse movement).

- Added the option to move (parallel to the ground) with the arrows or 'W', 'A', 'S' and 'D' keys.

A second pilot test had been planned, but as another improvement regarding the controls used to switch the navigation controls was not put online before the start of the scheduled user tests and the testing version did not suffer further changes, that test was also not considered a pilot test.

6.4 Summary

The primary objective of the experiment is to assess the tool's usability and understandability, addressing Research Question no. 3, which focuses on the clear perception of 3D model progression. Conducted in-person at CERN, participants interact with a web-based prototype visualizing the TCC2 building, navigating 3D models and manipulating a timeline to extract task-related information.

The experimental methodology involves a structured procedure where participants, after signing an informed consent form, complete demographic questions, explore the tool, and answer 15 task-specific questions to assess their ability to understand task statuses, perceive model evolution, and interact effectively. Testing is facilitated by the researcher, with support from CERN's Afonso Caiado, and utilizes a computer with a web browser accessing the development version of the CERN webpage. The main aspect being tested is the TDOI mechanism, to evaluate both its intuitiveness and usefulness.

Assessment methods comprise a formative questionnaire and a System Usability Scale (SUS) to evaluate both task performance and overall user satisfaction. The questionnaire includes pre-test demographic queries, task-specific questions to test visualization clarity, and post-test feedback on user preferences (e.g., outlines, wireframes, TDOI functionality).

A pilot test conducted with an EN-ACE-OSS colleague leads to refinements to the protocol. Initially exceeding the 30-minute target duration due to navigation difficulties, the pilot leads to the removal of six tasks and to a change in the navigation controls. These changes improve usability and reduce test time. The experiment's success criteria, based on questionnaire completion and task accuracy, provide a robust framework for validating the tool's performance.

Chapter 7

Results

This chapter presents and discusses the outcomes of the experimental evaluation of the prototype, involving 25 CERN members who interacted with a temporal visualization tool to explore the 3D model of a facility undergoing maintenance tasks. The findings reveal insights into the tool's usability and clarity in depicting task statuses through demographic questions, task-based questions, and a System Usability Scale (SUS) questionnaire. Organized into demographic summaries, task performance analyses, SUS scores, and correlations between demographics and performance, the results assess the tool's effectiveness and intuitiveness, informing potential enhancements for its application in maintenance planning. The answers are present in Appendix B.

7.1 Sample Characterization

The age distribution is as visible in Fig. 7.1. The youngest participants (2) were 23 years old, and the oldest was 63. Of the 25 testers, 19 were male and 6 female. 20 had never used the previous version of the 3D visualization tool, 4 had used the 2D version, and only 1 had used the 3D version and said they used it only a few times per year. The level of experience and familiarity is expressed in Fig. 7.2. There are more people with a greater than average experience level than those with less than average experience, and the average is 3.24. The participants are typical end-users of the tool, as it is meant to be used by all kinds of CERN workers.

7.2 Task Performance Results

The 15 tasks were evaluated based on correct or incorrect responses. Fig. 7.3 shows the frequency of correct responses per task.

7.2.1 Individual Task Analysis

An individual analysis can be made for each task:

Task 1 23 correct out of 25.

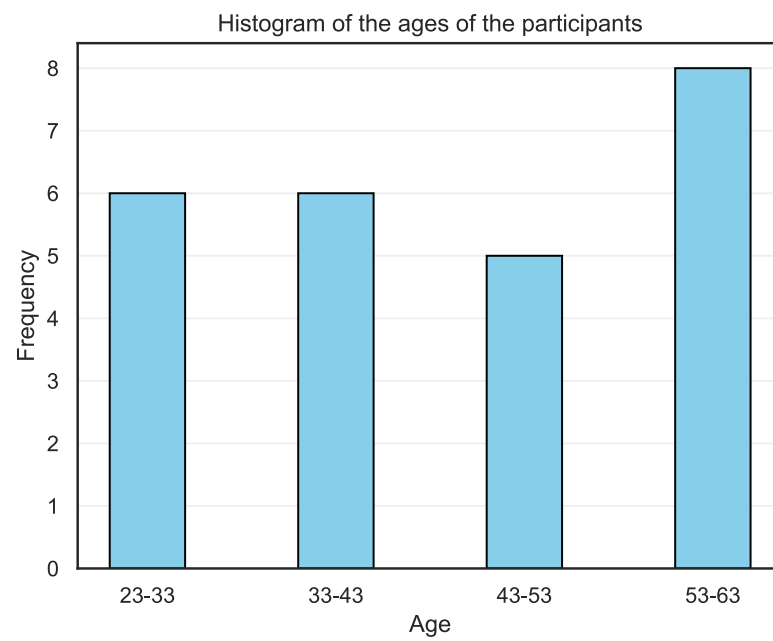


Figure 7.1: Histogram of the ages of the participants.

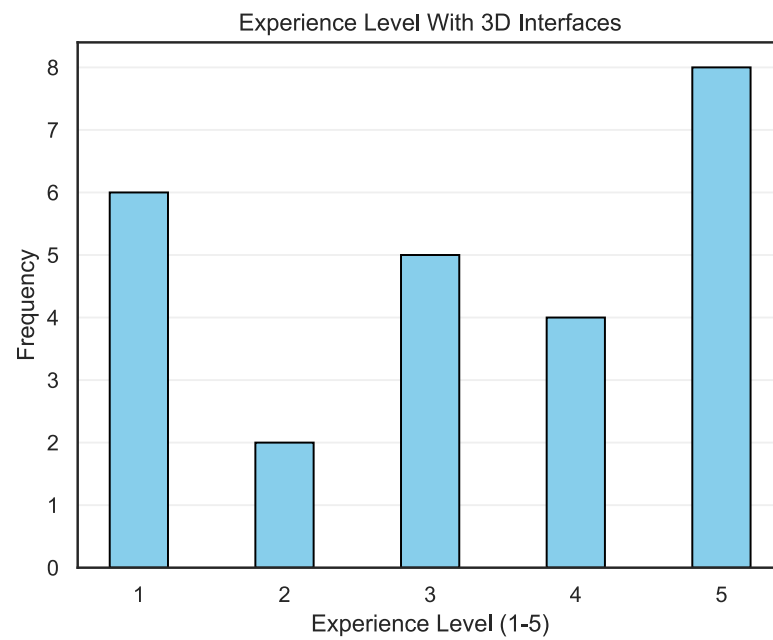


Figure 7.2: Level of experience using software to manipulate or visualize 3D models.

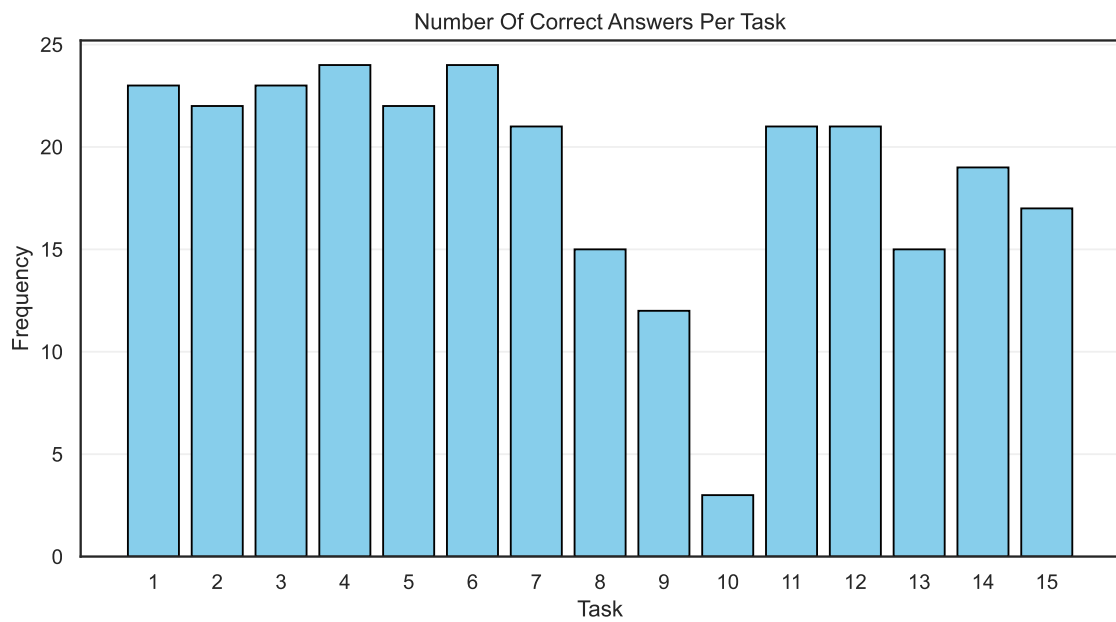


Figure 7.3: Number of correct answers for each task.

Task 2 22 correct out of 25.

Task 3 23 correct out of 25.

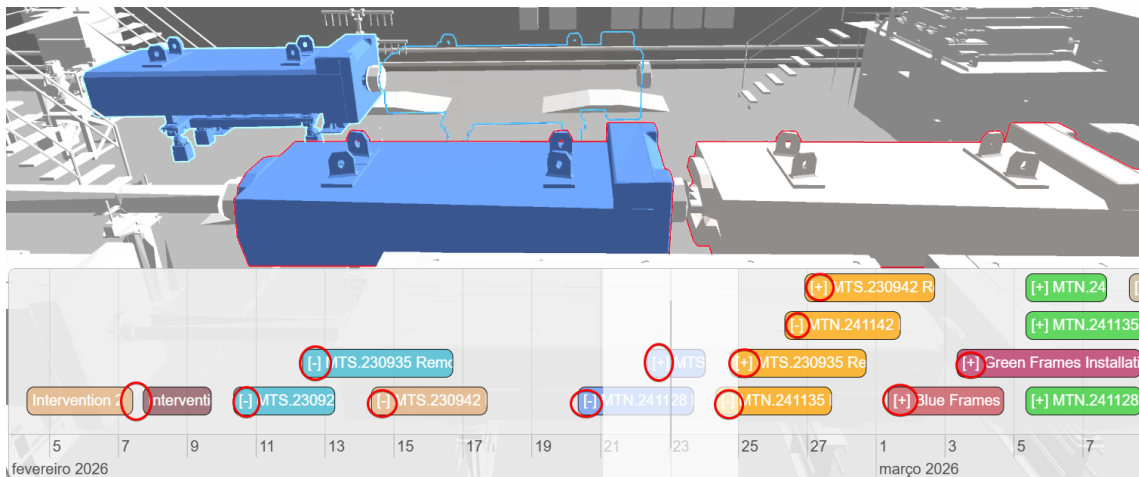


Figure 7.4: Tasks 1 to 3 — Highlight of task type markers on the timeline tasks.

These first questions were supposed to be easier if participants realized that the tasks were marked with a symbol indicating their type ("installation", "removal" or "maintenance") on the timeline, as highlighted in red in Fig. 7.4. Only 3 participants realized that. Most relied on moving the timeline and verifying if the components were changing or if outlines were present to distinguish between task types, which made people take much more time than expected.

Task 4 24 correct out of 25: 1 person mistook an ongoing maintenance task for an ongoing removal. Fig. 7.5 shows an example of what the testers saw during this task.

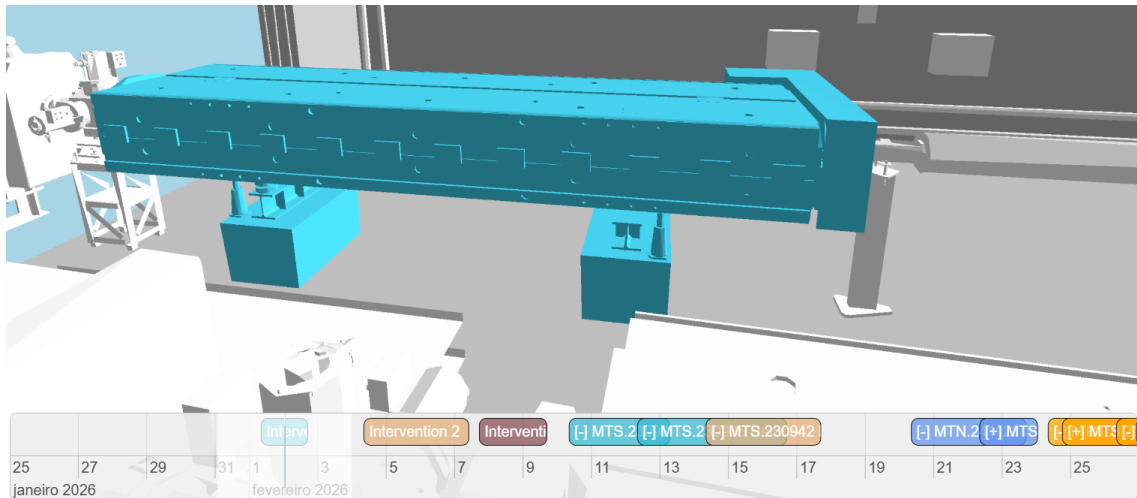


Figure 7.5: Task 4 — Ongoing maintenance task.

Task 5 22 correct out of 25: Few people incorrectly indicated that at least one of the tasks was ongoing. Like in Fig. 7.6, participants were allowed to inspect the task details to see the exact task intervals.

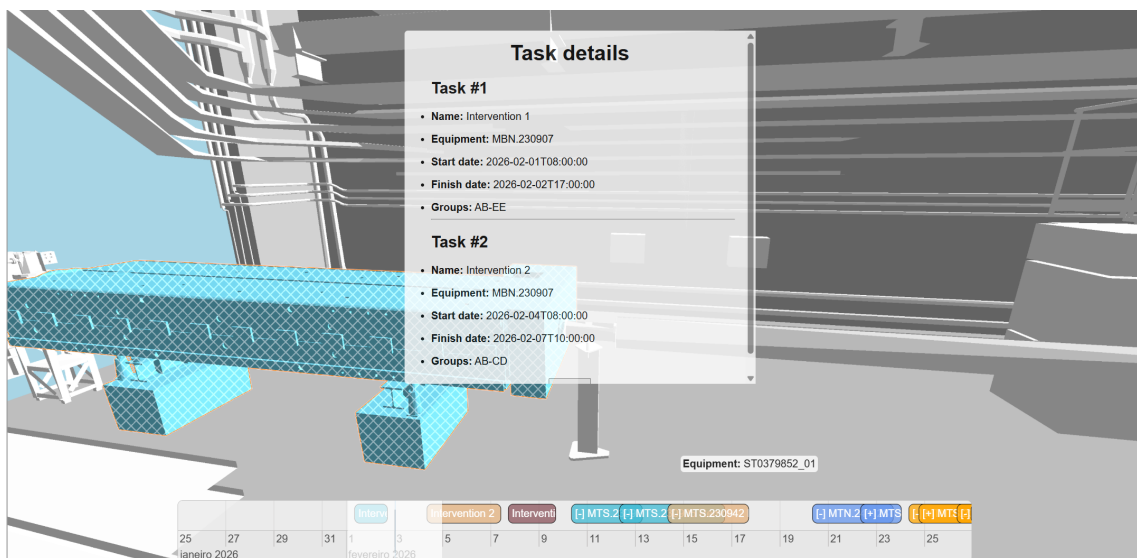


Figure 7.6: Task 5 — After maintenance and before another maintenance activity.

Task 6 24 correct out of 25: 1 person considered that the task had still not started. From task 6 onward (except task 14) the timeline items were hidden, just as in Fig. 7.7.

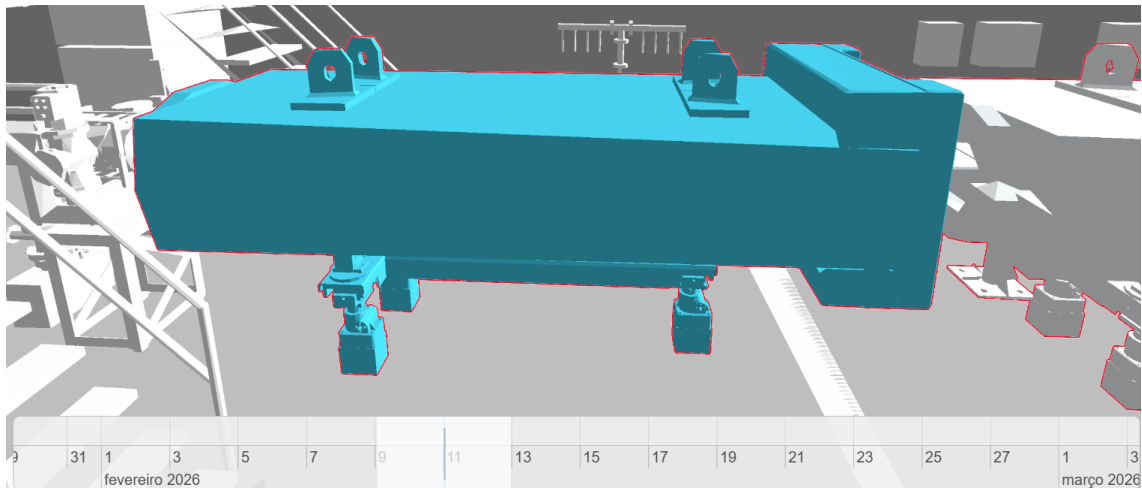


Figure 7.7: Task 6 — Ongoing removal task.

Task 7 21 correct out of 25: Some people had thought that the task on the equipment shown in Fig. 7.8 was ongoing or had already finished.

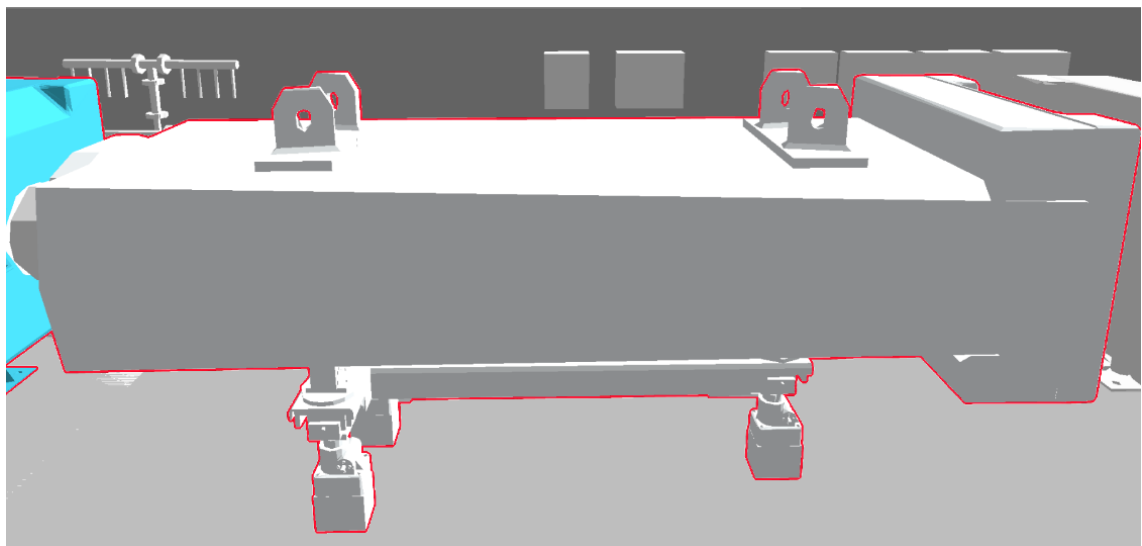


Figure 7.8: Task 7 — Before removal task.

Task 8 15 correct out of 25: All of the errors came from considering the task present in Fig. 7.9 ongoing and not finished. Without context, it can be an intuitive interpretation to consider a disappearing object as an ongoing removal of that object.

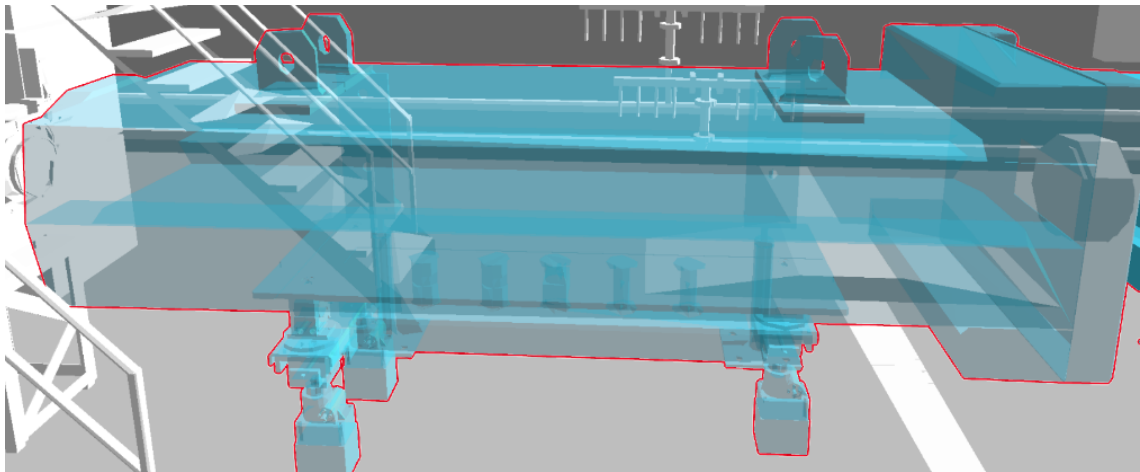


Figure 7.9: Task 8 — After removal task.

Task 9 12 correct out of 25: The removal present in Fig. 7.10 is not as unambiguous as the one in Fig. 7.8 since the color of the equipment is not fully white, but closer to the actual color of the task. Many participants did not move the window further to try to detect the remaining change in color and some other moved the window but did not notice it nevertheless, identifying the task as ongoing.

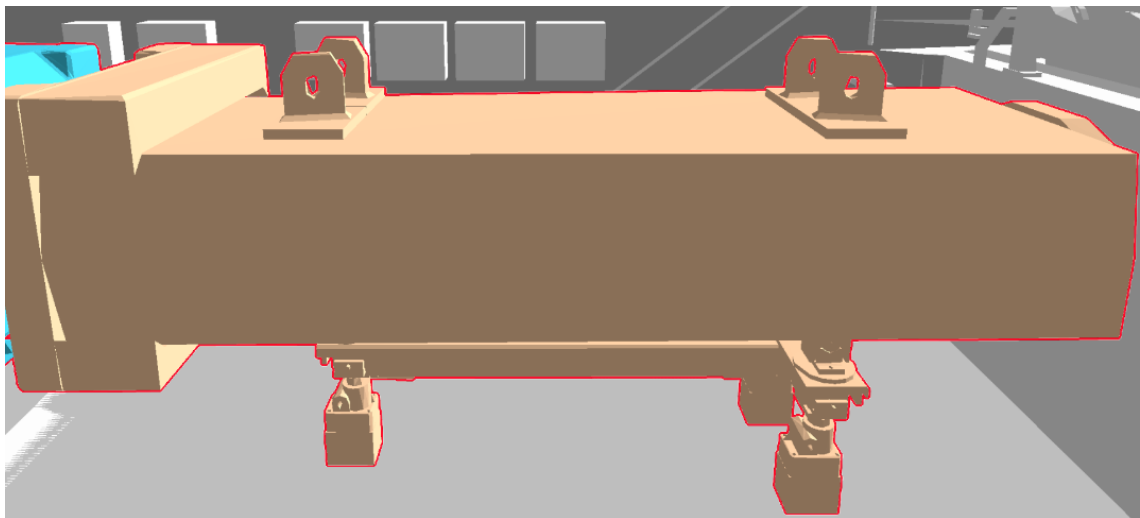


Figure 7.10: Task 9 — Before removal task.

Task 10 3 correct out of 25: Similarly, a few people did not realize the equipment in Fig. 7.11 was still transparent; however, the majority of the confusion stemmed from people considering that seeing a component progressively appear meant that the installation task was already occurring. Once again, this would be an intuitive implementation. Adding to this, 5 participants said that the task had already finished, maybe by comparing to the scenarios with removal of equipment. This was the question with the least amount of correct results.

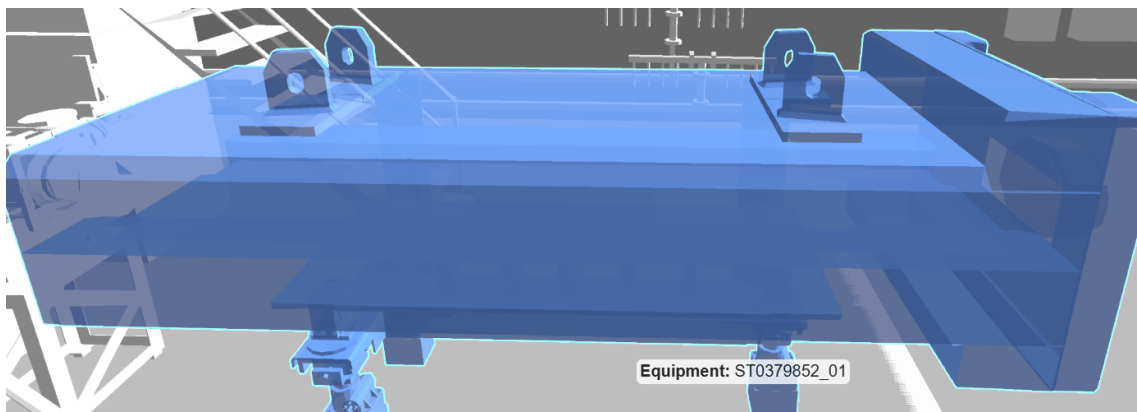


Figure 7.11: Task 10 — Before installation task.

Task 11 21 correct out of 25: Fig. 7.12 displays, in respective order, the equipment associated with tasks 11, 12 and 13. In this task, 4 people considered that the task had already finished.

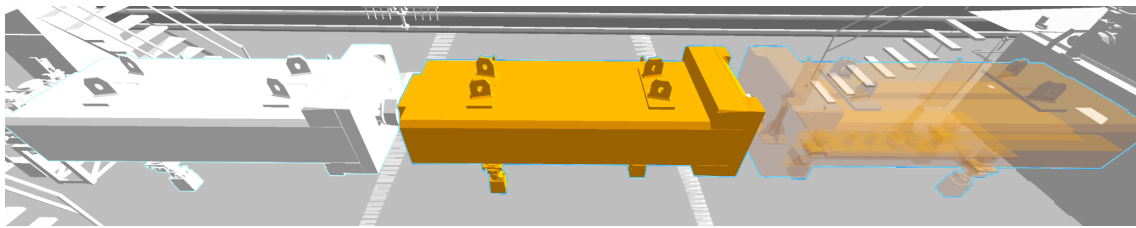


Figure 7.12: Tasks 11 to 13 — In order from left to right, after, ongoing and before installation tasks.

Task 12 21 correct out of 25: 3 people mistook the progress of the task as finished and one person indicated it was a maintenance activity.

Task 13 15 correct out of 25: Again the representation of an installation task before its start, similar to task 10. The results this time were better, likely because 3 equipments were shown at the same time and the participants compared them to each other, but it was still far from a very positive result.

Task 14 19 correct out of 25: Similarly to task 5, testers were allowed again to see the items on the timeline and to inspect task details. Fig. 7.13 shows the task details of both interventions with start and end dates. Some people considered that at least one of the tasks was ongoing.

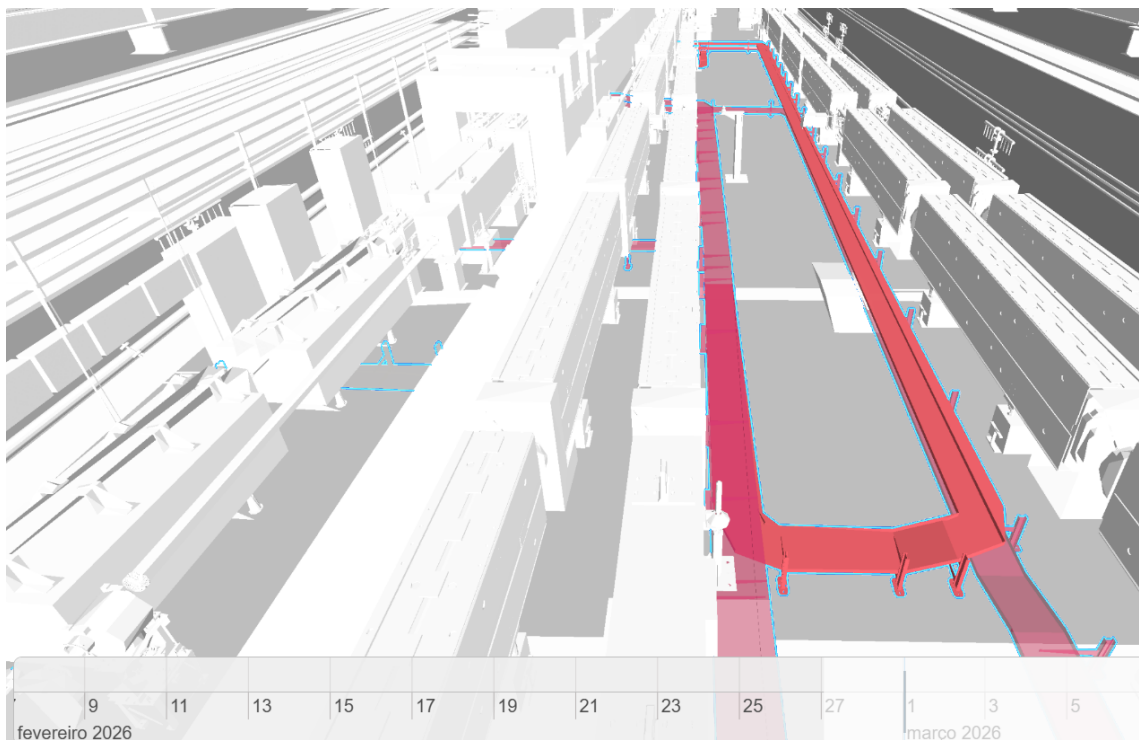


Figure 7.14: Task 15 — 3 installation tasks starting in the selected period.

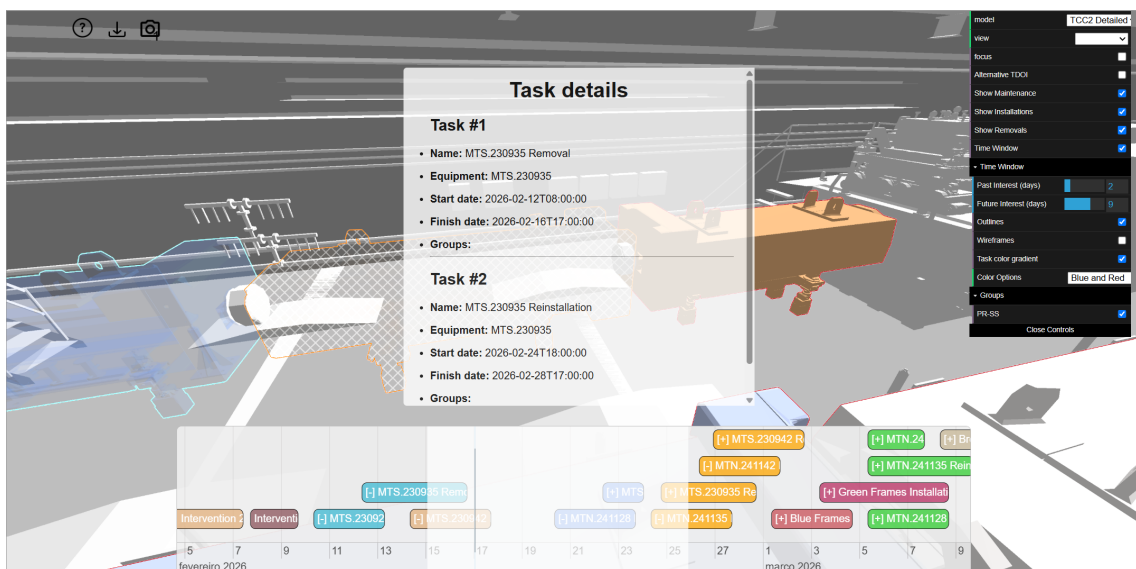


Figure 7.13: Task 14 — After removal and before reinstallation.

Task 15 17 correct out of 25: From the 8 incorrect answers, 3 testers were not able to distinguish the 2 different equipment that used similar colors, and the other 5 considered there were more than 3 different installations, likely because the selected components spanned a large area and had parts obstructed by other components. The initial camera is the one used in Fig. 7.14; however, the participants were free to move the camera anywhere.

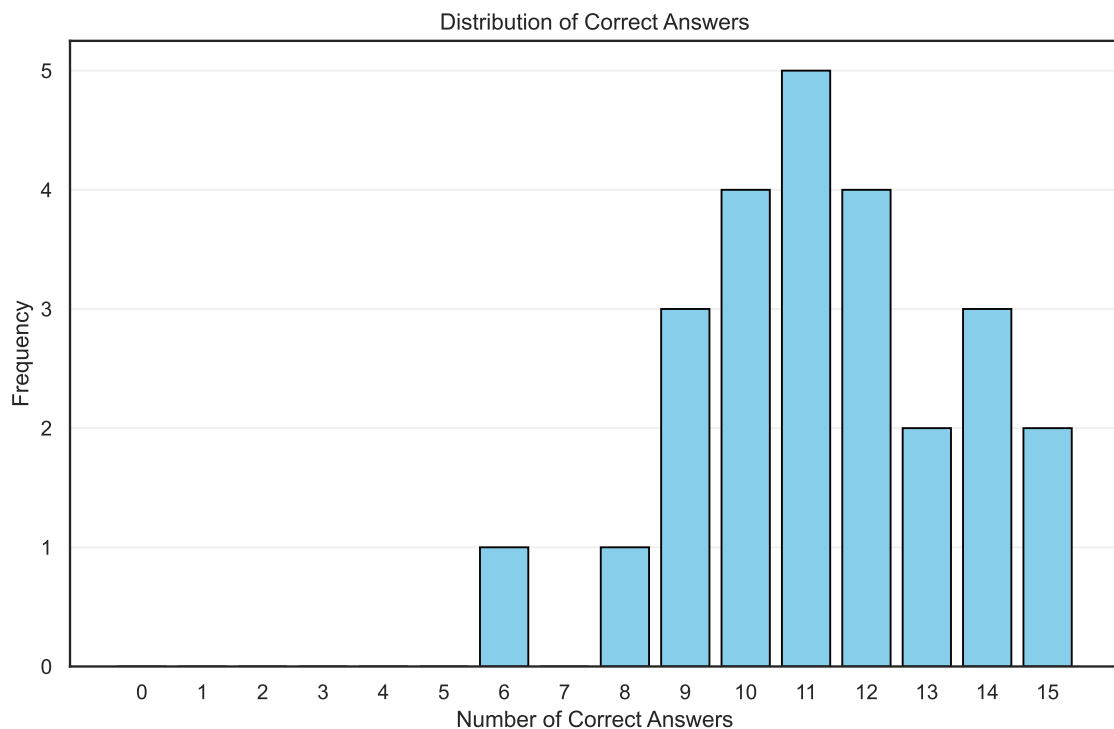


Figure 7.15: Distribution of the number of correct answers in a total of 15 tasks.

If the visual encoding is not to be changed, it seems best to let users know early, via a help menu, for example, the significance of all color and transparency effects. That is expected to greatly reduced the most common mistakes that were verified. Regarding the identification of the type of the tasks, errors are almost inexistent thanks to to the usage of the colored outlines.

7.2.2 Distribution of Correct Answers

The distribution of total correct answers across participants is shown in Fig. 7.15. The mode is 11 correct answers and 80% of the participants answered at least two thirds of the questionnaire correctly. Albeit the participant with the lowest score displayed some difficulty communicating in English, it is not clear if that was the main reason that lead to a worse result, since the participant seemed to have understood well what was being asked. Some details may not have been clear for the participant; however, being mainly a visual test, it may just be a coincidence.

7.2.3 User Preferences

The participants were inquired about 4 of their preferences: 83% prefer the outlines turned on and 66% prefer the wireframes turned on. At the start of the test, 12 participants started with the wireframes turned on and with the outline colors as green and magenta. The other 13 started with the wireframes off and the outlines colors as blue and red. Of the first group, 66% prefer the wireframes on and, of the second group 69% prefer them on too. Regarding the colors of the

outlines, 50% prefer green and magenta, 8% blue and red and the remaining have no preference. From the second group, 62% prefer blue and red and the remaining 38% have no preference. It can then be concluded that the participants tended to strongly prefer the initial colors they started the test with. The TDOI mechanism where the position of the time marker relative to the task interval is considered (which was the default one for all testers) was preferred 76% of the time, the other version was preferred by 4% of the participants (just 1) and 20% considered both options were valuable to have.

7.3 SUS Questionnaire Results

The SUS scores by participant are present in Fig. 7.16. Only 1 participant scored the system below 60 at 42.5, and 1 participant rated the system with 100 out 100 points. The scores exhibit considerable variation, suggesting diverse user perceptions of the system's usability. While it is believed that participants provided honest feedback, the variability may reflect differences in individual expectations and predispositions. Some users appear more easily impressed, readily assigning higher scores, whereas others are more discerning or reluctant to award the maximum score of 5 (even if they were able to use the tool rather well), contributing to the observed spread in ratings.

The questionnaire yielded a mean score of 79.4. According to common interpretations of SUS scores, this result is considered good but falls short of excellent. Bangor et al. (2008) suggests that SUS scores above 70 indicate acceptable usability, with scores around 80 classified as "good." However, scores closer to or above 85 are typically deemed "excellent." Similarly, Lewis and Sauro (2009) notes that a score of 79 places a system in the upper range of usability but below the threshold for exceptional user experience. These studies highlight that while a score of 79 reflects a positive usability experience, there is room for improvement to reach an outstanding level. This corresponds greatly to the feedback received. Most people found the system already adequate but still missing some options and changes that would make it clearer and easier to use.

7.4 Correlation With Demographics

This section presents the correlation of average correct responses from the 15-question experiment, grouped by age, experience level, and gender. The data is visualized using box plots, displaying the mean, minimum, and maximum values for each category, with sample sizes indicated as n .

The average correct answers across the four age brackets are present in Fig. 7.17. The highest average performance is observed in the ≤ 34 age group (12.3), while the 35–44 group shows the lowest average (9.7). A one-way analysis of variance (ANOVA) was conducted to evaluate differences in mean scores across the four age brackets: ≤ 34 , 35–44, 45–54, and ≥ 55 . The null hypothesis (H_0) posited that there are no differences in mean scores across these age groups, while the alternative hypothesis (H_1) suggested that at least one age group's mean score differs. The level of significance, α , is set to 0.05. The results obtained are an F-statistic of 1.7034 with a

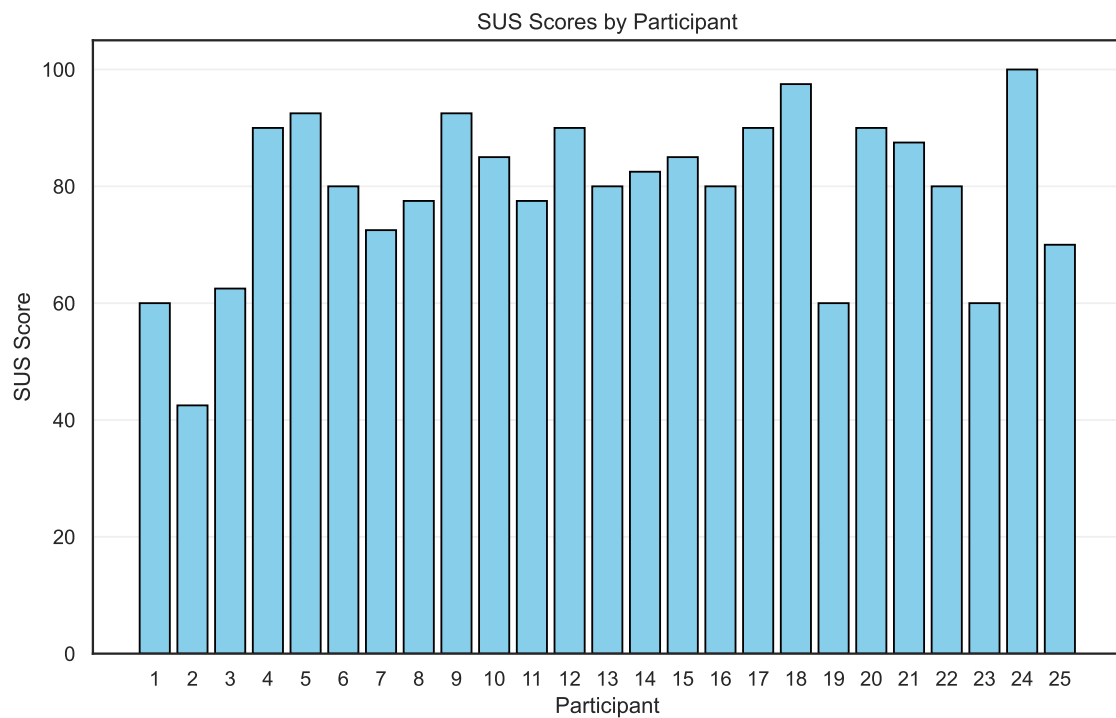


Figure 7.16: Distribution of SUS Scores

p-value of 0.1969. Since the p-value (0.1969) exceeds the significance threshold of 0.05, the null hypothesis is not rejected. This indicates that there are no statistically significant differences in the mean scores across the age brackets. Thus, participant age does not appear to significantly influence performance on the experiment.

The average correct answers across five experience levels (1 to 5) are visible in Fig. 7.18. The highest average is seen at experience level 3 (12.4), while level 4 shows the lowest (10.0). A separate one-way ANOVA was performed to assess differences in mean scores across the five experience levels. The null hypothesis (H_0) stated that mean scores do not differ across experience levels, while the alternative hypothesis (H_1) proposed that at least one experience level's mean score differs. The level of significance, α , is again set to 0.05. The results are an F-statistic of 0.9774 with a p-value of 0.4420. With a p-value of 0.4420, which is greater than 0.05, the null hypothesis is not rejected. This suggests that there are no statistically significant differences in mean scores across the experience levels. Participant experience does not appear to have a significant impact on performance in the experiment.

The average correct answers by gender are depicted in Fig. 7.19. Females exhibit a slightly higher average (12.0) compared to males (11.1), but the small sample sizes and overlapping ranges suggest no clear distinction.

The lack of significant differences in both statistical analyses implies that neither age nor experience level substantially affects performance on the 15-question experiment. This may suggest that the experiment's design is equally accessible across age groups and experience levels, or that the sample size ($n = 25$) may lack sufficient power to detect subtle differences. Future research

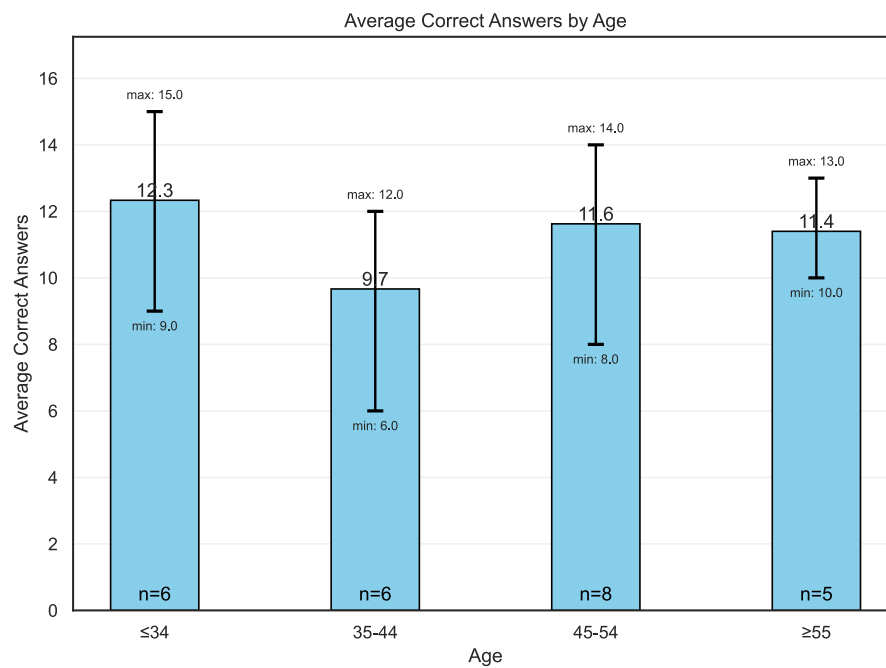


Figure 7.17: Average number of correct answers by age bracket.

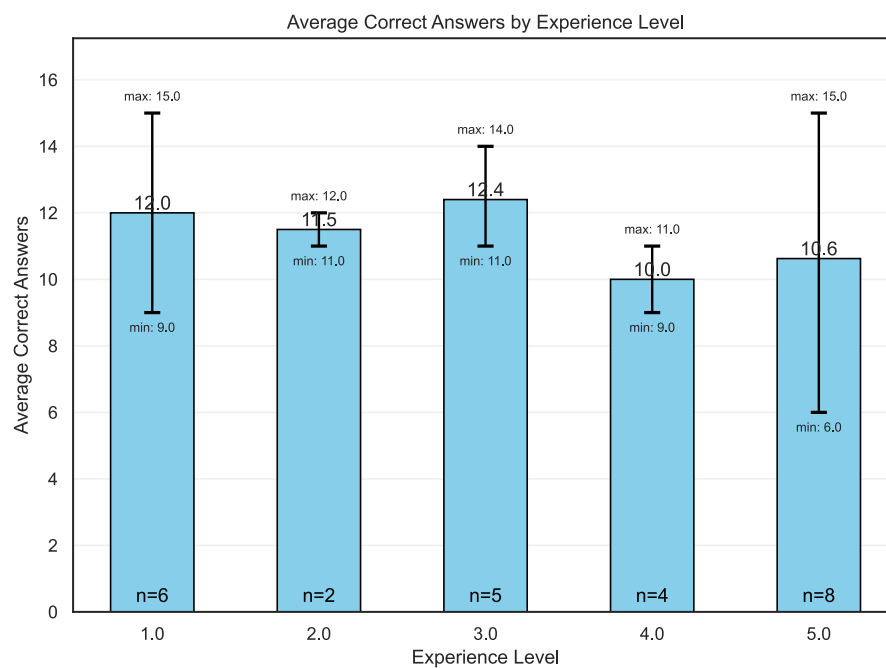


Figure 7.18: Average number of correct answers by experience level with 3D interfaces.

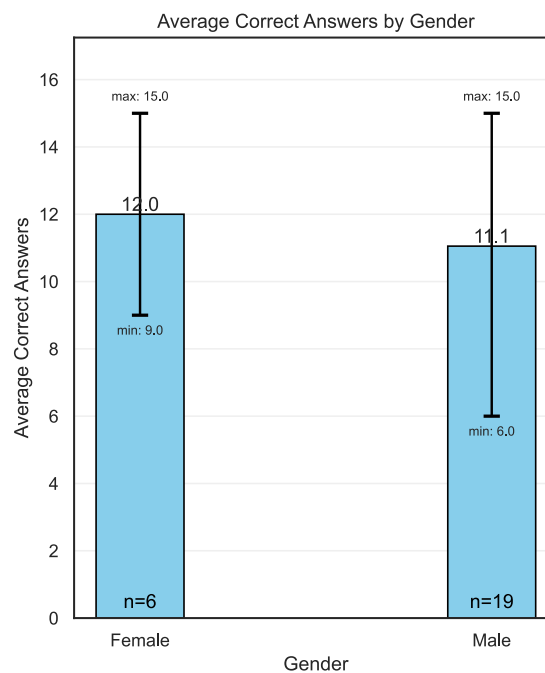


Figure 7.19: Average number of correct answers by gender.

could include a larger sample or additional variables (e.g., education) to refine these insights.

7.5 Feedback

All kinds of feedback were received: what were the worst parts of the experience, utility of the TDOI mechanism, user interface improvements and suggestions of new features. The feedback from participants regarding the worst aspects of the test highlights several key areas of concern: initial understanding of the tool, controls, information availability, and performance. Some users reported confusion, suggesting that the interface's onboarding or initial guidance was insufficient, indicating a need for improved tutorials or clearer introductory instructions to help users grasp the tool's functionality. A few users would have preferred different control mechanisms, similar to programs they use more frequently as CATIA. The lack of a legend and the option to filter tasks by group were recurring issues, and one user referenced that some elements were visually inconsistent with regards to color and scale of the UI elements, so enhancing information displays, including a legend could address this. Feedback such as "slow performance" or "clunkiness" occurred a few times, which highlights the importance of making interface well optimized so it can maintain responsiveness and smoothness for as many devices as possible.

The feedback regarding the TDOI feature reveals a range of opinions on its usability and preferred settings. Users generally find the feature useful, with several stating that it is "very useful" indicating its value. However, preferences for its default activation vary, reflecting differing usability needs. Several users suggest that TDOI should be an optional feature rather than the default. This suggests that while the feature is appreciated, forcing it on users at the start may

overwhelm or confuse some, particularly those unfamiliar with its purpose. Conversely, a few users advocate for it to be on by default, with one noting, "Should be the default option. It is useful to see many days all at once," highlighting its benefit for visualizing extended time periods.

The remaining feedback on the 3D interface, encompasses a mix of suggestions and general impressions, reveals diverse perspectives on usability, functionality, and applicability. Users provided valuable input across several areas, including interface clarity, navigation, visual aids, and integration features. Many users emphasized the need for improved visual clarity and information support. The lack of a clear reference for absolute or relative position within the model was noted, proposing a positional indicator at the top or bottom to aid orientation, especially given similar-looking equipment. Additional visual enhancements suggested include dashed and thicker outlines, transparent volumes to delimit blocked task areas (with adjustable sizes over time), painted paths on the floor, and bubbles or highlights to indicate expandable options like the timeline or task details. Maintenance activities often have prerequisites such as installing scaffolds and stairs, blocking the water supply and some electrical lines, and even other tasks or task steps like cabling. It was then proposed to add these prerequisites as a checklist or explicitly with their progress to the task details menu. Ideally, these activities should be visually represented too. Others requested the display of temperature and radioactivity levels that changes over time. This would require the creation of new APIs that served this information, but it underscores the enthusiasm and innovative ideas some users have for the tool. These proposed features could enhance its appeal, attract more users, and make it more compelling for existing ones.

Navigation and control suggestions focused on flexibility and intuitiveness. A few users would prefer controls similar to CATIA, however, the controls were well accepted in general. Most people did not use the keyboard to move. It was suggested that the timeline range should adjust when the TDOI window reaches the edges, and resizing both the TDOI window and the timeline vertical size directly was desired. A tutorial in slides or video format was proposed to improve initial understanding, alongside a loading indicator to manage user expectations during startup.

Functionality enhancements included options to remove the roof and walls, recenter to a task, filter by group, highlight tasks, and expand the timeline with buttons or borders. Users also suggested integrating important dates on the timeline or calendar, linking to external resources like a layout database or Gantt charts, and adding equipment names and category visibility on the timeline. Comments on task details included requests for links to tools used at CERN such as "Impact", "PLM" or the "Layout Database", summaries of multiple tasks, and activity-type or progress-based color coding instead of group-based colors, with balloons for task comments if available. Being able to have and possibly export an animation of the window sliding automatically was noted too.

Performance and applicability feedback varied. Some praised the controls and found the tool engaging for project leaders and office staff, but less practical for technicians or designers due to limited communication benefits or machine relevance. Concerns about slow loading and inconsistent GUI style (buttons and scale) were raised, alongside integration ideas like laser scanning and cross-page links with period indicators. A few noted issues with people not visiting the facilities in-person, suggesting the tool might exacerbate this, while others appreciated features like the

camera dropdown and proposed animations for installation processes (e.g., crane placement).

7.6 Summary

The evaluation of the temporal visualization tool involves 25 CERN participants, with ages ranging from 23 to 63 years (19 male, 6 female), providing a diverse demographic profile. Most participants lack prior experience with the tool, and experience with 3D model manipulation is average.

Task performance across 15 questions reveals varied success in understanding task statuses and model evolution, with correct response rates ranging from 3 to 24 out of 25. Tasks 1–3, designed to identify task types (installation, removal, maintenance) via timeline markers, achieve high accuracy (22–23 correct). Tasks 4–7 maintain strong performance (21–24 correct), but tasks 8–13, particularly those involving installation tasks before their start (e.g., Task 10 with only 3 correct), show significant errors due to misinterpretations of transparency and color cues, indicating challenges in perceiving pre-task states. Later tasks, such as Task 14 (19 correct) and Task 15 (17 correct), highlight additional difficulties. Task 14 benefits from visible timeline items and task details but still sees errors in identifying ongoing versus completed tasks. Task 15 suffers from confusion over equipment colors and obstructed views, with some participants overestimating the number of installations. These results suggest that while the tool effectively supports basic task identification, its representation of temporal progression, particularly for installation and removal tasks, requires clearer visual elements to enhance user comprehension.

Overall, the findings underscore the tool's potential for visualizing 3D model evolution while identifying areas for improvement in its intuitiveness. The varied task performance reflects challenges in interpreting visual indicators like transparency and color, especially for novice users or complex scenarios. These insights, grounded in the demographic and task-based data, inform refinements to the tool's design, aiming to enhance usability and clarity. The feedback and suggestions were varied and while it is not possible to implement most of them during this dissertation, the tool can evolve in several aspects.

Chapter 8

Conclusions

This dissertation set out to enhance CERN’s web-based 3D visualization system by developing an innovative approach to represent the temporal evolution of complex machines and their components. Through the implementation and evaluation of a prototype, this work addressed the challenges of visualizing large-scale models, their modifications over time, and associated maintenance tasks. This chapter summarizes the key findings, evaluates the achievement of the objectives, and discusses the implications and potential future directions for this research.

8.1 Final Remarks

The primary objective of this dissertation was to propose and validate a novel approach for visualizing the temporal evolution of complex machines within CERN’s web-based 3D visualization system. This goal was successfully achieved through the development of a usable prototype that adhered to the requirements. The prototype, utilizing a four-dimensional KD-Tree combined with a Temporal Degree of Interest (TDOI), enabled accurate representation of machine configurations at specific points in time, facilitating the visualization of maintenance tasks and structural changes.

Regarding the three research questions outlined in Section 1.4, the following can be concluded:

- **RQ1: How can 3D models representing parts of a facility at a specific time be identified and indexed to ensure accurate and efficient reconstruction of the complete machine from its components?** The use of a four-dimensional KD-Tree, inspired by Hertel et al. (2009), provided an effective method for indexing and reconstructing 3D models. This approach allowed for precise identification of model components at specific temporal states, ensuring assembly and accurate representation of the machine’s configuration with an acceptable efficiency, i.e., when selecting a different date on the timeline, the model updates quickly without noticeable delay; however, performing this operation many times per second reduces the performance.
- **RQ2: What are the most effective techniques for retrieving and assembling large 3D models with varying configurations and levels of detail from a model storage system**

to construct a scene, ensuring low latency? This question ended up not being explored explicitly. For the model tested, it is possible to change the visualization time and update the visibility of the components approximately 10 times per second at 10 frames per second. However, the full models are loaded initially and the model that exists at a given instant is "retrieved" by switching on and off the visibility of 3D objects. Not only that, but the LODs are not managed by a model storage system, but by the rendering system instead.

- **RQ3: How can the visualization of partial 3D models, which may overlap and evolve over time, be perceptible and convey information about ongoing maintenance tasks?**

The integration of the TDOI and the associated visual effects ensured that evolving model components and their associated maintenance tasks were clearly visualized. The prototype successfully conveyed temporal changes, such as component removals and installations, in a manner that was perceptible and informative for users. There can be simultaneous tasks in different but adjacent components and it is still possible to distinguish the activities completely. When the TDOI window intersects two different tasks on the same equipment (or if there are two different tasks simultaneously on the same equipment, although it may not be common), the orange outline mentioned in Section 5.4 serves only as an alert, meaning that no unambiguous visual system was created to depict this specific type of situation.

Usability tests conducted with CERN workers yielded positive results, confirming that the tool effectively is in a usable state and supports the needs of CERN workers in some capacity. The prototype's interface was found to be perceptible and partially intuitive, since some visual effects were not well understood before they were explained. However, they were very well accepted after being explained by almost all participants. Feedback from these tests highlighted the tool's ability to consolidate relevant information, reducing the need to cross-reference multiple sources. However, some areas for improvement were identified and in the experiment protocol as well. For instance, the number of questions in the usability tests could have been reduced to shorten the duration of the test (which was frequently longer than 30 minutes), and additional pilot tests would have been beneficial to notice and address early issues frequently noted by the participants, such as the lack of a legend.

Feedback collected during testing has led to several implemented features: a legend explaining group colors and outline meanings (top left of Fig. 8.1), task filtering by group, and the ability to select tasks on the timeline for a summary (Fig. 8.2). Timeline tasks now feature colored borders based on activity type (Fig. 8.2). ALT-clicking a task expands its details, while ALT+CTRL-clicking moves the camera near the associated equipment. Users can now hide components like roofs and walls, exemplified in Fig. 8.1 where the wall is not visible (it requires manual specification in a configuration file). Task details include a link to Impact when available and explicitly state the task type. The help menu has been overhauled, now offering detailed information on new features, organized into tabs and sections with images and animations (Fig. 8.3). Some performance optimizations mentioned in Section 5 (inverse hull outlines and LODs) were implemented post-testing (a computer with an AMD Ryzen 5 4600H CPU and NVIDIA GeForce GTX 1650 Ti

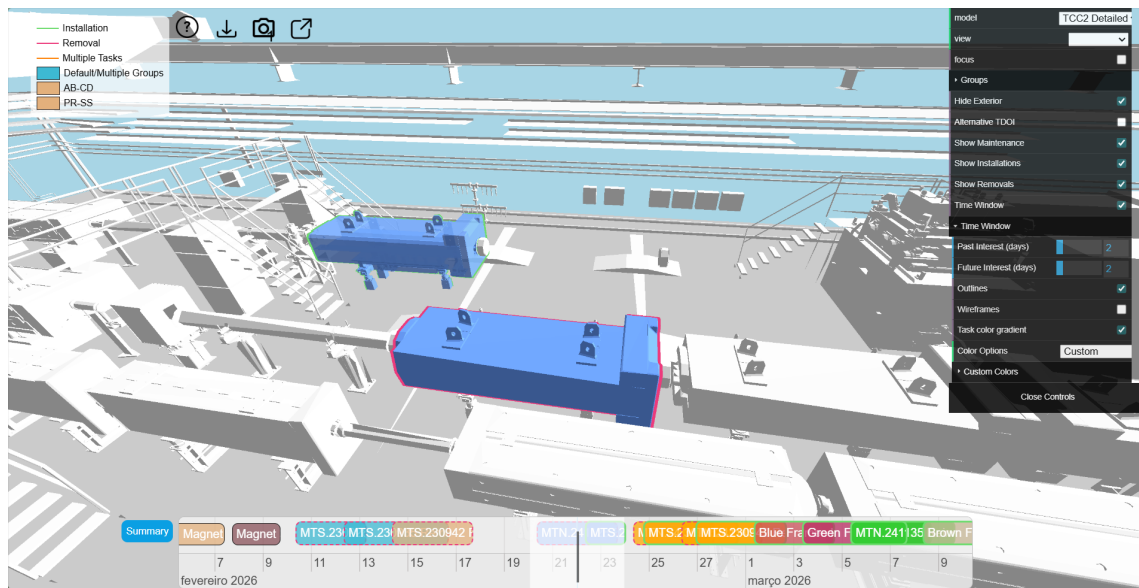


Figure 8.1: Visualization tool with improvements: ability to hide walls, legend.

Mobile now delivers an average performance of 60 FPS when facing a zone with lots of equipment and an average of 100 FPS when closer to areas with fewer objects). Additionally, a shortcut now enables automatic window movement across the visible timeline range. The remaining feedback, such as suggestions for additional features and further performance optimizations, has been documented for subsequent development.

8.2 Future Work

The feedback and insights gained from this study provide a foundation for further enhancements to the visualization system. Several areas for future development were identified, including:

- Standardize GUI elements (e.g., buttons, colors used, scale) to address inconsistencies.
- Integrate data from and link to other tools used at CERN, so that more details regarding the activities and the equipment are available.
- Add transparent volumes to represent the space that is required and blocked by the activities. Complementarily, identify clear paths to move equipment and personnel.
- Develop displays for temperature and radioactivity levels that change over time (requires new APIs to serve this data).
- Create animations and status identifiers for installation processes (e.g., crane and scaffolds placement) to visually demonstrate task steps.
- Add checklists or progress tracking for maintenance activity prerequisites (e.g., scaffolds, stairs, blockage of the water supply and electrical lines) in the task details menu.

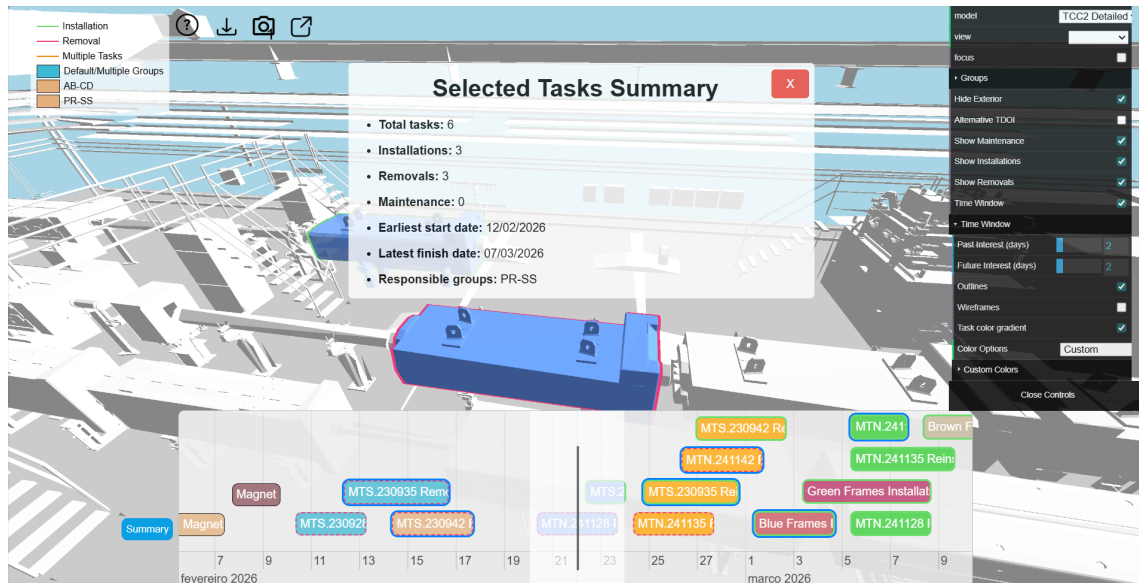


Figure 8.2: Summary of the selected tasks on the timeline and timeline items outlines: tasks outlined in blue are selected, task outlined in green are installations and tasks outlined in magenta are removals.

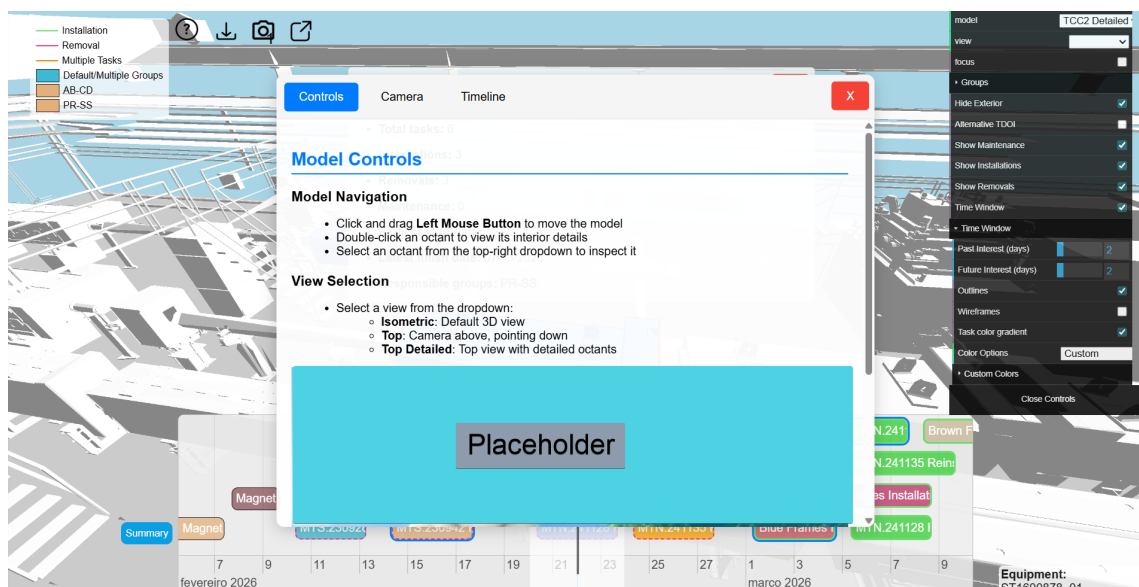


Figure 8.3: Help menu with tabs, sections and media (placeholder).

- Include visual representations of maintenance activity prerequisites to improve task context.
- Explore different indicators of temporal proximity other than transparency, such as symbols, gradients, textures, etc.
- Optimize loading times to improve performance (some steps may be parallelized).
- Explore the incompatibility issues that make the tool slower in Firefox and unable to execute on Safari.
- Improve the KD-Tree data structure or replace it with a more performant structure to enhance overall performance (maybe Interval trees or temporal Octrees).
- Explore dynamic LOD switching to automatically transition between global and detailed views based on proximity or zoom level, enabling seamless navigation between buildings or facilities.
- Add support for the comparison of more than two versions of a facility at a time, so that the timeline is not restricted to a fixed number of months.
- Explore the GLTFPack and meshoptimizer libraries (or alternatives) to develop a simplification program that is able to use different simplification targets for different objects of the same model, based on properties the size of the meshes.

These ideas represent opportunities to build upon the current implementation, making it more robust, scalable, and adaptable to the needs of CERN's teams. By addressing these areas, the visualization system can continue to evolve, supporting more efficient planning and execution of maintenance tasks and contributing to the overall productivity of CERN's scientific operations. In parallel, they can inspire new systems used in different contexts that further explore the visualization of temporal models.

In conclusion, this dissertation has successfully demonstrated the feasibility of a web-based 3D visualization system that captures the temporal evolution of complex machines. The combination of a four-dimensional KD-Tree and TDOI has proven effective in addressing the challenges of large-scale model visualization and maintenance task representation. The positive feedback from usability tests underscores the potential of this approach to enhance decision-making and operational efficiency at CERN. The proposed future work provides several options to improve this system and ensure its continued relevance and impact in the context of large-scale scientific facilities.

Bibliography

- Aigner, W., Miksch, S., Schumann, H., and Tominski, C. (2023). *Visualization of time-oriented data*. Human-Computer Interaction Series. Springer, London. DOI: 10.1007/978-1-4471-7527-8. ISBN: 978-1-4471-7527-8.
- Akenine-Möller, T., Haines, E., and Hoffman, N. (2018). *Real-time rendering*. CRC Press, 4th edition. ISBN: 9781138627000.
- Andrienko, N., Andrienko, G., and Gatalsky, P. (2003). Exploratory spatio-temporal visualization: An analytical review. *Journal of Visual Languages and Computing*, 14(6):503–541. DOI: 10.1016/S1045-926X(03)00046-6.
- Armstrong, M. P. (1988). Temporality in spatial databases. In *Proceedings of Accessing the World : Third Annual International Conference (GIS/LIS '88):, Exhibits, and Workshops. San Antonio, Texas. November 30-December 2, 1988*, pages 880–889. American Society for Photogrammetry and Remote Sensing.
- Bach, B., Dragicevic, P., Archambault, D., Hurter, C., and Carpendale, S. (2014). A review of temporal data visualizations based on space-time cube operations. In *Proceedings of the Eurographics Conference on Visualization (EuroVis), June 2014, Swansea, UK*, number 3, pages 43–62. DOI: 10.2312/eurovisstar.20141171.
- Bangor, Aaron, Kortum, P., T., P., Miller, and T., J. (2008). The system usability scale (SUS): an empirical evaluation. *International Journal of Human-Computer Interaction*, 24:574–. DOI: 10.1080/10447310802205776.
- Besaçon, L., Ynnerman, A., Keefe, D. F., Yu, L., and Isenberg, T. (2021). The state of the art of spatial interfaces for 3D visualization. *Computer Graphics Forum*, 40(1):293–326. DOI: 10.1111/cgf.14189.
- Borkowski, A. S. (2023). Evolution of bim: Epistemology, genesis and division into periods. *Journal of Information Technology in Construction*, 28:646–661. DOI: 10.36680/j.itcon.2023.034.
- Brazauskas, J., Verma, R., Safronov, V., Danish, M., Lewis, I., and Mortier, R. (2021). Real-time data visualization on the adaptive city platform. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys*

- '21), November 2021, Coimbra, Portugal, number 8 in BuildSys '21, pages 210–211, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3486611.3492225.
- Brooke, J. (1996). *Usability evaluation in industry*, chapter SUS-A quick and dirty usability scale, pages 189–194. CRC Press. DOI: 10.1201/9781498710411-35. ISBN: 9780748404605.
- Card, S. K. and Nation, D. (2002). Degree-of-interest trees: A component of an attention-reactive user interface. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '02), May 2002, Trento, Italy*, number 1 in AVI '02, pages 231–245, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/1556262.1556300. ISBN: 1-58113-537-8.
- Carvalho, A., de Sousa, A. A., Ribeiro, C., and Costa, E. (2008). A temporal focus + context visualization model for handling valid-time spatial information. *Information Visualization*, 7:265–274. DOI: 10.1057/palgrave.ivs.9500188.
- CGPress (2010). The history of 3D studio. Available at https://cgpress.org/archives/cgarticles/the_history_of_3d_studio, last accessed on June 2025.
- Cheymol, A., Fouche, G., Gramoli, L., Hirao, Y., Hummel, E., Mavromatis, M., Moullec, Y., Argelaguet, F., and Nouviale, F. (2022). The rubber slider metaphor: Visualisation of temporal and geolocated data. In *Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), 2022, Christchurch, New Zealand, VRW 2022*, pages 904–905. IEEE. DOI: 10.1109/VRW55335.2022.00303.
- Deloitte (2018). Augmented reality: on the cusp of reality. Technical report, Deloitte Insights. Available at <https://www.deloitte.co.uk/tmtpredictions2018/assets/img/downloads/FY18-Predictions-Augmented-Reality-Single-Prediction.pdf>, last accessed on June 2025.
- Dou, S. Q., Zhang, H. H., Zhao, Y. Q., Wang, A. M., Xiong, Y. T., and Zuo, J. M. (2020). Research on construction of spatio-temporal data visualization platform for gis and bim fusion. In Zhou, G. Q., Zhang, X. P., Wang, H. Y., Wan, W. Y., Cui, Y. X., Liu, G. Q., and Luo, M. H., editors, *Proceedings of the 2020 International Conference on Geomatics in the Big Data Era (ICGBD 2020), November 15-19, 2019*, volume 42, pages 555–563. International Society for Photogrammetry and Remote Sensing. DOI: 10.5194/isprs-archives-XLII-3-W10-555-2020.
- Dursun, I., Varllk, A., and Ayyıldız, E. (2022). Integration and web-based presentation of 3D city models and BIM data: The case of the Koycegiz campus. In Zlatanov, S., Sithole, G., and Barton, J., editors, *Proceedings of the 24th ISPRS Congress on Imaging Today, Foreseeing Tomorrow (ISPRS), 6-11 June 2022, Nice, France*, volume 43, pages 513–519. International Society for Photogrammetry and Remote Sensing. DOI: 10.5194/isprs-archives-XLIII-B4-2022-513-2022.

- Eligehausen, J., Riecken, L., and Borchardt, D. (2013). Blendgis: An open source tool for 3D and 4D visualization of ecology related spatial data. In Kolditz, O., Rink, K., and Scheuermann, G., editors, *Proceedings of the 1st Workshop on Visualisation in Environmental Sciences (EnvirVis), June 2013*, pages 1–5, Leipzig, Germany. The Eurographics Association. DOI: 10.2312/PE.EnvirVis.EnvirVis13.001-005.
- Erwig, M. and Schneider, M. (2002). *Mining spatio-temporal information systems*, chapter STQL — A spatio-temporal query language, pages 105–126. Springer US, Boston, MA. DOI: 10.1007/978-1-4615-1149-6_6. ISBN: 978-1-4615-1149-6.
- Eswaran, M. and Bahubalendruni, A. (2022). Challenges and opportunities on AR/VR technologies for manufacturing systems in the context of industry 4.0: A state of the art review. *Journal of Manufacturing Systems*, 65:260–278. DOI: 10.1016/j.jmsy.2022.09.016.
- Farrimond, B. and Hetherington, R. (2004). Using 3D to visualize temporal data. In *Proceedings of the Eighth International Conference on Information Visualisation (IV), July 2004, London, UK*, number IV 2004, pages 108–117. DOI: 10.1109/IV.2004.1320132.
- Foley, J. D. and Hughes, J. F. (2014). *Computer graphics: principles and practice*. Addison-Wesley, Upper Saddle River, NJ, 3rd edition. ISBN: 9780321399526.
- Fouché, G., Argelaguet, F., Faure, E., and Kervrann, C. (2023). Immersive and interactive visualization of 3D spatio-temporal data using a space time hypercube: Application to cell division and morphogenesis analysis. *Frontiers in Bioinformatics*, 3. DOI: 10.3389/fbinf.2023.998991.
- Friendly, M. (2008). A Brief History of Data Visualization. In Chen, C.-h., Härdle, W., and Unwin, A., editors, *Handbook of Data Visualization*, pages 15–56. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-540-33037-0_2. ISBN: 978-3-540-33037-0.
- Froehlich, B. and Bowman, D. (2009). 3D user interfaces. *IEEE Computer Graphics and Applications*, 29(6):20–21. DOI: 10.1109/MCG.2009.113.
- Gautier, C., Delanoy, J., and Gesquiere, G. (2023). Representation of urban geometry evolution through space-time cube. In *Proceedings of the 27th International Conference Information Visualisation (IV), July 2023, Vienna, Austria*, pages 414–419. DOI: 10.1109/IV60283.2023.00079.
- Golparvar-Fard, M., Peña-Mora, F., Arboleda, C. A., and Lee, S. (2009). Visualization of construction progress monitoring with 4D simulation model overlaid on time-lapsed photographs. *Journal of Computing in Civil Engineering*, 23(6):391–404. DOI: 10.1061/(ASCE)0887-3801(2009)23:6(391).
- Guo, C., Chen, Y. V., Miller, C. L., Hartman, N. W., Mueller, A. B., and Connolly, P. E. (2014). Information visualization for product lifecycle management (PLM) data. In *Proceedings of the 121st American Society for Engineering Education Annual Conference and Exposition (ASEE)*,

- June 15-18, 2014, Indianapolis, Indiana, number 106581. American Society for Engineering Education.
- Heok, T. K. and Daman, D. (2004). A review on level of detail. In *Proceedings. International Conference on Computer Graphics, Imaging and Visualization. CGIV July 2004. Menang, Malaysia*, pages 70–75. DOI: 10.1109/CGIV.2004.1323963.
- Hertel, S., Wagner, M., and Westermann, R. (2009). A system for visualizing spatiotemporal urban datasets. In *Proceedings of the 17th International Conference on Geoinformatics (GEOINFORMATICS), August 12-14, 2009, Fairfax, VA, USA*, number 17. DOI: 10.1109/GEOINFORMATICS.2009.5293315.
- ISO (2002). ISO 19108:2002(e): Geographic information — temporal schema. Standard, International Organization for Standardization, Geneva, CH.
- Ivson, P., Moreira, A., Queiroz, F., Santos, W., and Celes, W. (2020). A systematic review of visualization in building information modeling. *IEEE Transactions on Visualization and Computer Graphics*, 26(10):3109–3127. DOI: 10.1109/TVCG.2019.2907583.
- Ivson, P., Nascimento, D., Celes, W., and Barbosa, S. D. J. (2018). Cascade: A novel 4D visualization system for virtual construction planning. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):687–697. DOI: 10.1109/TVCG.2017.2745105.
- Jaillot, V. (2020). *3D, temporal and documented cities: Formalization, visualization and navigation*. PhD thesis, Université de Lyon.
- Jang, E. (2000). 3D animation coding: its history and framework. In *Proceedings of the IEEE International Conference on Multimedia and Expo, (ICME), August 2000*, volume 2, pages 1119–1122 vol.2, New York, USA. DOI: 10.1109/ICME.2000.871557.
- Kim, K., Carlis, J. V., and Keefe, D. F. (2017). Comparison techniques utilized in spatial 3D and 4D data visualizations: A survey and future directions. *Computers and Graphics*, 67:138–147. DOI: 10.1016/j.cag.2017.05.005.
- Kim, S.-S., Lee, S.-H., Kim, K.-H., and Lee, J.-H. (2003). A unified visualization framework for spatial and temporal analysis in 4D GIS. In *Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS), 21-25 July 2003, Toulouse, France*, volume 6, pages 3715–3717. DOI: 10.1109/IGARSS.2003.1295236.
- Lewis, J. and Sauro, J. (2009). The factor structure of the system usability scale. In *Proceedings of the 1st International Conference on Human Centered Design: Held as Part of HCI International, (HCD '09), July 2009, California*, volume 5619, pages 94–103. DOI: 10.1007/978-3-642-02806-9_12; ISBN: 978-3-642-02805-2.

- Li, L., Qu, L., Ying, S., Liang, D., and Hu, Z. (2009). Use Google SketchUp to implement 3D spatio-temporal visualization. In *Proceedings of the International Symposium on Spatial Analysis, Spatial-Temporal Data Modeling, and Data Mining, October 2009, Wuhan, China*, volume 7492. DOI: 10.1117/12.837572.
- Mao, B., Yu, Z., and Cao, J. (2016). Large scale spatial temporal data visualization based on spark and 3D volume rendering. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN), July 24-29, 2016, Vancouver, Canada*, number 2161-4407, pages 1879–1882. DOI: 10.1109/IJCNN.2016.7727428.
- Masoodian, M., Yusof, A. B. M., and Rogers, B. (2015). Supporting focus and context awareness in 3D modelling tasks using multi-layered displays. *Computer Graphics Forum*, 34(6):1–12. DOI: 10.1111/cgf.12513.
- McKinney, K. and Fischer, M. (1998). Generating, evaluating and visualizing construction schedules with CAD tools. *Automation in Construction*, 7(6):433–447. DOI: 10.1016/S0926-5805(98)00053-3.
- Minard, C. J. (1869). Carte figurative des pertes successives en hommes de l’armée française dans la campagne de russie 1812–1813. Drawn by Charles Joseph Minard, Inspector General of Bridges and Roads in retirement. Often referred to as "Napoleon’s March" map. Reproduction and analysis available at <https://www.edwardtufte.com/product/napoleons-march-french-and-english-translation/>.
- Mota, R., Ferreira, N., Silva, J. D., Horga, M., Lage, M., Ceferino, L., Alim, U., Sharlin, E., and Miranda, F. (2023). A comparison of spatiotemporal visualizations for 3D urban analytics. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):1277–1287. DOI: 10.1109/TVCG.2022.3209474.
- Müller, R. and Zeckzer, D. (2015). Past, present, and future of 3D software visualization: A systematic literature analysis. In *Proceedings of the 6th International Conference on Information Visualization Theory and Applications (VISIGRAPP), March 2015, Berlin, Germany*, pages 63–74. INSTICC, SciTePress. DOI: 10.5220/0005325700630074.
- Neumann, U., You, S., Hu, J., Jiang, B., and Lee, J. (2003). Augmented virtual environments (AVE): Dynamic fusion of imagery and 3D models. In *Proceedings of the IEEE Virtual Reality (VR), March 22-26, 2003, Los Angeles, CA, USA*, pages 61–67. IEEE Computer Society. DOI: 10.1109/VR.2003.1191122. ISBN: 0769518826.
- Ni, Y. Q., Lin, K. C., Wu, L. J., and Wang, Y. W. (2017). Visualized spatiotemporal data management system for lifecycle health monitoring of large-scale structures. *Journal of Aerospace Engineering*, 30(2). DOI: 10.1061/(ASCE)AS.1943-5525.0000622.
- Nielsen, J. (1994). *Usability engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. DOI: 10.5555/2821575. ISBN: 9780080520292.

- Nienhaus, M., Kirsch, F., and Döllner, J. (2006). Illustrating design and spatial assembly of interactive CSG. In *Proceedings of the 4th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa (AFRIGRAPH), January 2006, Cape Town, South Africa*, number 4, pages 91–98, Cape Town, South Africa. ACM. DOI: 10.1145/1108590.1108605. ISBN: 978-1-59593-288-4.
- Nunes, R. A. (2023). A web-based system for visualizing engineering tasks on multi-lod large-scale cad models. Master's thesis, Faculty of Engineering, University of Porto, Porto, Portugal.
- Peng, Y., Dong, J., Chen, L., Chu, H., and Yong, J. (2011). An optimal color mapping strategy based on energy minimization for time-varying data. In *Proceedings of the 12th International Conference on Computer-Aided Design and Computer Graphics (CADGraphics), September 2011, Jinan, China*, number 12, pages 411–417. DOI: 10.1109/CAD/Graphics.2011.68.
- Pindat, C., Pietriga, E., Chapuis, O., and Puech, C. (2013). Drilling into complex 3D models with GimLenses. In *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology (VRST '13), October 2013, Singapore*, VRST '13, pages 223–230, New York. Association for Computing Machinery. DOI: 10.1145/2503713.2503714. ISBN: 978-1-4503-2379-6.
- Piringer, H., Kosara, R., and Hauser, H. (2004). Interactive focus+context visualization with linked 2D/3D scatterplots. In *Proceedings of the Second International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV), July 2004, London, UK*, number 2, pages 49–60. DOI: 10.1109/CMV.2004.1319526.
- Ponchio, F. and Hormann, K. (2008). Interactive rendering of dynamic geometry. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):914–925. DOI: 10.1109/TVCG.2008.35.
- Qing, X., Yang, Z., Baoming, Z., and ChaoZhen, L. (2011). Modeling and visualization of space objects spatio-temporal data. In Remondino, F. and El-Hakim, S., editors, *Proceedings of the 4th ISPRS International Workshop on 3D Virtual Reconstruction and Visualization of Complex Architectures (3DARCH 2011), March 2-4, 2011, Trento, Italy*, volume 38, pages 155–162. International Society for Photogrammetry and Remote Sensing.
- Renolen, A. (2000). Modelling the real world: Conceptual modelling in spatiotemporal information system design. *Transactions in GIS*, 4(1):23–42. DOI: <https://doi.org/10.1111/1467-9671.00036>.
- Rodríguez-Gonzálvez, P., Muñoz-Nieto, A. L., DelPozo, S., Sanchez-Aparicio, L. J., Gonzalez-Aguilera, D., Micoli, L., Barsanti, S. G., Guidi, G., Mills, J., Fieber, K., Haynes, I., and Hejmanowska, B. (2017). 4D reconstruction and visualization of cultural heritage: Analyzing our legacy through time. In Remondino, F., Kersten, T., Georgopoulos, A., Stathopoulou, E., and Aguilera, D., editors, *Proceedings of the 3D Virtual Reconstruction and Visualization of Complex Architectures, March 1-3, 2017, Nafplio, Greece*, volume 42, pages 609–616. International

- Society for Photogrammetry and Remote Sensing. DOI: 10.5194/isprs-archives-XLII-2-W3-609-2017.
- Samet, H. (2016). Sorting in space: Multidimensional data structures for computer graphics and vision applications. In *Proceedings of the SIGGRAPH ASIA 2016 (SIGGRAPH ASIA), December 2016, Macau*, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/2988458.2988503. ISBN: 978-1-4503-4538-5.
- Sando, T., Tory, M., and Irani, P. (2009). Effects of animation, user-controlled interactions, and multiple static views in understanding 3D structures. In *Proceedings of the Symposium on Applied Perception in Graphics and Visualization (APGV), 2009, Chania, Greece*, number 78543, pages 69–76. DOI: 10.1145/1620993.1621008. ISBN: 978-1-60558-743-1.
- Siabato, W., Claramunt, C., Ilarri, S., and Manso-Callejo, M. A. (2018). A survey of modelling trends in temporal gis. *ACM Computing Surveys*, 51(2). DOI: 10.1145/3141772.
- Simões, B., Marangon, M., and De Amicis, R. (2015). C-space: A mobile framework for the visualization of spatial-temporal 3D models. In Damiani, E., Howlett, R. J., Jain, L. C., Gallo, L., and De Pietro, G., editors, *Proceedings of the 8th KES International Conference on Intelligent Interactive Multimedia Systems and Services (IIMSS-2015), June 2015, Sorrento, Italy*, volume 40 of *Smart Innovation, Systems and Technologies*, pages 35–45. Springer. DOI: 10.1007/978-3-319-19830-9_4. ISBN: 978-331919829-3.
- Sourina, O. (2007). Visual spatio-temporal function-based querying. *The Visual Computer*, 23(5):335–346. DOI: 10.1007/s00371-007-0108-9.
- Space.com (2024). Large hadron collider: facts about the world’s greatest particle accelerator. Available at <https://www.space.com/large-hadron-collider-particle-accelerator>, last accessed on June 2025.
- Teplá, M., Teplý, P., and Šmejkal, P. (2022). Influence of 3D models and animations on students in natural subjects. *International Journal of STEM Education*, 9(65).
- Tornincasa, S. and F, D. (2010). The future and the evolution of cad. In *Proceedings of the 14th International Research/Expert Conference, "Trends in the Development of Machinery and Associated Technology", (TMT), 2010*, pages II–1.
- Tsang, M., Morris, N., and Balakrishnan, R. (2004). Temporal thumbnails: Rapid visualization of time-based viewing data. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '04), May 2004, Gallipoli, Italy, AVI '04*, pages 175–178. DOI: 10.1145/989863.989890. ISBN: 978-1-58113-867-2.
- Tufte, E. R. (2001). *The visual display of quantitative information*, pages 161–176. Graphics Press, Cheshire, CT, 2nd edition. ISBN: 978-1930824133.

- University of Washington (2024). History of the northwest region. Available at https://courses.washington.edu/eatreun/html/history/h_nw.html, last accessed on June 2025.
- Vergara Fernandez, E., Boussard, V., Ansel, A., Kauniskangas, A., Baltasar Dos Santos Pedrosa, F., Bernardini, M., Chemli, S., Coupard, J., Ronfaut, V., and Tock, J.-P. (2023). Scheduling tools development to manage CERN accelerators programmed stops and facilities installations. In *Proc. 14th International Particle Accelerator Conference, (IPAC), 2023, Venice*, number 14 in IPAC'23 - 14th International Particle Accelerator Conference, pages 2364–2366. JACoW Publishing, Geneva, Switzerland. DOI: 10.18429/JACoW-IPAC2023-TUPM073; ISBN: 978-3-95450-231-8.
- Vranic, D. V., Saupe, D., and Richter, J. (2001). Tools for 3D-object retrieval: Karhunen-Loeve transform and spherical harmonics. In *Proceedings of the IEEE Fourth Workshop on Multimedia Signal Processing (MMSP), October 2001, Cannes, France*, number 01TH8564, pages 293–298. DOI: 10.1109/MMSP.2001.962749.
- Wang, H., Zhang, X., Lv, H., and Xu, X. (2016). Spatial-temporal data modeling and visualizing method for UUV environmental perception. In *Proceedings of the 28th Chinese Control and Decision Conference (CCDC), May 2016, Yinchuan, China*, number 123193, pages 5395–5402. Institute of Electrical and Electronics Engineers. DOI: 10.1109/CCDC.2016.7531962.
- Wen, W. and Lin, X. (2010). A managed object based method for representing interactive and dynamic features. In *Proceedings of the 18th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL), November 2010, New York, NY*, number 18, pages 394–397, New York, NY. ACM. DOI: 10.1145/1869790.1869845. ISBN: 978-1-4503-0428-3.
- Xia, J., Zhao, J., Sheeley, I., Christopher, J., Wang, Q., Guo, C., Zhang, J., Ebert, D. S., Chen, Y. V., and Qian, Z. C. (2014). AnnotatedTimeTree: Visualization and annotation of news text and other heterogeneous document collections. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST), October 2014, Paris, France*, pages 337–338. DOI: 10.1109/VAST.2014.7042554.
- Yang, B., Li, Q., and Zuo, X. (2005). An algorithm for the progressive transmission of three-dimensional models over the Internet. In *Proceedings of the SPIE International Society for Optical Engineering: MIPPR 2005: Geospatial Information, Data Mining, and Applications (SPIE), October 31 - November 2, 2005, Wuhan, China*, volume 6045. DOI: 10.1117/12.651342.

- Ye, F., Wang, H., Ouyang, S., Tang, X., Li, Z., and Prakash, M. (2012). Spatio-temporal analysis and visualization using SPH for dam-break and flood disasters in a GIS environment. In *Proceedings of the International Symposium on Geomatics for Integrated Water Resource Management (GIWRM), October 2012, Lanzhou, China*, pages 1–6. DOI: 10.1109/GIWRM.2012.6349636.
- York University Computing Museum (2024). Evolution of 3D visualization: Introduction of animations. Available at <https://museum.eecs.yorku.ca/items/show/340>, last accessed on June 2025.

Appendix A

User Testing Script

A.1 Introduction

Hello, good morning/afternoon. Thank you very much for reserving some of your time to help us today.

Please prepare your laptop and ensure you can connect to <https://dev-oss-coordination.web.cern.ch/visualization>.

Please do not use Firefox for faster performance; use Chromium or Webkit browsers instead.

A.2 Informed Consent Form

The informed consent form was presented as follows.

A.2.1 Introduction:

You are invited to participate in a research study to evaluate the usability of a web-based visualization system designed to display 3D models of complex machines and facilities at CERN. This experiment is part of a Master's Dissertation aimed at making the system able to represent changes in the composition of the facilities according to an evolutive planning. Your participation will help us understand how effectively users can interact with the system and extract information from its visualizations.

A.2.2 Voluntary Participation:

Participation in this experiment is entirely voluntary. You are free to decide whether or not to take part. If you choose to participate, you may withdraw at any time during the experiment without any consequences or need to provide a reason. To withdraw, simply inform the researcher, and any data collected from you up to that point will be deleted and not used in the study.

A.2.3 Experiment Overview:

As a participant, you will be asked to use a web-based system to interact with a visualization of 3D models of the TCC2 building in the North Area. You will perform a series of tasks to be able to give short answers to several questions, such as identifying modifications or periods of activity, based on the information presented in the visualization. After completing the tasks, you will answer questions about your experience and provide feedback on the system.

A.2.4 Data Collection:

We will register every answer related to the tasks performed. Additionally, we will ask for general demographic information and your feedback on the system, including suggestions for improvement, to gain deeper insights into your experience.

A.2.5 Data Usage and Confidentiality:

All data collected during this experiment will be used solely for research purposes as part of this Dissertation. Your personal information and responses will remain confidential and will be anonymized. Only the researcher and the assistant that will help him conduct the user testing will have access to the raw data, which will be securely stored and deleted after the study is complete.

A.2.6 Contact Information:

If you have any question about the experiment or your participation, please contact: João Matos, up202006280@edu.fe.up.pt, *Master in Informatics and Computing Engineering student at Faculdade de Engenharia da Universidade do Porto*

A.2.7 Consent :

I have read and understood the information provided above. I understand that my participation is voluntary, that I can withdraw at any time without consequence, and that my data will be used only for research purposes. I consent to participate in this experiment, including the collection of demographic data and the provision of feedback for system improvements.

Name:

Signature:

Date:

A.3 Pre-test Questionnaire

I'm going to ask you a few questions to gather some background information. Is that okay?

1. What is your age?
2. What is your gender?

3. Have you previously used the 3D visualization tool at Scheduling Tools? (Yes/No/2D)
 - a. If you have, how frequently do you use it? (Used it just once / A few times per year / A few times per month / Every week)
4. On a scale from 1 to 5, where 1 means “no experience” and 5 means “very experienced,” how much experience do you have using software tools to visualize and manipulate 3D models?

A.4 Task-specific Questionnaire

Please be aware that this is a test environment, and I/João is not completely familiar with these machines and the work they perform. As such, fake group names, group colors, component names, tasks, etc., may have been used.

Pre-conditions:

- Select: tab 3D view, facility North Area, building TCC2, date 01/02/2026, and load the 3D visualization tool.

You can freely use and get a feel of the tool. Right-click rotates the camera, left-click moves forwards/backwards or right/left. You can also move using the WASD keys, and the RF keys move vertically. Press *Q* or the camera icon on the top left to move the camera with the mouse. The middle mouse button and scroll are both used to zoom in and out. You can Shift-scroll on the timeline to adjust the visible number of days.

You can drag the timeline to change the time range. You can also drag the white window or double-click on the timeline to change the selected time.

You can press “c” to expand the timeline.

%% Help navigating the model and identifying places/machines/components can be provided.

%% Help to understand what the temporal visual elements mean should not.

We want to test a temporal window functionality, which will be optional. The idea is to set a window with a period of interest, like 2 days into the past and 2 days into the future, to capture information about all tasks that take place during that interval.

If the equipment is colored and/or has an outline, it means some installation, removal, or maintenance task has occurred in the near past, is occurring, or will occur in the near future.

A.4.1 Task 2.1

Select the view “Camera 1”.

How many tasks of the types ‘Installation’, ‘Removal’, and ‘Maintenance’ start in February?

A: Maintenance: 3; Installation: 3; Removal: 6

A.4.2 Task 2.2

Re-select the view “Camera 1”. You can zoom and move around, but keep this camera as your reference. Hide timeline items by pressing “t”. You can move the time marker around.

%% The future and past interest must be 2 days.

What information do you have about the following equipment when the 4-day time window is set to the indicated dates?

The options are:

“no task”;

“before removal / installation / maintenance task”;

“ongoing removal / installation / maintenance task”;

“after removal / installation / maintenance task”;

a combination of previous options.

You can move the window around.

- 02/02:
 - MBN.23097 - A: ongoing maintenance
- 03/02:
 - MBN.23097 - both before and after maintenance tasks

Now please do not double-click equipment to expand task information.

- 11/02:
 - MTS.230928 - ongoing removal
 - MTS.230935 - before removal
- 14/02:
 - MTS.230928 - after removal
 - MTS.230942 - before removal
- 22/02:
 - MTS.230928 - before installation
- 26/02:
 - MTS.230928 - after installation
 - MTS.230935 - ongoing installation
 - MTS.230942 - before installation

Expanding task information is allowed. Press “t” to show tasks on the timeline again.
Adjust the time window “future interest” to 9 days.

- 17/02:
 - MTS.230935 - before installation and after removal

A.4.3 Task 2.3

Set the date in the timeline to 1st of March and select the view “Camera 4”. Hide the timeline items by pressing “t”. Without changing the date, how many installations start between 01/03 and 10/03?

A: 3

A.4.4 Task 2.4: Options Feedback

Pay attention to the options inside the Graphical User Interface on the right. Set the past and future interest to 2 days. Move and use any camera and date as you wish.

%% Suggest selecting “Camera 2” and the date 23/02.

Play with the different options under “Time Window” in the graphical user interface.

%% Here everything the participant asks can be explained.

%% To be faster, maybe perform the actions instead of the tester.

- Do you prefer Outlines on or off?
- Do you prefer Wireframes on or off?
- Which color option do you prefer? Do you think having custom colors is necessary?

Task 2.4.1 For installation tasks, the Temporal Degree of Interest (TDOI) / Time Window mechanism works as follows:

- When a task’s time interval begins to intersect with the time window, the equipment’s appearance transitions from fully transparent to opaque, adopting the task’s designated color.
- While the time marker remains within the task’s interval, the color remains constant.
- Once the time marker moves beyond the task but the time window still overlaps with the task’s interval, the color gradually fades to white.

The Alternative TDOI functions differently:

- As the time window begins to intersect with the task’s interval (from 0% to 100% overlap), the equipment’s opacity increases progressively.
- During full intersection (100% overlap), the equipment displays the task’s designated color.

- As the intersection decreases, the equipment's color gradually shifts toward white or gray.

Removal tasks show the same effects but in reverse order.

You can try them both again.

Which one do you prefer?

A.5 Post-test Questionnaire

A.5.1 System Usability Scale (SUS)

Please rate the following statements from 1 to 5, where 1 means “strongly disagree” and 5 means “strongly agree”.

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

A.5.2 Open Feedback

1. What were the worst parts of the experience?
2. Do you think that the temporal features of the system are useful to plan or execute maintenance activities?
3. Do you have any comments, including suggestions for improvement or feature Ros1 feature recommendation?

Appendix B

Questionnaire Answers

Table B.1: Questionnaire answers excluding open-ended questions (Part 1: Demographic Information).

User ID	Age	Gender	Previous Use	Frequency of Use	Experience
1 - Pilot	27	Female	Only the 2D tool	Not anymore	3
2	59	Male	Only the 2D tool	At least once a year	5
3	23	Female	No		1
4	53	Male	No		1
5	50	Female	No		3
6	57	Male	Only the 2D tool		2
7	47	Male	Only the 2D tool		5
8	63	Male	No		4
9	32	Male	No		5
10	53	Male	No		3
11	26	Female	No		1
12	51	Male	No		5
13	48	Female	No		1
14	25	Male	No		4
15	24	Male	No		5
16	52	Male	Only the 2D tool		3
17	38	Male	No		2
18	61	Male	No		1
19	59	Male	No		3
20	23	Male	No		3
21	40	Male	No		4
22	41	Female	No		5
23	38	Male	No		5
24	40	Male	No		5
25	37	Female	No		4
26	54	Male	Yes	At least once a year	1
27 - Incomplete	45	Male	No		5
28 - Incomplete	60	Male	No		1

Table B.2: Questionnaire answers excluding open-ended questions (Part 2: Tasks 1 to 3). Number of maintenance, installation and removal tasks in February.

User ID	T1: #Maintenance	T2: #Installations	T3: #Removals
1 - Pilot	3	3	10
2	3	3	6
3	3	3	6
4	3	2	6
5	3	3	6
6	3	3	7
7	3	3	6
8	3	3	6
9	3	3	6
10	3	3	6
11	3	3	6
12	3	3	6
13	3	3	6
14	3	4	6
15	3	3	6
16	2	4	6
17	3	3	6
18	3	3	6
19	3	3	6
20	3	3	6
21	3	3	6
22	3	3	6
23	3	3	6
24	1	3	8
25	3	3	6
26	3	3	6
27 - Incomplete			
28 - Incomplete			

Table B.3: Questionnaire answers excluding open-ended questions (Part 3: Tasks 4 to 9). Status of equipment MBN.23097 on 02/02 [T4] and 03/02 [T5]; status of MTS.230928 [T6] and MTS.230935 [T7] on 11/02; and status of MTS.230928 [T8] and MTS.230942 [T9] on 14/02. B, O and A mean, respectively, "before", "ongoing" and "after". R, I and M mean, respectively, "removal", "installation" and "maintenance".

User ID	T4	T5	T6	T7	T8	T9
1 - Pilot	O.M.	B.M., A.M.	O.R.	O.R.	B.R.	B.R.
2	O.M.	B.M., A.M.	O.R.	A.R.	B.M., B.R., B.A.	O.R.
3	O.M.	A.M., O.M.	O.R.	O.R.	A.R.	B.R.
4	O.M.	B.M., A.M.	O.R.	A.R.	B.R.	O.R.
5	O.M.	B.M., A.M.	O.R.	A.R.	B.R.	B.R.
6	O.M.	A.M., B.M.	O.R.	A.R.	B.R.	B.R.
7	O.M.	B.M., A.M.	O.R.	A.R.	B.R.	O.R.
8	O.M.	B.M., A.M.	O.R.	O.R.	B.R.	O.R.
9	O.M.	B.M., A.M.	O.R.	A.R.	B.R.	B.R.
10	O.M.	B.M., A.M.	O.R.	A.R.	B.R.	O.R.
11	O.M.	B.M., A.M.	O.R.	A.R.	B.R.	B.R.
12	O.M.	B.M., A.M.	O.R.	A.R.	B.R.	B.R.
13	O.M.	B.M., A.M.	O.R.	A.R.	B.R.	B.R.
14	O.M.	B.M., A.M.	O.R.	O.R.	B.R.	O.R.
15	O.M.	B.M., A.M.	O.R.	A.R.	B.R.	B.R.
16	O.M.	A.M., B.M.	O.R.	A.R.	B.R.	B.R.
17	O.M.	B.M., A.M.	O.R.	O.R.	B.R.	O.R.
18	O.M.	B.M., A.M.	O.R.	O.R.	B.R.	B.R.
19	O.M.	B.M., A.M.	O.R.	O.R.	B.R.	B.R.
20	O.M.	B.M., A.M.	O.R.	A.R.	B.R.	O.R.
21	O.M.	B.M., A.M.	O.R.	A.R.	B.R.	O.R.
22	O.M.	B.M., A.M.	O.R.	A.R.	A.R.	O.R.
23	O.M.	A.M., B.M.	O.R.	A.R.	O.R.	O.R.
24	O.R.	O.M.	B.R.	O.R.	B.R.	B.R.
25	O.M.	B.M., O.M.	O.R.	O.R.	B.R.	O.R.
26	O.M.	B.M., A.M.	O.R.	O.R.	B.R.	O.M.
27 - Incomplete						
28 - Incomplete	O.M.					

Table B.4: Questionnaire answers excluding open-ended questions (Part 4: Tasks 10 to 15). Status of equipment MTS.230928 on 22/02 [T10]; status of MTS.230928 [T11], MTS.230935 [T12] and MTS.230942 [T13] on 26/02; status of MTS.230935 on 17/02 [T14]; and number of installations in March [T15]. B, O and A mean, respectively, "before", "ongoing" and "after". R, I and M mean, respectively, "removal", "installation" and "maintenance".

User ID	T10	T11	T12	T13	T14	T15
1 - Pilot	B.A., A.R.	O.A.	A.A.	O.A.	O.A.	3
2	nothing	O.A.				
3	B.A., A.R.	A.A.	A.A.	A.A.	B.A.	3
4	B.A., A.R.	A.A.				
5	B.A., A.R.	O.A.	B.A.	O.A.	A.A.	3
6	B.A., A.R.	O.A.				
7	O.R., B.A.	A.A.	A.A.	O.A.	B.A.	2
8	B.A., A.R.	O.A.				
9	B.A., A.R.	B.A.	A.A.	O.A.	B.A.	3
10	B.A., A.R.	O.A.				
11	B.A., A.R.	B.A.	A.A.	O.A.	O.A.	3
12	O.R., B.A.	O.A.				
13	B.A., A.R.	A.A.	B.A.	O.A.	A.A.	2
14	B.A., A.R.	O.A.				
15	B.A., A.R.	A.A.	A.A.	O.M.	B.A.	4
16	B.A., A.R.	O.A.				
17	A.R., B.A.	O.A.	A.A.	O.A.	B.A.	3
18	A.R., B.A.	O.A.				
19	B.A., A.R.	O.A.	A.A.	O.A.	O.A.	3
20	B.A., A.R.	A.A.				
21	B.A., A.R.	O.A.	A.A.	O.A.	B.A.	3
22	B.A., A.R.	O.A.				
23	A.R., O.A.	O.A.	A.A.	O.A.	B.A.	3
24	O.R.	B.A.				
25	B.A., A.R.	O.A.	A.A.	O.A.	B.A.	3
26	O.R., B.A.	O.A.				
27 - Incomplete			A.A.	O.A.	O.A.	3
28 - Incomplete						

Table B.5: Questionnaire answers excluding open-ended questions (Part 5: Visual Preferences).

User ID	Outlines	Wireframes	Preferred Colors	Custom Colors Needed
1 - Pilot	On	Off	Green and Magenta	No
2	Off	On	Green and Magenta	Yes
3	On	Off	Blue and Red	Yes
4	On	Off	No preference	No
5	On	On	No preference	No
6	On	On	Green and Magenta	Yes
7	On	On	Green and Magenta	Yes
8	Off	Off	Green and Magenta	No
9	On	On	Green and Magenta	Only as a colorblind option
10	On	On	Green and Magenta	Only as a colorblind option
11	Off	On	Green and Magenta	No
12	On	On	No preference	No
13	On	On	No preference	No
14	Off	On	No preference	No
15	On	On	No preference	No
16	On	Off	No preference	Yes
17	On	Off	No preference	Only as a colorblind option
18	On	Off	Blue and Red	No
19	On	On	Blue and Red	Only as a colorblind option
20	On	Off	Blue and Red	No
21	On	On	Blue and Red	Only as a colorblind option
22	On	Off	Blue and Red	Only as a colorblind option
23	On	On	Blue and Red	Yes
24	On	On	Blue and Red	No
25	On	On	Blue and Red	Yes
26	On	On	No preference	Yes
27 - Incomplete	On			
28 - Incomplete				

Table B.6: Questionnaire answers excluding open-ended questions (Part 6: Temporal Degree of Interest preference and initial settings). The "Wireframes" column indicates whether they were turned on (1) or off (0) at the beginning of each test. The "Outlines" column indicates whether they were turned on green and magenta (1) or blue and red (0) at the beginning of each test.

User ID	TDOI Preference	Wireframes	Outlines
1 - Pilot	Standard	0	0
2	Standard	1	1
3	Alternative	1	1
4	Standard	1	1
5	Standard	1	1
6	Standard	1	1
7	Standard	1	1
8	Standard	1	1
9	Standard	1	1
10	Both valuable	1	1
11	Both valuable	1	1
12	Both valuable	1	1
13	Standard	0	0
14	Standard	0	0
15	Standard	0	0
16	Standard	0	0
17	Both valuable	1	1
18	Standard	0	0
19	Standard	0	0
20	Standard	0	0
21	Standard	0	0
22	Standard	0	0
23	Both valuable	0	0
24	Standard	0	0
25	Standard	0	0
26	Standard	0	0
27 - Incomplete		1	1
28 - Incomplete		0	0

Table B.7: Questionnaire answers excluding open-ended questions (Part 7: System Usability Scale).

[illegible]