



JOÃO PAULO CABETE GONÇALVES DIOGO
Bachelor of Computer Science and Engineering

INTEGRATING 3D OBJECTS AND POSE ESTIMATION FOR MULTIMODAL VIDEO ANNOTATIONS

MASTER IN COMPUTER SCIENCE
NOVA University Lisbon
<September>, <2022>



INTEGRATING 3D OBJECTS AND POSE ESTIMATION FOR MULTIMODAL VIDEO ANNOTATIONS

JOÃO PAULO CABETE GONÇALVES DIOGO

Bachelor of Computer Science and Engineering

Adviser: Nuno Manuel Robalo Correia
Full Professor, NOVA University of Lisbon

MASTER IN COMPUTER SCIENCE

NOVA University Lisbon
⟨September⟩, ⟨2022⟩

Integrating 3D objects and pose estimation for multimodal video annotations

Copyright © João Paulo Cabete Gonçalves Diogo, NOVA School of Science and Technology, NOVA University Lisbon.

The NOVA School of Science and Technology and the NOVA University Lisbon have the right, perpetual and without geographical boundaries, to file and publish this dissertation through printed copies reproduced on paper or on digital form, or by any other means known or that may be invented, and to disseminate through scientific repositories and admit its copying and distribution for non-commercial, educational or research purposes, as long as credit is given to the author and editor.

ACKNOWLEDGEMENTS

Firstly, I would like to extend my deepest gratitude to the Faculdade de Ciências and Tecnologias, NOVA University of Lisbon. I am privileged to have received such overwhelming support from both a personal and academic level. I am grateful as well for everything I encountered throughout my education. Even if, at times, obstacles seemed so challenging to overcome, no doubt overcoming each of them shaped me to become a better overall professional and person.

Additionally, I must express my appreciation to all of my professors, especially professor Nuno Correia for the patience, guidance, and support while writing this thesis, as well as Rui Rodrigues, who accompanied me in this project. Similarly, I am thankful for the *WEAVE* project and respective partners for the experience I was honored to have [1].

The most profound thank you to my biggest supporters in life to whom I owe too much to pay in a single lifetime: Manuela, my father, and my mother. Every day reminds me of how fortunate I am to share your company, wisdom, and each of my failures and accomplishments. To the best covid call partner, and outstanding acrobat reader software, I am thankful to have you in my life Inês.

To my medical experts, Inês and Guilherme, I appreciate our lifetime friendship contract and look forward to being a future patient. To Pedro, Filipe, Guilherme, Catarina, Manuel, Jorge, Miguel, Rodrigo, António, Rui, Carolina, Francisco, and Daniel, thank you for every adventure, laugh, and meal we have ever had together. Lastly, I want to thank the best person this university has given me, Diogo. To everyone here and others who I definitely forgot to mention, thank you for motivating me to do my best in everything I do.

*“You cannot teach a man anything; you can only help him
discover it in himself.” (Galileo)*

ABSTRACT

With the recent technological advancements, using video has become a focal point on many ubiquitous activities, from presenting ideas to our peers to studying specific events or even simply storing relevant video clips. As a result, taking or making notes can become an invaluable tool in this process by helping us to retain knowledge, document information, or simply reason about recorded contents.

This thesis introduces new features for a pre-existing Web-Based multimodal annotation tool, namely the integration of 3D components in the current system and pose estimation algorithms aimed at the moving elements in the multimedia content. Therefore, the 3D developments will allow the user to experience a more immersive interaction with the tool by being able to visualize 3D objects either in a neutral or 360° background to then use them as traditional annotations. Afterwards, mechanisms for successfully integrating these 3D models on the currently loaded video will be explored, along with a detailed overview of the use of keypoints (*pose estimation*) to highlight details in this same setting.

The goal of this thesis will thus be the development and evaluation of these features seeking the construction of a virtual environment in which a user can successfully work on a video by combining different types of annotations.

Keywords: Video Annotation, Note-making, Virtual 3D Models, Pose Estimation, Multimodal Interfaces, HCI, Ubiquitous Environments.

RESUMO

Ao longo dos anos, a utilização de vídeo tornou-se um aspecto fundamental em várias das atividades realizadas no quotidiano como seja em demonstrações e apresentações profissionais, para a análise minuciosa de detalhes visuais ou até simplesmente para preservar vídeos considerados relevantes. Deste modo, o uso de anotações no decorrer destes processos e semelhantes, constitui um fator de elevada importância ao melhorar potencialmente a nossa compreensão relativa aos conteúdos em causa e também a ajudar a reter características importantes ou a documentar informação pertinente.

Efetivamente, nesta tese pretende-se introduzir novas funcionalidades para uma ferramenta de anotação multimodal, nomeadamente, a integração de componentes 3D no sistema atual e algoritmos de *Pose Estimation* com vista à deteção de elementos em movimento em vídeo. Assim, com estas *features* procura-se proporcionar uma experiência mais imersiva ao utilizador ao permitir, por exemplo, a visualização preliminar de objetos num plano tridimensional em fundos neutros ou até 360° antes de os utilizar como elementos de anotação tradicionais.

Com efeito, serão explorados mecanismos para a integração eficiente destes modelos 3D em vídeo juntamente com o uso de *keypoints* (*pose estimation*) permitindo acentuar pormenores neste ambiente de visualização. O objetivo desta tese será, assim, o desenvolvimento e avaliação continuada destas funcionalidades de modo a potenciar o seu uso em ambientes virtuais em simultâneo com as diferentes tipos de anotações já existentes.

Palavras-chave: Anotações em vídeo, Modelos 3D, Interfaces Multimodais, Deteção de Movimento.

CONTENTS

List of Figures	xvii
List of Tables	xix
1 Introduction	1
1.1 Context	1
1.2 Motivation and Problem Definition	1
1.3 Research Challenges and Objectives	3
1.4 Contributions	3
1.5 Document Structure	4
2 Related Work	7
2.1 Video Annotation	7
2.1.1 Similar Systems and Modalities	8
2.1.2 Annotation Types	11
2.2 3D in Virtual Environments	12
2.2.1 Concepts and Possible Applications	13
2.2.2 Web-based Development	17
2.3 Motion Tracking and Estimation	21
2.3.1 Concepts	21
2.3.2 Pose Estimation paradigms	22
2.3.3 Technologies and Applications	24
3 Design and Implementation	27
3.1 Design	27
3.2 Implementation	31
3.2.1 System Overview	31
3.2.2 3D Model Visualization	34
3.2.3 Pose Estimation	45

CONTENTS

4	Evaluation and Results	59
4.1	Preliminary User Tests	59
4.1.1	Case Study: Traditional Dances	59
4.1.2	Case Study: Basketball	62
4.2	Final User Tests	66
4.2.1	Participants and Evaluation Method	67
4.2.2	Usability Scores	67
4.2.3	Results and Discussion	69
5	Conclusions and Future Work	75
5.1	Conclusions	75
5.2	Future Work	76
	Bibliography	79
	Appendices	
A	Usability Questionnaire <i>WEAVE Online</i>	89
B	Usability Test Guide <i>WEAVE Online</i>	99
C	Sports Guide <i>Use Case: Basketball</i>	103
D	Usability Test Guide <i>Final Evaluation</i>	107
E	Consent Form for Usability Test <i>Final Evaluation</i>	111
F	Usability Questionnaire <i>Final Evaluation</i>	113

LIST OF FIGURES

1.1	Sketch of the MotionNotes annotation tool.	2
2.1	Input and output modalities for multimodal interaction [16].	8
2.2	Examples of annotations tools using different types of annotations (e.g., ink strokes, text, marks) [24, 25].	10
2.3	Integration of various annotations types (e.g., text, drawings, 3D) on a video element.	13
2.4	Obtaining 3D models examples: real-world direct extraction, software-based designing and pre-existing models.	16
2.5	Tree representation of the Three.js structure [52].	19
2.6	Gltf standard format structure as presented by Miao et al. [54].	20
2.7	Simulation of the detected keypoints and an actual posterior segmentation available in the COCO dataset.	21
2.8	Visual representation of different approaches for multi-person pose estimation based on [62].	23
2.9	Common pose estimation models explored by T. L. Munea et al. [64].	24
2.10	Examples of research using PoseNet and OpenPose [65, 69–72].	25
3.1	WebGL, Three.js and gLTF brief summary and their connection.	28
3.2	Example of PoseNet (left) and OpenPose (right) pose estimation inference.	29
3.3	Initial prototype of the MotionNotes annotation tool.	32
3.4	Simple overview of the client-server architecture implemented on MotionNotes	33
3.5	Interface for the 3D model visualizer containing neutral (left) and 360° (right) interactive backgrounds.	35
3.6	Illustration of the frustum view in eye coordinates by Martin Kraus.	36
3.7	Simple interactions over 3D elements on the 3D Model Manager interface.	37
3.8	Interfaces for 3D model usage: personal (left) and public 3D models (right).	38
3.9	Uploading 3D models operations execution flow.	39
3.10	Cube map for a general skybox and Pé de Xumbo’s studio environments.	41

LIST OF FIGURES

3.11 Interface to add annotations (center) and respective annotation tracks (highlighted).	42
3.12 Adding and customizing 3D annotations.	43
3.13 Settings interface for annotation-specific options.	44
3.14 PoseNet (top) vs OpenPose (bottom) examples of keypoint estimation. . . .	46
3.15 Settings interface for extra pose estimation options.	47
3.16 Pose estimation client-server's communication overview.	48
3.17 Pose estimation keypoint data generation.	50
3.18 Example of a dynamic annotation (arrow - elbow).	52
3.19 Motion UI interface examples.	53
3.20 Keypoint tracking across multiple frames.	56
3.21 Pose Estimation 3D tracking layer.	57
4.1 Sample annotation scheme used to introduce 3D functionalities in the described workshop.	61
4.2 Sample frames selected in the workshop to demonstrate a player's shooting motion.	62
4.3 MotionUI images portraying the executed basketball shot motion (<i>Person 0</i>). .	65
4.4 Answers for the comfort level using video annotations.	67
4.5 SUS questions regarding system complexity.	68
4.6 Final user tests results.	70
4.7 Results from statements 18 to 21 - 3D annotation.	71
4.8 Results from statements 23 to 27 (left to right, respectively) - Pose Estimation.	72

LIST OF TABLES

3.1	API Endpoints	34
3.2	OpenPose’s most prominent flags.	49
4.1	SUS scores.	68

INTRODUCTION

In this initial chapter, the context in which this thesis is integrated is introduced as well as the motivation behind it, previously developed prototypes, a proposed solution, the expected contributions, and the document's structure.

1.1 Context

This thesis is inherently connected to a large-scale European research project (WEAVE - Widen European Access to cultural communities Via Europeana) in collaboration with NOVA LINCS and the Department of Computer Science of the NOVA University of Lisbon. Financed by the CEF Connecting Europe Facility Programme, this initiative aims to expand collaborative research among the international partners, thereby seeking to enrich Europeana¹ through the heritage of cultural communities. The project's main objective is to further develop a pre-existing video annotation tool (MotionNotes²) by adding 3D elements for versatile use such as in large-scale immersive model visualization (or for interactive, personalized 3D objects), as well as pose estimation as a means to enrich multimedia content.

Moreover, this document intends to provide a captivating and engaging narrative showing the usefulness behind those features and how they can potentially improve productivity in the working process.

1.2 Motivation and Problem Definition

Productive communication, in the sense of passing the message from one recipient to another as effectively as possible, has seen continuous adjustments over the years. As a result, one such adjustment has been the increased use of multimedia elements like video, which can provide the recipient(s) of the messages being conveyed with a better understanding of such contents.

¹Europeana was created by the *European Union* with the goal of protecting cultural heritage contents: <https://www.europeana.eu/en>.

²<https://motion-notes.di.fct.unl.pt/>

Intuitively, we can easily observe this by thinking of most presentations we see nowadays, usually containing some visual aid such as PowerPoint to captivate the audience's attention. Similarly, using video as a complement to audio for online communication can significantly improve the interaction between the parties involved by stimulating the participant's engagement further [2, 3].

The need for note-taking or note-making is also present in our everyday lives, whether for leisurely activities, hobbies, or even in the work environment. Admittedly, using pen and paper remains a viable means of taking and making notes and is still common in contexts such as education. However, most of us now have access to different tools enabling us to easily annotate on a virtual setting instead of using the conventional pen and paper, thereby allowing applications such as Samsung Notes to have well over a billion downloads³. Furthermore, some device-based technologies have shown the relevance of in-class note-taking as vital in the learning process and how such devices can go beyond the limitations of physical annotations and even help people with disabilities [4, 5].

Annotation tools that employ multimedia elements do exist, yet they tend to focus primarily on simple text/drawing annotations. There is thus a lack of complete multimodal systems using other types of annotations, which this thesis will introduce. In addition, the resulting functionalities are strongly supported by modern technological capabilities. Even though Moore's law⁴ is no longer applicable for the CPU (Central Processing Unit), the GPU has seen a growth in performance compared to the CPU's growth in recent years [6]. These increased capabilities in the Graphics Processing Unit have enabled its wide use in fields such as Artificial Intelligence and improved the overall quality of computer graphics.

As a result, the development of 3D annotations in this context of annotating over multimedia elements becomes feasible due to the processing power behind the rendering of potentially complex 3D models and objects. Similarly, the integration of pose estimation algorithms on such a system can also benefit from these improved capabilities, for instance, by allowing the smooth detection of *keypoints* on successive frames on a video.

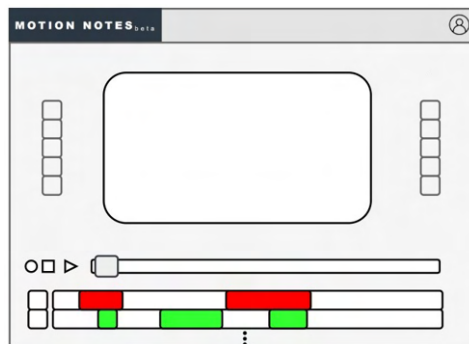


Figure 1.1: Sketch of the MotionNotes annotation tool.

³*Samsung Notes* is an application for Android devices. For more information, consult Google Play Store.

⁴Moore's Law stated the number of transistors that fit on a microchip would double every 3/2 years

Both of these features have great potential because they can further enrich multimedia content by adding more expressiveness, for example, by highlighting hidden and unnoticed details or unveiling specific patterns in movements not perceivable before.

1.3 Research Challenges and Objectives

This thesis will contribute to the development of new features on a pre-existing multi-modal Web-Based annotation tool (fig. 1.1) that currently enables a given user to work on multimedia content by combining various types of annotations, namely: text, ink strokes, audio, and personalized imagery (marks) which will be further detailed in-depth on a future chapter.

Initially, the primary goal consists of integrating 3D models on the existing system so as to allow users to place 3D objects on the currently selected video. However, various challenges arise when analyzing the best way to achieve this objective, such as how to visualize a 3D model before actually inserting it onto the scene or what kinds of interactions with these objects should be made possible. While the latter can intuitively be solved by allowing the rotation, translation, and resize of those objects to first observe them, there is a need to create a simple 3D visualizer possibly permitting those same interactions. Another challenge is how to represent those elements, and for that, care must be taken when choosing what formats should be accepted (e.g., *.obj*, *.gltf*, or *.glb* file extensions).

In addition, combining pose estimation applied to the consecutive set of frames on the loaded video with the various annotation features is also one of the objectives. Evidently, some challenges can also be found here since the detection of keypoints might be computationally demanding and thus lead to longer execution times. Similarly, placing the keypoints correctly and making a smooth transition between frames will definitely constitute an essential factor to consider.

Thus it is intended that both the 3D-based features as well as the pose estimation components be integrated seamlessly into the current annotation system while overcoming some of the challenges mentioned above in order to enhance the user experience. Moreover, gathering valuable feedback from users will be essential to better understanding the relevance inherent to these technologies and help guide further developments.

1.4 Contributions

As a result of the work presented throughout this thesis, the main contributions are as follows:

- **3D Model Visualiser** - Development of a 3D Model Visualizer that displays the currently selected 3D entities. Consequently, selecting a given 3D object makes it

easily observable on the screen. Some interactions, such as changing the object's orientation, are also possible in this environment.

- **Integrating 3D elements for annotations** - Integration of 3D-based annotations on the currently loaded video. Adding a 3D object to the current frame implies being able to place it anywhere on the scene with a customizable position, rotation and size.
- **360 environments for 3D object visualization** - Creation of mechanisms to handle 360° visualization of environments for 3D object visualization. As a result, this feature allows the user to switch the default neutral background in the 3D Model Visualizer with the intent of providing a more immersive experience.
- **Pose estimation on video content** - Implementation of features regarding pose estimation algorithms. Therefore, exploring and integrating existing models for *keypoint* detection with subsequent visualization is one of the central goals. Additionally, complementary features such as dynamic annotations and the automatic detection of specific action/movements are also integrated into the annotation tool.
- **System Evaluation and Publication** - Evaluation of the system through usability tests. Furthermore, attempts at contributing to the scientific community through the creation of scientific papers have led to the publication of two poster papers at the *IMX*⁵ and *MUM*⁶ International Conferences (Chapter 4). In addition, current developments seek to submit another research article in the form of a full paper.

1.5 Document Structure

This document's structure is divided into the following chapters:

- **Chapter 1 - Introduction:** This first chapter will discuss the context and motivation behind developing the previously mentioned features for a multimodal Web-Based annotation tool. It also introduces a brief description of the proposed solution, some of the approached challenges, and the main contributions that can be expected.
- **Chapter 2 - Related Work:** This chapter will discuss the fundamental concepts inherent to the development of features for this system (e.g., 3D annotations). Similarly, the analysis of related work in the field of HCI (Human-Computer Interaction) relevant to the context of this work and partially similar systems will also be studied in detail.
- **Chapter 3 - Design and Implementation:** In this chapter, the proposed solution is now explained in-depth by highlighting the intricacies behind the choices made in

⁵<https://imx.acm.org/2022/>

⁶<https://mum-conf.org/2022/>

the development process. The distinction between previously implemented features and the updated system will also be analysis subjects, including a summary of the work centered around the annotation system.

- **Chapter 4 - Evaluation and Results:** Both preliminary and final organized interview sessions and respective analysis are presented here. For each, a brief overview of the intended goals and participant's demographic is described, followed by a description of the feedback received throughout the discussion of the explored topics.
- **Chapter 5 - Conclusions and Future Work:** This last chapter discusses the conclusions regarding these features' development process and evaluation as well as possible developmental routes and improvements to be conducted further in the future.

RELATED WORK

The previous chapter introduced a brief overview of the context and motivation behind this thesis, as well as some challenges and goals related to this specific project in the *HCI* (Human-Computer Interaction) field of study. To gain a more detailed understanding of the different aspects of the development process portrayed throughout this document, this chapter will go over each of those factors in the following four sections.

The first section presents essential concepts and features used when annotating video content by exploring similar systems and tools. The second introduces the use of 3D elements in various contexts, different forms of representation, possible applications, and how they can enrich multimedia content. Lastly, the third section will study motion tracking and detection using pose estimation algorithms by describing key factors inherent in their implementation and integration.

2.1 Video Annotation

Throughout the years, children quickly learn the importance of note-taking in an academic environment, for instance, to document and keep track of the discussed subjects in class or even to later recall details that might otherwise go unnoticed. Similarly, adults in professional settings must often have a keen eye and attention to detail, thereby making annotations potentially vital in their respective fields of work. A valuable example is the need for note-taking in clinical and medical documentation areas. As a result, some attempts are continuously made to improve patient care, treatments, or even information sharing and research where annotation systems can be essential [7].

With the continuous technological growth in the past decades, people are now able to access technology in which the process of creating annotations can be done effortlessly (e.g., using personal smartwatches/smartphones or laptops). Consequently, annotating on a virtual setting opens possibilities on the types of content where the notes can be created.

One such multimedia element is video, which is widespread and comes with a broad range of potential uses. Besides allowing viewers to enjoy its content, popular platforms

such as YouTube are a great example of what can be achieved using video. For instance, by going beyond its straightforward use for simple content visualization and allowing students to use it as an extension of the available knowledge and information [8]. Another option worth exploring is the prospect of expanding the use of video for personal usage only. A shared video editing environment can be vital for sharing knowledge in a team or any collaborative setting, and as a result, existing tools already strive towards computer-supported collaborative video analysis [9]. This type of digital content can thus serve as a powerful aid in a variety of contexts. However some challenges will naturally arise when trying to conceive systems capable of integrating different types of annotations on video elements further explored below.

2.1.1 Similar Systems and Modalities

Conceiving a multimodal system as opposed to a unimodal one can intuitively provide the user flexibility over which modality to employ (fig. 2.1). Moreover, other advantages such as meeting most users' needs due to the variety of choices presented can consequently reach a broader range of users, which in turn may motivate the development of multimodal vs. unimodal systems [10, 11]. People also tend to process information more quickly when multiple modalities are available [12]. Even so, it may not translate to a significant increase in efficiency [13]. The need to interact with computers in a multimodal manner has long been sought after, perhaps due to the fact that when immersed with our surroundings, people employ multiple senses [14]. In addition, the rapid advancement in mobile technology provides opportunities for exploring these interactions beyond the standard keyboard-mouse usage, where systems such as Apple's Siri or Amazon's Alexa [15] are notable examples.

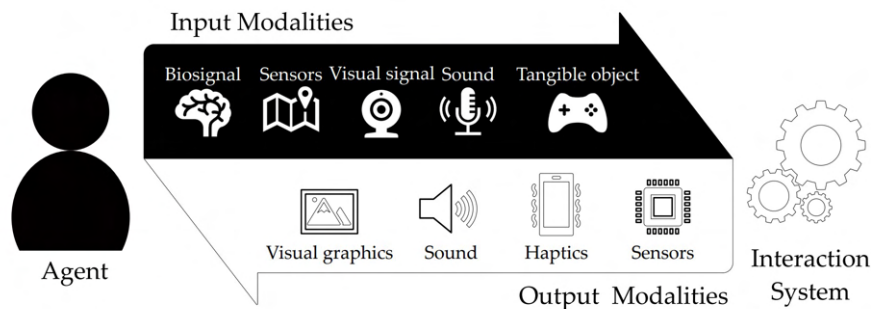


Figure 2.1: Input and output modalities for multimodal interaction [16].

However, when devising the types of modalities intended for the given system, care must be taken by considering the value behind such features and what specific goals they are trying to fulfill. For instance, even though efficiency is most often intended, it may not be the primary objective when developing multimodal systems [17].

Furthermore, although multiple interactions are possible, it does not necessarily translate to a user utilizing all modalities, which should be accounted for, especially when there are visible connections between them.

In the context of this thesis, the multimodal system in question aims to provide the user with a set of various annotation types, which enables various possible interactions over the selected video. For instance, at a given moment, a user might want to create an annotation using a text annotation or a speech annotation, thereby combining these two possible modalities. This latter one can be especially relevant in areas such as education, where teachers frequently need to move around the classroom either to help illustrate the current subject in proximity to certain learning materials or to get closer to a student during an explanation. As a result, teachers naturally move away from as well as towards their computers which may be straining at times and cause students to lose focus due to the gap between interactions. The use of voice commands can be a tremendous aid in these instances by partially removing such challenges [18].

Education areas can thus greatly benefit from these types of technologies. Consequently, the research and development of annotation systems for the academic environment is a constant topic of innovation and popularity [19]. There are many representative examples of annotation systems designed towards the educational field, namely the Microsoft Research Annotation System (*MRAS*). This tool was initially developed to support work done on video archives where users could interact collaboratively and asynchronously, targeted mainly at academic environments [20]. Here, users participated through the use of comment-based annotations on their work. Following a comment, other users were able to participate in that comment's respective thread, either public or privately. Eventually, *Microsoft* had to change their approach of viewing and developing this application from a flexible generic tool to a toolkit-guided platform for this type of interaction.

Nevertheless, this application served as a valuable reference for future annotation systems. The collaborative lecture annotation system, also known as CLAS, is one such system. Using computer-supported collaborative learning (CLCS), students work on pre-recorded video by selecting moments they deem relevant and annotate accordingly. Afterwards, the system gathers the collective data and amalgamates it into a group graph representing the crucial points identified by all the students, thus enabling students to opt between using their personal set of annotations and the group's resulting notes [21].

This system enforces the notion that learning while on a lecture presupposes being capable of identifying and synthesizing important concepts in order to better comprehend the studied subjects and strive for academic performance. Related research shows correlations between the use of video annotations created by students and positive results on academic evaluations such as tests and exams. Conversely, a more detailed analysis also reveals that other factors contribute to the overall grades and educational performance, such as test anxiety, and the way students approach the learning experience [22]. Likewise, it is essential to bear in mind that the link between the capabilities behind these

systems and the user's mindset when interacting with its features is of great significance. For instance, in this case, a student's willingness to focus and annotate during a lecture will undoubtedly impact the resulting learning experience and tool interaction.

In the medical context, students, doctors, and the healthcare personnel, in general, can also take advantage of note-making mechanisms to be applied in a wide range of situations. Medical trainees often resort to role-playing situations such as pretending to be a patient in order to comprehend and examine the patient's perspective. As a result, being able to share their experiences with other colleagues throughout this and other medical processes (e.g., surgical procedures) using video recordings can be particularly helpful. A. Pless et al. [23] describe how the use of annotations in such scenarios can improve the communication and comprehension of information in this environment. Interestingly, some trainees even found their colleague's notes more useful than their own, therefore reinforcing once again the relevancy of collaborative approaches in annotation tools.

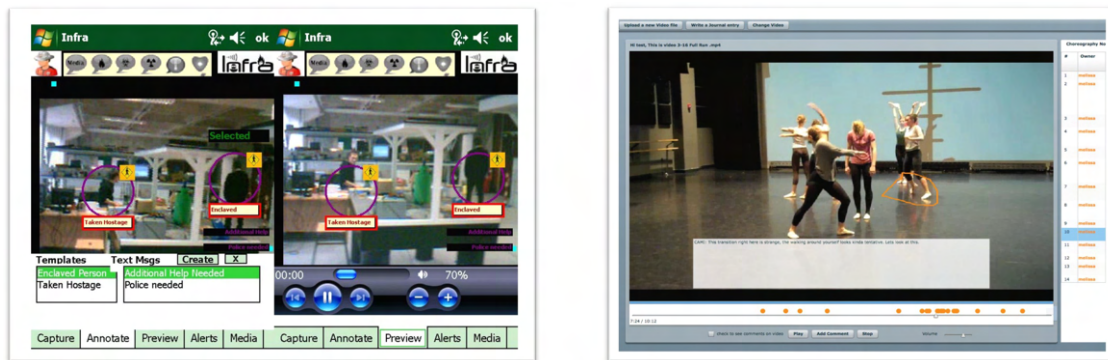


Figure 2.2: Examples of annotations tools using different types of annotations (e.g., ink strokes, text, marks) [24, 25].

Furthermore, in the case of emergency healthcare situations where first responders must be quick to arrive at the scene and act upon the incident, the use of efficient annotation mechanisms can be of extreme importance. Mobile devices such as PDAs¹ provide healthcare personnel with the means of transmitting information swiftly between first responders and their respective command posts. M. Bakopoulos et al. [24] verified that enriching video footage by inserting intuitive visual annotations such as alert signs and messages mitigated the possible miscommunication between the two by allowing an intuitive comprehension of the information being transmitted.

Besides academic and medical settings, performing arts is another excellent example where annotation systems can become an invaluable resource throughout the creative process. In this area, the act of reflecting upon artistic processes and learning moments can consequently encourage the use of robust methodologies for perfecting and improving overall performance, where video annotations can play a vital role [26]. Moreover,

¹PDA - Personal Digital Assistant.

the stages preceding the final performance, for instance, in music or dance, typically require multiple rehearsals in order to polish their execution. As a result, there is a need to optimize work during the creative process, considering the inevitable time limitations in studios/rehearsals. Tools such as the web-based Choreographers' Notebook [25] application allow choreographers and fellow partners to make a more efficient use of the resources they have at their disposal. Similar to the system where this thesis' project is integrated, annotations using text and ink strokes are available, allowing the user to combine them when working on a video. This multimodal annotation tool enables users to anchor annotations in a video's timeline at a specified moment, thereby enhancing the artist's ability to understand and contextualize said annotations. As a result, this system can be advantageous in the workflow of the creative processes, for example, by improving time efficiency and providing a collaborative environment where artists can intervene [25]. This tool's features and intended usage are deeply related to this project's goals since it is immersed in a similar artistic context containing a subset of the annotation types available on this project's system.

2.1.2 Annotation Types

Several of the mentioned annotation systems in the previous subsection relied upon different types of possible annotations. Interestingly, the most prevalent remain simple text annotations. However, each can serve diverse purposes and be more beneficial according to the context in which they are used. Earlier, it was discussed how speech interactions could potentially mitigate the educator's back and forth between their computers and the rest of the classroom. When mentioning the *Choreographers' Notebook*, the use of ink stroke annotations was also one of the possibilities given to users when interacting with this application. Additionally, in the presented case study [24], healthcare personnel controlled mobile devices to annotate during emergency medical situations where annotations using alert symbols (e.g., risk of fire, chemical spills, a person in danger) facilitated the communication amongst the parties involved. Therefore, a variety of types of annotations and their possible applications are inherent to most annotation systems. As a result, even though there may be similarities between the different types of annotations, it is important to comprehend their set of characteristics to utilize them fully on video content:

- **Text** - Featuring in most annotations systems, text annotations remain one of the most useful and utilized. In this thesis, despite both resulting in placing words on a visualization medium, text annotations, unlike speech annotations, will be referred to as written notes created, for instance, through the use of a keyboard. This form of explicitly displaying words on the screen is natural for most users. In a collaborative setting, comment-based mechanisms can be one way of enriching multimedia content as previously discussed [20].

- **Speech** - Creating speech annotations presupposes receiving audio as input and decoding it into written words, resulting in displayed text. Voice commands are also common in this scenario, usually preceding some form of trigger for the system to understand it may start processing information (e.g., MotionNotes, write (...)).
- **Ink Strokes** - With the growth in popularity of mobile devices (e.g., smartphones, tablets) in recent decades, the ability to easily write on interfaces using touch (besides the standard mouse), for instance, through the use of a certain type of stylus, has become quite prevalent. Using ink stroke annotations can thus be a comfortable and efficient way of note-making in the context of annotation systems.
- **Marks** - Mark annotations require the use of images or symbols. Inserting visual elements on video clips can provide for an intuitive and rapid understanding on the message trying to be conveyed. Moreover, these characteristics can be extremely important in time-sensitive situations [24].
- **Hyperlink** - This form of annotation can be regarded as an interactive text annotation, i.e., by clicking on the displayed text representing a given *URL*² the user will be redirected to that *URL*'s respective web page.
- **3D** - 3D annotations was one of the primary focuses throughout this project's implementation. Using 3D models representing virtual objects allows users to place them on top of the current video frame and, after that, control them. Besides being able to move a 3D model around the scene, resizing and rotating are some of the potential interactions that enable users to further observe possibly concealed details. Manipulating said objects in real-time can also provide a more engaging experience in collaborative situations or contexts requiring presentations.

Even though this thesis's initial focus is centered around developing mechanisms to create 3D annotations, retaining these concepts and the inherent features of these annotations is crucial. When working on video through this project's system (and similar ones), completely isolating different types of annotations may not be possible since integrating and combining multiple annotation types is quite frequent. In Fig. 2.3 we can observe annotation work on a professional sports environment where various annotations are inserted and co-exist with each other.

2.2 3D in Virtual Environments

Employing 3D elements in various contexts has been a subject of attention and research by diverse scientific fields. However, although this section will dedicate a few paragraphs to better describe possible applications and contextualize their use - thereby introducing relevant common concepts - it is essential to keep in mind that the focus will be centered

²Uniform Resource Locator - represents internet addresses of unique web resources.

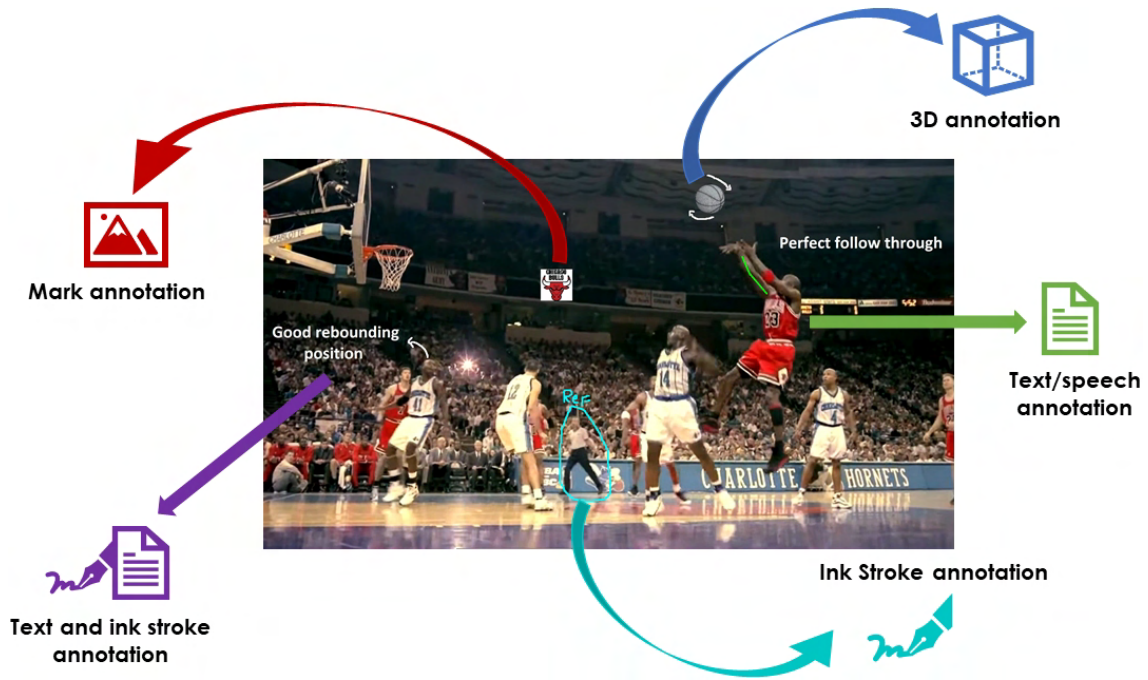


Figure 2.3: Integration of various annotations types (e.g., text, drawings, 3D) on a video element.

around 3D functionalities and their integration into the MotionNotes' system. For that reason, this next part will now illustrate topics such as in what contexts 3D models are helpful, how they can be created and transmitted, and details about their implementation and possible applications.

2.2.1 Concepts and Possible Applications

The potential and value in applying 3D-based features to real and virtual scenarios derive in great part from the possibilities behind being able to explore three-dimensional spaces by observing elements from different angles, moving them in various directions, and zooming in and out. In the past, using 3D models was limited mainly by the hardware and software capabilities, resulting in restrictions regarding their integration with other multimedia content. Nonetheless, as previously mentioned, the recent increase in computational power related to the improvements of graphics cards as well as the availability of high-speed internet connections opened multiple possibilities in this research area [6].

Video games played a significant role in what led to this increase in computational power. The ability to perform rapid rendering and creation of high-quality graphics was one of the motivations that eventually improved aspects such as the smoothness and enhanced visual representation of objects displayed throughout gameplay [27, 28]. However, transitioning the use of these capabilities to other fields of work allowed scholars and professionals to apply them to tackle a broader range problems, including parallel computation for large volumes of data as well as Artificial Intelligence. The latter is

especially relevant in the context of this thesis when we describe motion tracking and detection later on.

Due to these aspects, significant developments in integrating 3D elements in virtual scenarios such as in Virtual Reality (VR) and Augmented Reality (AR) environments are a constant topic of innovation. In fact, the recent popularity of VR systems has been accompanied by an increase in accessibility through affordable platforms such as Google Cardboard³ that allow a straightforward approach to using VR for mobile devices. Comparative studies show how these technologies may perform differently at times regarding factors such depth-perception [29]. Moreover, AR usually focuses more on interactions with real-world objects that could not be achieved in purely virtual environments [30]. However, both are relevant examples of the successful use of tridimensional features well accepted by the general public.

Several studies concentrate their attention on the possible applications of these tools for a variety of scientific and professional areas [31]. Most notably as it relates to the context of this thesis, employing 3D contents through VR and AR technologies either for cultural heritage applications or in the field of the performing arts has been thoroughly explored since it can facilitate the engagement and comprehension of visual information by the general public [32]. For instance, H. Southall et al. [33] describes their application for the recreation of a historical dance hall structure to study the virtual space and interpret the logistics behind its use. Similarly, A. Rácz et al. [34] detail how they can be employed to create a virtual exhibition on the early history of Hungarian ballet where three-dimensional objects such as vintage ballerina's clothing items are available to be seen and analyzed. This ability to reconstruct specific environments, objects, or structures - even those that do not presently existing - through the use of 3D elements is quite powerful and a common objective in this project's annotation system.

The previously mentioned annotations tools also enabled its possible applications to branch out into different professional settings (e.g., Education, Medicine, Performing Arts) due to its variety in possible uses. Likewise, 3D functionalities have long been explored ever since it was possible to do so. For instance, in clinical situations 3D imaging provides medical personnel a means of observing human tissue through different angles and perspectives otherwise impossible through the use of standard 2D representations [35, 36]. Similarly, in the education sector, students' focus and ability to learn is partially dependent on the learning materials as well as the educator's ability to convey the intended message. As a result, 3D-based educational resources can help improve their comprehension of the given subjects by providing a more visually immersive experience throughout the learning process [37]. In addition, STEM⁴ education and respective professional areas benefit from the student and professional's proficiency to reason using their spatial abilities, where 3D-based approaches can aid to develop these skills [38].

³<https://developers.google.com/vr/discover/cardboard>.

⁴STEM stands for Science, Technology, Engineering, and Mathematics.

The use of 3D-based objects across these and other contexts implicitly requires a way to obtain them. There are multiple ways to acquire these three-dimensional models and each with their implications. They are done mainly through their extraction from real-world scenarios, object creation on virtual environments using specialized tools (e.g., Blender), or using already created ones available on compatible platforms (fig. 2.4). Until recently, there were several limitations when trying to extract 3D models from the physical world in large part due to the hardware and software capabilities of the existing devices. However, now even the current iOS and Android systems are providing accessible use of innovative 3D-based features, thereby mitigating such challenges. Apple's object capture API⁵ makes it possible to create 3D objects based on the data obtained from a series of 2D images. This practical process of extracting 3D models from real-world objects by converting photographs with different angle representations of those physical items is called Photogrammetry. Even though care must be taken regarding correct image placement and proper surrounding lighting, its use can be invaluable in various scenarios [39]. Similarly, another valuable example is Samsung's 3D Scanner⁶, which enables users to create 3D model representations by scanning the surrounding object using a specialized camera⁷. This particular component of the mobile device relies upon the use of a depth sensor as a means of computing the time it takes light to return to it upon being reflected by the object - concept known as time of flight(*ToF*) - in order to capture the three-dimensional object successfully.

This latter approach is quite similar to the one used in depth cameras, also commonly referred to as RGB-D cameras. Intuitively, besides being capable of handling the RGB (Red, Green and Blue) components of the image, it can also detect the depth of the elements in the captured scene usually through the use of wavelengths in the range of the infrared (IR) spectrum [40]. The recent advancements in new sensing technology allow affordable depth sensors to become widely available, therefore impacting the way in which 3D scene reconstruction can be utilized using yet another technique [41]. A. Kanazaki et al. describe the possibilities behind using RGB-D datasets obtained from a depth camera, for instance, by computing object candidates using points clouds in the 3D scene space [42]. Point clouds constitute a common relevant concept across 3D model creation and visualization as they represent a set of data points in a three-dimensional space for a given object's representation. However, there is frequently a need for point cloud meshing, i.e., creating polygon models formed from points clouds by connecting points, thereby creating triangles. Besides being especially useful in Web-based content due to its compatibility with popular 2D and 3D rendering APIs (e.g., WebGL) - which will be covered in the next section - mesh-based approaches are becoming increasingly

⁵https://developer.apple.com/documentation/realitykit/capturing_photographs_for_realitykit_object_capture/.

⁶<https://www.samsung.com/global/galaxy/what-is/3d-scanner/>.

⁷<https://www.samsung.com/global/galaxy/what-is/3d-depth-camera/>.

popular due to their rapid rendering speed, compact model size and ability to create better visual effects when compared to the point-cloud-based approaches [43]. Consequently, several studies describe the inner workings of this process of converting real-world object representations to a virtual setting using depth-cameras, highlighting its straightforward use by both experts and amateurs alike [44, 45]. Additionally, such devices can facilitate expanding the use of static 3D objects to animated 3D models as well as their conversion to standard 3D file formats (e.g., *.fbx*, *.gltf*, *.glb*).

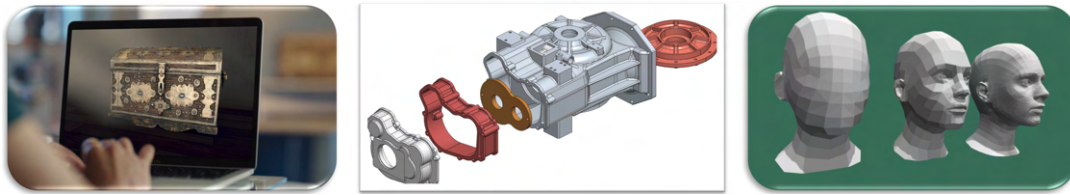


Figure 2.4: Obtaining 3D models examples: real-world direct extraction, software-based designing and pre-existing models.

Computer-aided Design (CAD) is yet another driving force for 3D modeling. This design process is a way of creating, modifying, and analyzing virtual 2D/3D models of real-world products with regards to the respective industry, usually to help with the manufacturing process. Ever since the Sketchpad - arguably the first CAD system - was first developed, allowing users to interact with the computer's monitor directly by drawing, numerous applications of CAD-based prototypes have been widely utilized. Improving the overall CAD system performance and features remains a constant focus for the professionals behind their development as problems including geometric accuracy, visualization quality and methods to identify and evaluate topological features of 3D models are an ongoing priority [46, 47]. Even so, various CAD systems continue to play a vital role, particularly in construction-related engineering projects such as for BIM (Building Information Modeling) usually supported by well-established software (e.g., AutoCAD⁸, Revit⁹). Y. Zhang et al. [48] presents a practical example of how CAD can help professionals in decision-making situations by potentially improving the integrity between the design and construction process simulation, thereby enhancing its visualization, which facilitates the early detection of deficiencies during the construction stage. Moreover, further 3D based environments and animations are also explored in this paper reinforcing once again the usefulness behind their possible applications when a detailed view of objects is needed.

There exist several other software tools used to generate three-dimensional items. As a result, even though artistic tools such as Blender¹⁰ are not CAD tools since they lack the precision and accuracy needed for manufacturing purposes, they present a viable

⁸<https://www.autodesk.com/products/autocad>

⁹<https://www.autodesk.com/products/revit>

¹⁰<https://www.blender.org/>

solution for 3D model creation, analysis, and visualization. Furthermore, this specific widespread toolset has become increasingly popular and of great significance when applied to cultural heritage contents, which is crucial as it relates to the *WEAVE*'s project goals. In this context, A. Guidazzoli et al. [49] describe how Blender's framework flexibility provides opportunities for cultural heritage-based scenarios to easily blend in with virtual environments and facilitates sharing as well as collaboration amongst its users.

Lastly, another common way of acquiring 3D models is through the use of appropriate platforms that publicly make these types of content available for the community. Notably, Sketchfab¹¹ is one of the most well-known platforms providing users with the ability to publish and discover 3D models. Moreover, one typical advantage of utilizing this and similar tools is the flexibility of the provided file formats, allowing users to choose from the multiple 3D format extensions the one(s) that better suit them.

Thus, across multiple areas and fields of work, there is clearly the motivation for developing means of acquiring 3D models and utilizing 3D-based visualization effortlessly to employ these mechanisms in a wide range of situations. Moreover, in the context of the MotionNotes annotation tool, utilizing 3D models allows users to augment their annotation work due to these models' ability to add extra unique information to the scene. As such, specific requirements - besides the intrinsic characteristics of the models - must be met when managing 3D models as time-framed digital annotations, which will be discussed further in the next chapter. Just now, this section explored different examples of approaches to obtain 3D models either by extracting them from real-world scenarios, using software-based designing of objects, or searching for pre-existing models (fig. 2.4). However, due to the Web-based nature of this project's system, it is essential to understand the behavior and implications of using certain features and popular APIs such as WebGL, which will be detailed next.

2.2.2 Web-based Development

Numerous advantages can motivate the Web-driven development of applications and software. Besides usually relying on programming languages known to most developers (e.g., HTML, CSS, and JavaScript), there is no need to download software in order to run content on the Web. Furthermore, web applications must run on any operating system, which makes it easier to meet the requirements of users that have different OS. As a result, different devices are able to use those applications despite having different inherent hardware and software specifications. This is especially relevant in the context of the MotionNotes annotator since it facilitates the flexibility users have of switching between annotation types. For instance, at a given time, a video can contain text annotations received from a personal laptop and later be added an ink stroke annotation created on a mobile device.

¹¹<https://sketchfab.com>

There are thus implicit requirements when developing content for web applications in order to successfully interact with the browser across different devices. As such, the emergence of HTML5 as well as the WebGL API provide developers new opportunities for the way content can be explored. The HTML5 is a widely used markup to facilitate presenting and structuring information on the Web. This standard text-encoding system, also known as the HTML Living Standard, is usually referred to in combination with the CSS and Javascript languages representing the basis for most Web-driven development. On the other hand, WebGL is a popular JavaScript API for rendering 2D and 3D graphics within most major browsers (e.g., Mozilla Firefox, Google Chrome, Microsoft Edge, Opera). Besides being able to perform 3D accelerated rendering in an HTML canvas without needing browser plug-ins, it is widely available for any developer as an open-source API/library and closely conforms to a pre-existent OpenGL embedded system (OpenGL ES) [50]. Despite some limitations, the OpenGL system contains many similar patterns to the ones in WebGL, such as the use of simple geometric primitives (e.g., points, triangles) to draw possibly complex three-dimensional structures still used to this day. Many relevant software tools are still using these APIs, namely, the Blender platform mentioned before for publishing and discovering 3D models that uses OpenGL, for instance, to draw the GUI (Graphical User Interface). As a result, the *Khronos group*¹² company responsible for the development of the WebGL system to tackle some of the constraints inherent to OpenGL by allowing it to run on the browser independently of the operating and window systems [51]. However, working directly with the WebGL API may add unnecessary overhead to the development of graphics for web applications.

Abstracting WebGL's low-level details can significantly improve the productivity in the work in progress. Consequently, the scientific community is constantly studying tools and mechanisms to try and prevent long periods of development while ensuring coding quality with shorter testing periods. Three.js¹³ is one such notable example of a successful application for creating and displaying animated 3D computer graphics in a web browser. Thus, the integration of the 3D features for the MotionNotes is supported by this Javascript 3D library, making it substantially more attainable to author complex 3D computer animations that display in the browser without the effort required for a traditional application. At the core of Three.js, there is scene graph structure, i.e., in this 3D engine, there is a hierarchy of nodes in a graph where each element defines its respective space. This tree-like structure simplifies the analysis of the elements present in the scene as reasoning about the relative position of three-dimensional objects becomes easier when looking at their surrounding elements. The scene graph supports the creation an interaction between various objects such as geometric objects (e.g., cubes, spheres), cameras, textures, lighting and controls [51].

The Three.js library offers numerous functions to conceive the different parts of the intended canvas. Consequently, there are many possible ways of interacting with the

¹²<https://www.khronos.org/>.

¹³<https://github.com/mrdoob/three.js/>.

virtual environment at hand. However, in order to first display visual information on the canvas, the following three components are needed:

- **Camera** - Firstly, properly customizing the camera's settings, including its position, field of view, and aspect ratio are vital so as to actually visualize the scene. The resulting camera properties describe a *frustum* in which the rendering computation will be performed.
- **Scene** - Summarily, the scene is where the actual objects will be added in order to later be rendered. By default, objects are added at the origin of the origin (0,0,0) coordinates.
- **Renderer** - The renderer uses both the created Scene and Camera components to compute and project the scene into the canvas. Using WebGL enables the renderer to allow GPU-accelerated features to draw on the screen.

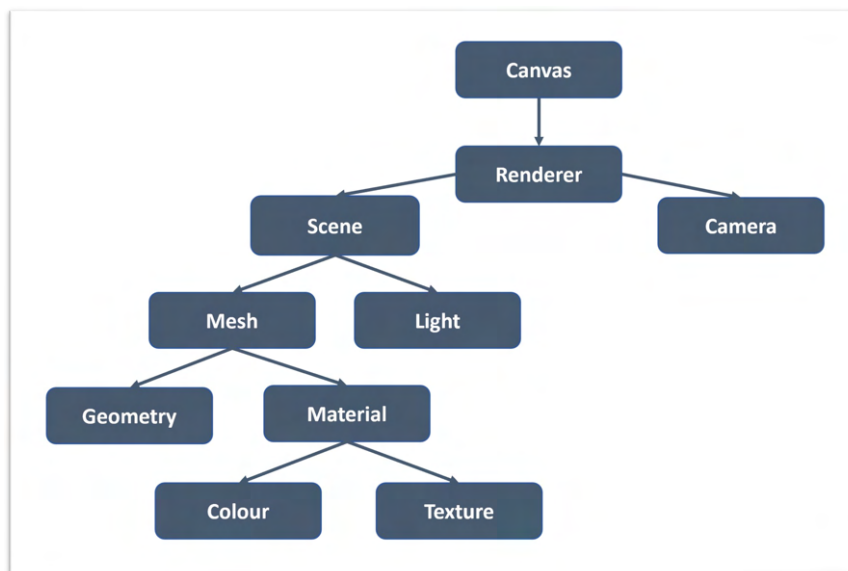


Figure 2.5: Tree representation of the Three.js structure [52].

The way in which developers can manipulate and control the scene requires at times the ability to leverage the Three.js's resources. Besides the brief overview given above there are many more details to take into consideration when conceiving the scene graph, for instance, objects added to the scene may be subject to different lighting. Moreover, the mentioned objects are actually referred to as *meshes* containing their own geometry and materials which in turn can have different textures (fig. 2.5).

The creation and research of interactive interfaces using the Three.js API allows for further development of the existing features and enables the exploration of different ideas for its possible utilization. One such example is its application on the development of e-learning interfaces due to the potential behind enhancing the learning process through

an immersive and accessible medium [52]. Additionally, this JavaScript 3D library can be extremely helpful for 3D data visualization on the Web [53]. The existing modules provided by Three.js can be of further help by allowing additional control over the data (e.g., OrbitControls, DragControls), thus enabling the manipulation of the orientation and position of elements. However, there is one underlying challenge: how to opt between the existing 3D file formats when bringing external 3D models?

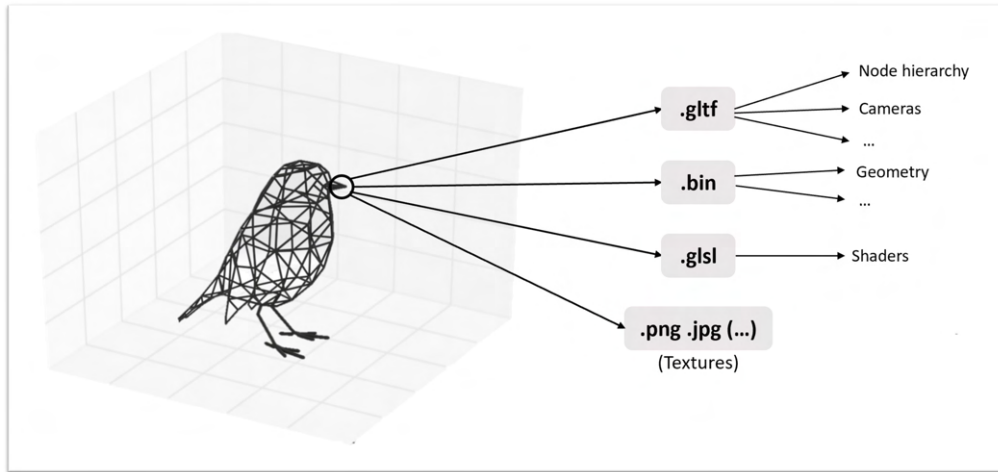


Figure 2.6: Gltf standard format structure as presented by Miao et al. [54].

There is an immense variety of possible 3D file formats (e.g., .obj, .fbx, .gltf, .glb, .stp) available to be shared on the Web. Nevertheless, especially when working on Three.js-based applications, there is a common popular format that stands out amongst the rest: the gLTF format. Perhaps not coincidental, the creators behind the making of the gLTF 3D models are also the ones who developed the WebGL API¹⁴. Therefore, the 3D library - Three.js - offers excellent compatibility with the .gltf with an optimized built-in submodule capable of loading any version of this format¹⁵. Moreover, as one would expect, .gltf stores information inherent to the 3D model characteristics such as its geometry, texture, and colors [54].

This format is highly flexible, enabling assets to be provided in JSON (.gltf) or binary (.glb), both suffering continuous improvements to maintain their ability to adjust to their use. It is also worth noting that a given gLTF asset may deliver multiple scenes, including all the components mentioned above (e.g., meshes, animations, cameras, lights).

The gLTF will be the initial basis for the integration of external three-dimensional models into the MotionNotes annotator. As a result, various factors must be taken into consideration, including the load it might place on the system. Fortunately, besides the general optimism due to the constant updates on the gLTF standard, studies show

¹⁴Khronos Group mentioned before is a prominent actor in the world 3D content creation behind the development of WebGL and the gLTF file format.

¹⁵<https://threejs.org/docs/#examples/en/loaders/GLTFLoader>.

that this specification allows for the efficient transmission and loading of 3D scenes and models by 3D applications [55].

2.3 Motion Tracking and Estimation

Motion and pose occur often in video-based footage, especially in the performing arts context, which is inherently connected to cultural heritage contents relevant to the goals of this thesis. Hence, a previous iteration of the MotionNotes system [56] tried to integrate pose estimation as a possible new feature in the annotation tool. The currently available annotation mechanisms enable users to enrich their multimedia content by personalizing the way they can add valuable information over the video’s time frames. Consequently, the resulting feedback from this initial testing of pose estimation functionalities revealed that identifying a person’s pose might open new possibilities for the way users interact with video, for instance, by aiding in the creation of new annotations.



Figure 2.7: Simulation of the detected keypoints and an actual posterior segmentation available in the COCO dataset.

2.3.1 Concepts

Motion tracking, in general, is a popular research area where numerous attempts at making the most of the hardware and software capabilities are constantly made to improve the performance of the existing/new algorithms. Moreover, even though there is potential in identifying objects’ motion to keep track of their movement, due to the nature of this project, the primary focus will be on human motion tracking and pose-estimation. It is common to refer to both as synonyms, however for the sake of clarity, when referring to pose estimation throughout this document, there is an implicit distinction, i.e., pose estimation usually relies on prior knowledge (e.g., previously classified datasets) to make its estimates whereas motion tracking, in general, might not need to do so.

Human pose estimation aims to identify the spatial location of a person’s body parts/joints,

also known as *keypoints*. In an initial phase, these joints must be located based on the different *keypoints* considered (e.g., head, elbows, knees, feet). Thus, using popular datasets containing these elements is a common step used in order to correctly determine their position, where the COCO¹⁶, MPII¹⁷, and CMU¹⁸ datasets are notable examples (fig. 2.7). Nevertheless, care must be taken when choosing the most suitable dataset since there usually are minor subtle differences between them, for instance, in the number of *keypoints* they provide (e.g., COCO considers 17 body parts, MPII considers 14 body parts). The obtained keypoint candidates are then used to try and create a correct estimation of that given image's human pose configuration.

Machine learning - more specifically, deep learning models - plays a vital role in this field of computer vision, having recently greatly outperformed previous traditional approaches in the accuracy behind correctly determining *keypoints* in pose estimation algorithms [57]. There are many details to assess when developing or expanding pose estimation implementations, however, the primary goal in this thesis' context is to integrate motion tracking applied to human motion using already developed tools. Even so, to take full advantage of these tools, it is crucial to grasp the inherent concepts of any of the proposed algorithms to understand the subsequent implications of using such tools. As a result, knowing the difference between single vs. multi-person pose estimation, top-down approach in detriment to a bottom-up approach, and the loss functions applied in the machine learning algorithms are some of the fundamental aspects to bear in mind. This latter factor is intrinsic to the machine learning algorithm's side of things where evaluating the results from a given loss function allows for a better understanding of a given prediction and enables the algorithm to learn and adjust (e.g., L2 ¹⁹ loss function).

2.3.2 Pose Estimation paradigms

Intuitively, multi-person pose estimation attempts to identify the pose of all the individuals in an image or video frame as opposed to single-person pose estimation, which only focuses on estimating pose for a single person in the same setting. Despite the advancements gained by shifting from traditional human pose algorithms to machine learning-based ones, there are still some challenges in accurately determining the existing *keypoints* [58]. Moreover, particularly for the multi-person variant, some of these challenges become heightened. For example, the varying scale/position and interaction between people may lead to an increased occlusion of body parts or simply a decrease in the overall visibility.

There is also an essential distinction to be made regarding the type of approach used for pose estimation: top-down and bottom-up. A top-down approach determines the

¹⁶<https://cocodataset.org/#explore>.

¹⁷<http://human-pose.mpi-inf.mpg.de/>.

¹⁸<http://dome.db.perception.cs.cmu.edu/>.

¹⁹This loss function, commonly known as the Mean Squared Error (MSE), measures the quadratic difference between the predicted and real values in the given dataset.

position of the individual instances in a given image/frame to compute a bounding-box around them in order to later identify their poses, usually from low to high resolutions. In this context, methods such as CPN [59] and HRNet [60] are noteworthy examples, aiming at solving problems such as dealing with difficult keypoints and maintaining high-resolution representations throughout the network, respectively [61]. In addition, top-down approaches achieve better accuracy when compared to bottom-up methods. However, they carry higher computational costs due to repeatedly performing single-person pose estimation for each detection, thus growing in complexity proportionally to the number of people in the image/frame.

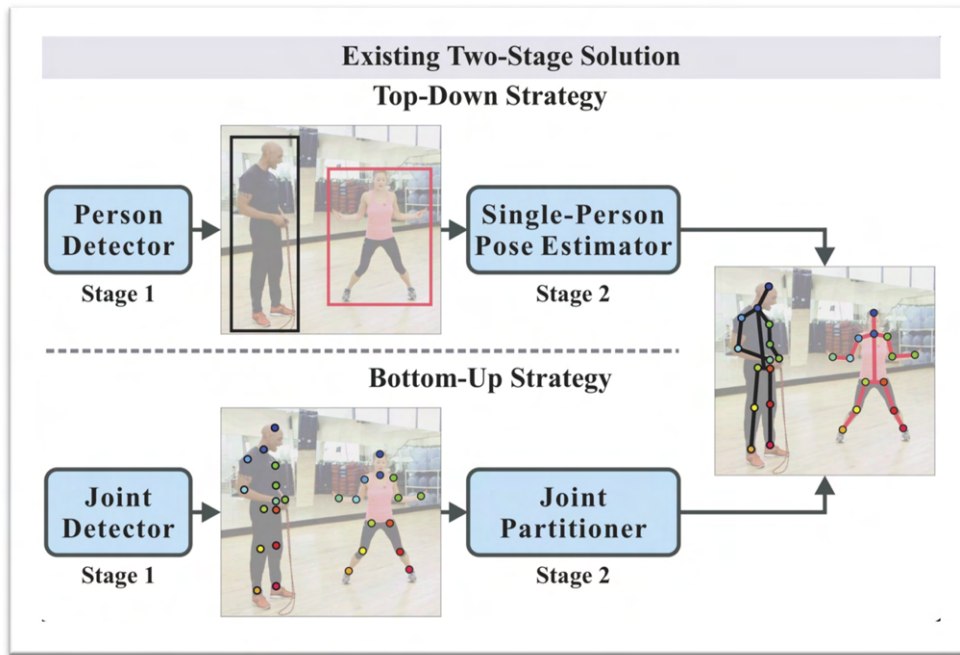


Figure 2.8: Visual representation of different approaches for multi-person pose estimation based on [62].

On the other hand, bottom-up approaches such as OpenPose [63] determine multi-person keypoints in the image directly, which are then assembled into full-body poses. Interestingly, OpenPose utilizes Part Affinity Fields (PAFs) to learn to link the body's joints with people in an image, which provided a boost in the overall accuracy of this type of approach [61, 64, 65]. Nevertheless, bottom-up methods are not without challenges, i.e., despite having better real-time performance, they usually face adversity in dealing with overlapping body parts as well as inferior accuracy in contrast to top-down approaches (fig. 2.8). Thus, one might prematurely infer that there is a division between approaches where top-down should be applied in single-person pose estimation, whereas a bottom-up approach ought to be used for multi-person algorithms. However, even though this might be true in some instances, it is not necessarily always the case. For example, S. Chang et al. [66] demonstrates the use of a top-down approach to employ single-person pose

estimation applied successively in order to detect each person in a crowded area. Recently, due to limitations in both bottom-up and top-down methods (e.g., needing additional person detectors, top-down, grouping keypoints heuristically after keypoint prediction - bottom-up), single-stage methods are now explored to surpass these constraints. X. Nie et al. [62] presents a valuable model using this paradigm to try and simplify the two-stage pipeline typical in both and lift the efficiency for multi-person pose estimation. However, despite the potential behind novel single-stage methods, they still fall behind the state-of-the-art bottom-up approaches in terms of performance [61].

2.3.3 Technologies and Applications

T. L. Muneo et al. [64] reported their findings regarding the behavior and characteristics of the different relevant pose estimation models ever since DeepPose was first created until recent times (fig. 2.9). Google introduced DeepPose [67] by presenting pose estimation as a DNN²⁰-based regression problem and is regarded as a reference for pose estimation progress. Interestingly, popular frameworks and libraries such as OpenPose use CNNs - a particular type of DNN - in their pose estimation algorithm. However, new modern approaches (e.g., HRNet, CPN) now use ResNet²¹ due to its ability to tackle the problem of vanishing gradients in the backpropagation algorithm of CNN architectures.

Models	Single / Multi-person	Top-down / Bottom-up	Dataset used	Loss function
DeepPose (2014)	Single Person	Top-down	FLIC, LSP	L2 Loss
ConvNet Pose (2015)	Single Person	Top-down	FLIC, MPII	L2 Loss
Convolutional Pose Machines (2016)	Single Person	Top-down	FLIC, LSP, MPII	L2 Loss
Stacked hourglass (2016)	Single Person	Bottom-up & Top-down	FLIC, MPII	L2 Loss
DeeperCut (2016)	Both Single and Multi-Person	Bottom-up	COCO, LSP, MPII	Cross-Entropy loss, L1 Loss
Human Pose Estimation with Iterative Error Feedback (2016)	Single Person	Top-down	LSP, MPII	L2 Loss
Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields (2017)	Multi-Person	Bottom-up	COCO, MPII	L2 Loss
Cascaded Pyramid Network for MP pose estimation (2018)	Multi-Person	Top-down	COCO	L2 Loss, L2
Simple Baselines for Human Pose Estimation and Tracking (2018)	Multi-Person	Top-down	COCO	L2 Loss
HRNet: Deep High-Resolution Representation Learning for Human Pose Estimation (2019)	Multi-Person	Bottom-up	COCO, MPII	L2 Loss
Cascade Feature Aggregation for Human Pose Estimation (2019)	Single Person	Top-down	MPII, LIP	L2 Loss
Human Pose Estimation for Real-World Crowded Scenarios (2019)	Multi-Person	Top-down	CrowdPose, JTA	L2 Loss
Distribution-Aware Coordinate Representation for Human Pose Estimation (2019)	Single Person	Top-down	COCO, MPII	L2 Loss

Figure 2.9: Common pose estimation models explored by T. L. Muneo et al. [64].

The Microsoft Kinect sensor [68] is a notable example of the resulting advancements in sensing technology that allows affordable depth sensors previously mentioned in the

²⁰A Deep Neural Network is a neural network with several layers between the input and outputs layers.

²¹Residual Neural Network.

context of 3D environments to become widely available. Moreover, despite its limitations in the integration of mobile devices, given a single depth image (RGB-D), it can detect the 3D position candidates in order to create a human skeleton representation, thus becoming an initial notable contributor for pose estimation systems. Nevertheless, more powerful techniques using machine learning models such as those presented before have allowed for a vast amount of possibilities in how they can be employed.

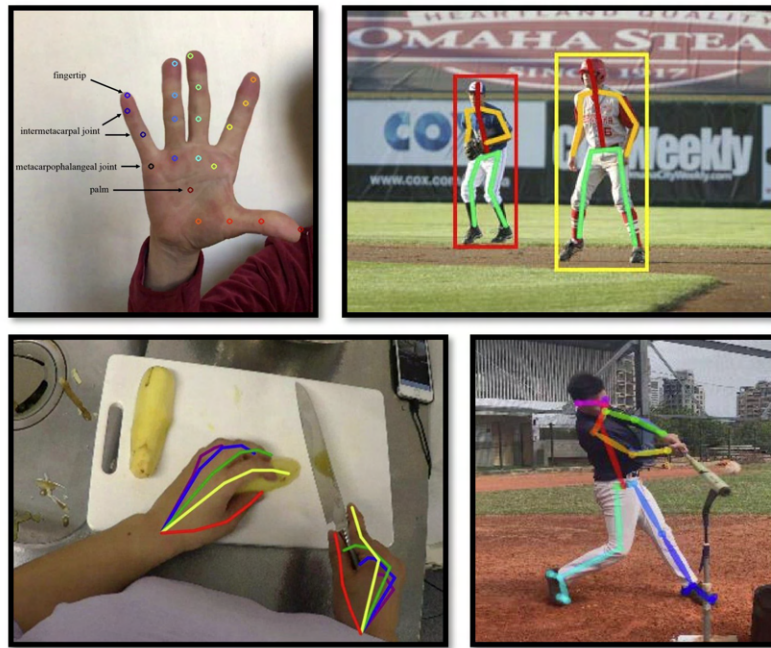


Figure 2.10: Examples of research using PoseNet and OpenPose [65, 69–72].

The OpenPose [63] model proposed by Carnegie Mellon University (CMU) is one such example introducing a bottom-up approach capable of real-time detection and pose estimation with considerable performance as well as improved accuracy over previous bottom-up methods. Moreover, the computation can be performed over standard RGB images without requiring depth data. Even though this technology is relatively recent, numerous studies and applications have provided insight into the potential behind its use. For example, Y.-C. Li et al. [65] conducted an experiment using OpenPose as a real-time multi-person detection system to evaluate baseball players' swings and help them analyze and correct their poses to improve their technique. In other areas, OpenPose can be used as an integrated component of a complex system, for instance, to compute keypoint extraction and aid in recognizing specific actions [69].

Another popular pose estimation model of significant importance is PoseNet [73].

Google introduced PoseNet aiming at the efficient use of pose estimation for mobile devices. However, this top-down approach, when applied to multi-person pose estimation, decreases in performance in proportion to the number of people as one could expect for this type of approach [72]. Even so, various projects rely on PoseNet due to its respectable performance and accuracy [70, 71]. The presented solutions using OpenPose and PoseNet for efficient pose estimation are undoubtedly a central focus for development and testing in this thesis’s section. However, care must be taken when considering their different characteristics. For instance, PoseNet was conceived for lightweight devices, thus showing potential in browser-based environments, whereas OpenPose is more accurate and relies on the GPU to perform pose estimation. Considering the hardware and software specification inherent to the MotionNotes annotation tool evaluating their behavior is vital in order to better understand their possible future integration. Furthermore, it is also feasible to explore viable alternatives due to the ongoing development of pose estimation mechanisms.

Expanding pose estimation beyond simply computing the skeletal image of a given person in an image or video frame relies upon actually keeping track of each person in a continuous setting (e.g., video) when more than one person is present. As a result, there is ongoing research that aims at accurately identifying people in successive video frames across different professional areas [74, 75], yet most either rely upon computationally intensive machine learning algorithms or additional equipment (e.g., sensors, smartphones) to fully work. Interestingly, the OpenPose framework even tried to expand the pose estimation algorithm by including person tracking through the use of deep learning techniques. However, due to the complexity inherent to doing so, given all the possible variables (e.g., distance to the camera, environment conditions, clothing), the number of people capable of computing pose estimation on is restricted to one per frame when applying person tracking as well. Consequently, in this context, there is an intrinsic limitation to person tracking when there are multiple people in a video.

It is relevant to reinforce that attempting to implement some of the existing general people tracking methods would add another complex layer to the current system. However, pose estimation makes it possible to track one’s coordinates throughout time and thus deduce where each person is at a given timestamp. Additionally, due to people’s movements during the course of a video (e.g., people overlapping), depth data can be critical to precisely retain positional information. For instance, F. Fang et al. [76] describes the use of cameras with depth sensors applied to people tracking in a video setting. Even so, in this project’s context, videos, more specifically, monocular videos, will contain no such inherent information. Still, recent developments in the computer vision field make inferring image depth accessible where *MiDas* [77] and *LeRes* [78] are notable examples which will be further explored in chapter 3.

DESIGN AND IMPLEMENTATION

As previously mentioned in Chapter 1 of this thesis, the MotionNotes annotation system has been further developed to now include 3D elements as well as pose estimation features. It is essential to highlight that the implementation details described here are always mentioned in the context of an existing intricate structure and must always consider its inherent characteristics (e.g., MotionNotes is a browser-based application).

Thus, this chapter discusses the approaches and decisions made throughout the development process, detailing the most relevant choices and techniques inherent to its implementation.

3.1 Design

The system now integrates two distinct components: 3D functionalities and pose estimation over multimedia content. Chapter 2 explored how users can easily acquire and use 3D models (e.g., 3D modeling, real-world extraction, existing libraries) and their application in different fields of work. Integrating these tridimensional objects into this system requires mechanisms capable of actions such as loading and displaying models in a virtual environment. Given the web-based nature of MotionNotes, a suitable underlying library capable of supporting the rendering of 3D graphics was needed. As previously referred, WebGL remains a prevalent library in web development, allowing the use of 2D and 3D graphics across any browser. As a result, Three.js was explored as a potential candidate framework due to it being able to use WebGL while hiding some of the unessential low-level details. Moreover, other factors were considered (e.g., lightweight framework, large community, good documentation available), ultimately making it the preferred solution.

Even though the system should preliminarily allow users to visualize and interact with 3D models effortlessly, the main objective is to utilize them as annotations similar to other existing ones, such as text or drawing annotations. Consequently, the internal format of these objects can impact the system's overall performance since, for instance, multiple 3D annotations can coexist at a given time. Furthermore, considering possible

future developments in the MotionNotes system, choosing the right format is key. Thus, the *gLTF* format was selected. Besides permitting efficient transmission and loading of assets, it is highly flexible, and its compatibility with WebGL (fig. 3.1) and subsequently Three.js were the dominant deciding factors.

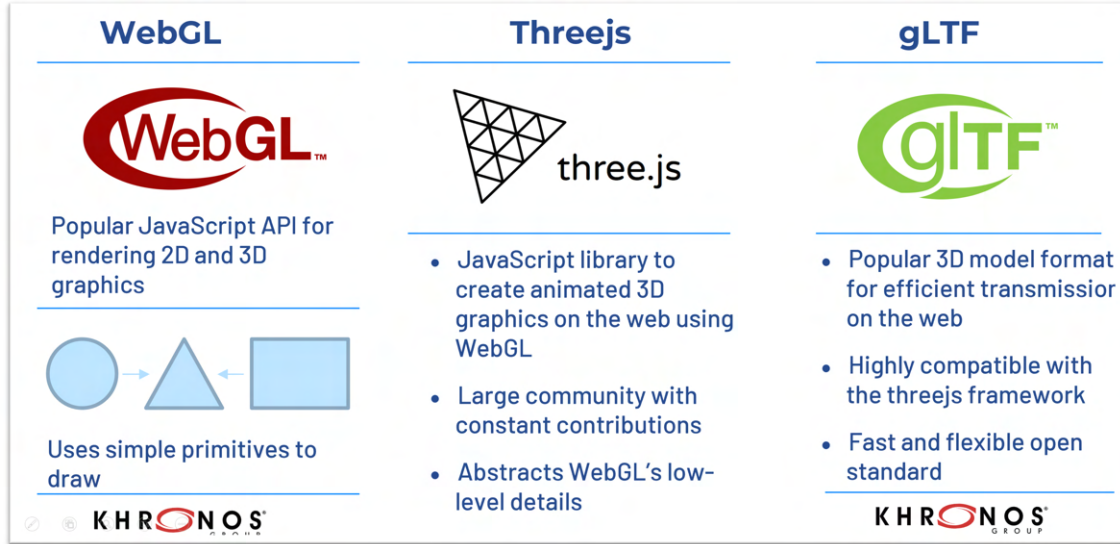


Figure 3.1: WebGL, Three.js and gLTF brief summary and their connection.

Once integrated in the system, care must be taken in how they are presented in the system's interface before using them as annotation mechanisms. As such, besides the inherent factors of tridimensional visualization mentioned ahead, it is essential that 3D annotations are in most ways identical to traditional MotionNotes annotation types. Therefore, some existing behaviors, such as possessing an adjustable timestamp (fig. 3.3, labels C and D) on the annotation track, and allowing the annotation to be re-positioned on the screen by clicking on the top right trigger (fig. 3.3, label B), are implemented to strive towards a seamless integration of the new 3D functionalities into an existing system. Even though there are several possible approaches to obtain 3D models, upon importing these objects into the system and before using them as personalized annotations, users should first be capable of visualizing their models and performing simple interactions (e.g., object rotation). This is especially relevant since the devices used to interact with MotionNotes contain a 2D surface for visualization, thus naturally inhibiting the possibility of viewing hidden surfaces of 3D objects (e.g., the back of a cube).

For the purposes of this thesis, two different components allow the pre-visualization of tridimensional objects: a *personal area* for importing/adding 3D annotations and a *3D Models Provider* containing existing models from our project's partner Arctur¹. Once again, the interaction mechanisms are meant to be similar in both, with the key difference

¹The models made available here are also present at <https://weave-3dviewer.com>. Every item is directly related to the theme of the WEAVE project, thereby having some relation to European cultural heritage.

of being able to import and delete models in the user's personal 3D window and not the 3D Weaver Models, which will be further explored in the upcoming section.

Besides the straightforward interactions such as moving and scaling a 3D model, whether on the visualization window (fig. 3.5) or on the video itself, the possibility of using basic 360° environments for object visualization was chosen as a feature as well. This option was considered due to the possible need for artists², for instance, to picture how an object would blend in with a given environment. In addition, future developments can contemplate using entirely virtual 360° environments and 360° videos, thereby motivating the implementation of this functionality.

The idea of determining a person's pose or posture can, in a general sense, be helpful since it allows for a better understanding of someone's positioning in any given setting. Furthermore, it is often essential in areas such as professional sports, where analyzing the quality of an athlete's movements is directly related to their performance. In this project's context, it becomes especially relevant due to the ongoing relationship the tool has with the performing arts world³. In fact, the discussion of how annotation mechanisms over multimedia content can help dance specialists and dancers, more specifically using 3D elements, led to the publication of the paper further described in Chapter 4. Consequently, the following developmental step focused on the automatic identification of a person's pose, which is often necessary in the (traditional) dance context since it facilitates the visualization of angles between a person's joints as well as determining specific movement patterns.

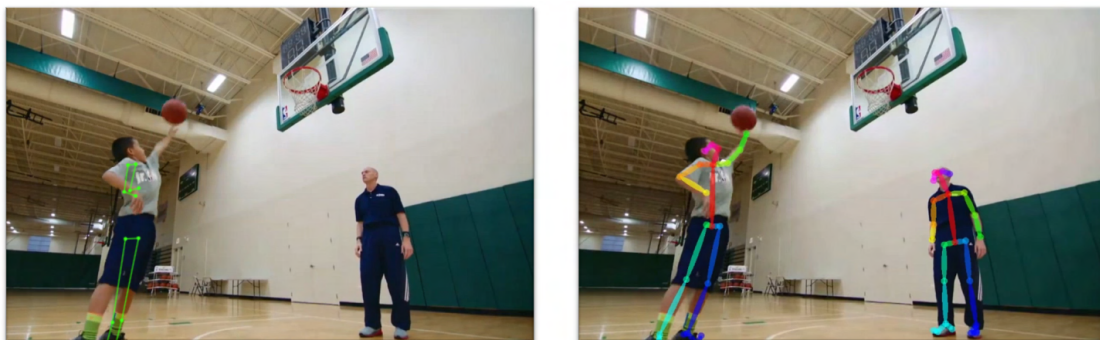


Figure 3.2: Example of PoseNet (left) and OpenPose (right) pose estimation inference.

The two previously mentioned approaches for pose estimation could mainly be either bottom-up or top-down, each with their respective implications. A prototype of the latter one exists in the MotionNotes tool using the PoseNet top-down model developed in a prior version of the system. Besides other factors such as decreasing performance in proportion to the number of people in a video - a common obstacle to this approach

²This was corroborated by the workshop session conducted with traditional dance experts to receive preliminary feedback on the 3D features.

³The PédeXumbo Portuguese association is one of the WEAVE's project partners in close contact with the annotation tool: <http://pedexumbo.com/>.

- in comparison to OpenPose, PoseNet revealed to be much less reliable in accurately identifying keypoints throughout a video (fig. 3.2).

This thesis also intends to explore possible expansions over the basic pose estimation functionalities (e.g., dynamic annotations, gesture recognition) where stable keypoint identification is paramount due to the need for consistent body part tracking across successive video frames. In addition, the MotionNotes system is expected to have its server-side hardware upgraded with new graphics cards, thus mitigating possibly long computation times inherent to the OpenPose algorithm. Considering all of the mentioned factors, the OpenPose pose estimation library is thus integrated into the annotation system.

Since the execution of OpenPose-related commands falls upon the server's responsibility, users are given the option of selecting between the existing lightweight PoseNet model to be executed or the new pose estimation library. Despite being further discussed in the next section, it is important to mention that OpenPose processes the video and then outputs different items upon completion. In this case, the most relevant ones used are *json* files containing pose data and the video given as input with keypoint information in each frame. Users may find it necessary to have only the keypoint information on the video, which is directly embedded by the OpenPose library itself. This option may allow for smoother pose visualization by having the keypoints embedded in the video as opposed to relying on the browser to illustrate the identified skeletal image(s). As a result, there are two options for users to utilize pose estimation in this recent setting:

- **Static View Mode** - Selecting this mode will allow the user to visualize the computed pose estimation data directly. Even though additional features (e.g., dynamic annotations, gesture recognition) are restricted to the default mode, future plans may aim to make these output videos downloadable.
- **Default Mode** - Upon receiving keypoint information, it keeps track of its data relative to the ongoing frame. In this mode, other features are available in the Motion UI tab (fig. 3.19).

Due to the nature of pose estimation algorithms, the primary concern given an image, multiple images, or a video is to identify people despite possible adverse ambient conditions (e.g., lighting, image or frame quality, rapid movements) and their respective poses. Therefore, even though some algorithms attempt to use temporal information to keep identifying people throughout a video, current pose estimation techniques struggle to actually track and distinguish humans between themselves in successive video frames. As a result, if a person is attributed the X identifier (e.g., Person 0) in a given video frame, there is no guarantee that will be the case in the following video image. Evidently, it is natural to happen in some situations since, for instance, the number of people may vary, or different individuals may switch in and out of the recording. However, when the environment is somewhat stable, meaning that for some duration of time, the identified

people move along an identifiable path (e.g., left to right), then it is possible to infer that person X should not change identifier during that period. Naturally, it is liable to fail when people and the environment change abruptly in between frames. However, it should be consistent as long as the environment allows it.

Interestingly, OpenPose tried to expand their implementation to successfully id people throughout the video. However, some restrictions are implemented due to the complexity and considerable processing time associated with implementing such a solution using machine learning techniques. In this particular case, OpenPose only allows one person per frame to be tracked and have pose estimation computed on.

There are several possible routes to achieve person tracking (e.g., facial recognition) that could be used to tackle this problem, yet this thesis aims to implement and study the relevancy of pose estimation in the annotation tool. As a result, integrating another possibly computationally demanding algorithm for person tracking in addition to pose estimation is impractical. Nevertheless, by having the pose estimation information for each video frame, it is possible to keep track of the identified people by looking at their positional changes in consecutive frames. Expanding the current pose estimation can thus rely on its own keypoint data to identify people consistently in successive video images.

3.2 Implementation

The design choices that led to the integration of this system’s novel features certainly played a part in how the major implementation details are currently in place. Notably, some of the existing functionalities in the annotation tool also influenced the creation of these new components. As a result, this section will initially provide an overview of the MotionNotes system as a whole prior to this project’s development.

Afterwards, the focus is divided between the tool’s primary integrations: 3D annotations and pose estimation. Here, an in-depth discussion of the individual implementation details is presented, along with the manner in which the respective technologies and annotation mechanisms co-exist within the system.

3.2.1 System Overview

The MotionNotes tool has been a target of multiple incremental development phases. As previously mentioned, the system already contained various features allowing users with the capability of annotating video in different ways using: text notes, ink strokes, audio, user-configured marks, and URL hyperlinking capable of supporting work done at both the professional and amateur level (fig. 2.3). Due to the nature of the international project where this system is being developed, MotionNotes was designed to integrate various technologies, multimedia elements, multimodal interaction, AI, and 3D modeling, especially with regards to the cultural heritage and performing arts-education contexts [56, 79].

Hence, it combines several of the functionalities mentioned in other annotation systems (e.g., [4, 5, 7, 19–21]) yet combining other annotation types and modes.

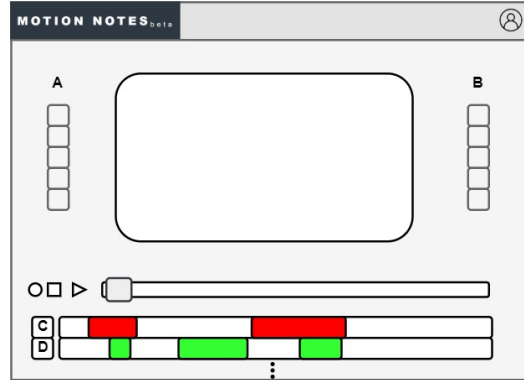


Figure 3.3: Initial prototype of the MotionNotes annotation tool.

Figure 3.3 provides a simple and visual representation of the MotionNotes⁴ page. Menus are available in the header section at the upper part of the page for settings such as customizing the page’s appearance or selecting and loading videos. On the left (**A**) the different input modalities can be selected while their properties are accessible on the opposite side (**B**); for example, upon selecting an ink stroke annotation, users can change the line’s color according to their representation (e.g., RGB, HSL). Intuitively, the middle area displays the previously recorded or current video recording. Besides the most common tasks, i.e., pausing, playing, or stopping a video, actions such as selecting and placing annotations are possible here due to the canvas layer placed over this section. Consequently, a user can draw on top of the video, and MotionNotes is able to save information about the region where the ink strokes were drawn and the respective timestamp of when they were created. Lastly, the annotation tracks relative to each annotation type (e.g., **C** and **D**) are positioned at the bottom of the page. The red and green slots represent the timestamps of the simulated annotation types **C** and **D** where further editing and customization are possible including changing their start time and duration.

On a more technical level, MotionNotes is a single-page application implemented with a client-server architecture using HMTL5, CSS3 and Javascript (ES6) on the client-side, whereas the server components are built upon the Node.js framework (fig. 3.4). Since the MotionNotes tool is Web-based, accessibility becomes a noteworthy characteristic since users are thus able to interact with the system through a wide range of devices with internet access. As a result, further interactions become natural such as extending the use of traditional interfaces (e.g., computer’s keyboard) to a more ubiquitous human computer interaction, for instance, by using a smartphone to annotate directly through touch thereby creating a drawing annotation. The way in which the client-side communicates with the server is quite straightforward regarding its implementation: the client

⁴The current version of MotionNotes can be found at <https://motion-notes.di.fct.unl.pt/>

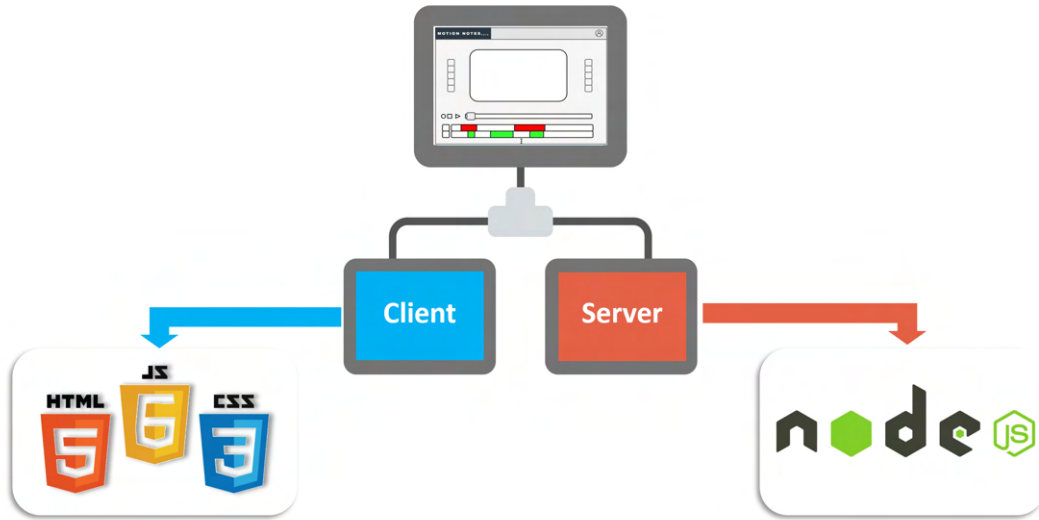


Figure 3.4: Simple overview of the client-server architecture implemented on Motion-Notes

emits a request to be processed on the server-side, which passes through an intermediate layer responsible for building said request and sending it to the node front which finally processes the result and returns back to the client. For instance, in the context of this system, the `/video` endpoint (GET) is used when a user either imports or loads an available video. This is materialized by creating an *Ajax XMLHttpRequest*⁵ which sends the request with its respective arguments (e.g., video file name) and waits for the server's response, in this case, the video chunks, in order to display the selected video.

Moreover, in the context of the annotation tool, all traditional annotation types (e.g., drawing, text, and speech annotations) have a specified position on the canvas and timestamp - temporal duration in the video - which the user establishes upon adding it to the screen and directly visible on the annotation tracks. Consequently, this information must be stored in order to permit persistent annotation data for the purpose of allowing to user to incrementally work on the system without losing prior developments. Intuitively, a more exhaustive look at how this works presupposes the existence of endpoints for saving and obtaining each previously added annotation (e.g., GET and POST at the `/annotation` endpoint). Naturally, this behavior along a respective annotation track now extends to 3D annotations to allow all annotation types to be coherent with each other.

Summarily, in the case of 3D annotations, the communication between the client and server fronts mainly occurs to upload or retrieve models for future visualization on the browser's page. The models are then loaded using the Three.js and displayed on the 3D model visualizer. On the other hand, pose estimation endpoints trigger actions such as obtaining a video with embedded keypoint data or simply a JSON file containing explicit

⁵This Javascript built-in browser object allows the HTTP requests to be made to transfer data between the web browser and the web server.

keypoint information. After that, since there is knowledge regarding the spatial and temporal location of the identified people’s body parts, the pose estimation features can be utilized by the user (e.g., dynamic annotations and skeletal view on Motion UI).

Type	Endpoint	Description
GET	/models	Gets the thumbnail for each of the user’s 3D models
GET	/load3dModel	Gets the selected 3D model
GET	/getAssets	Gets the thumbnails from the 3D Weaver dataset
GET	/getAssetModel	Gets the selected 3D model from the Weaver dataset
GET	/motion	Gets the PoseNet’s saved keypoint data
GET	/getOpenPose	Gets the OpenPose’s enriched* keypoint data
GET	/getOpenPoseVideo	Gets the automatically generated OpenPose’s video
POST	/motion	Sends the PoseNet generated keypoint data
POST	/user3dModel	Sends the imported 3D model
DELETE	/model	Removes the selected model’s data

Table 3.1: API Endpoints

Evidently, other existing endpoints were used and modified throughout the implementation of this project’s features, such as the */motion* endpoint in the table below focused on the previous iteration of the PoseNet pose estimation framework. Another noteworthy example is the endpoint to delete the user’s video(s) */videos* which, now, in addition to deleting the selected video and respective annotation file containing the timestamps for each created annotation, also removes the loaded 3d models and pose estimation files (e.g., keypoint information, OpenPose automatic video). Table 3.1 summarizes the more specific endpoints used and implemented in this thesis.

3.2.2 3D Model Visualization

The early research and development of functionalities aimed at implementing effective ways to insert 3D models in MotionNotes considering the different formats widely available (e.g., *.gltf*, *.obj*, *.glb*, *.step*), their intrinsic characteristics, and bearing in mind details such as whether the model’s textures were embedded or separate from the object. Throughout this process, a variety of tools and concepts related to the possibilities of 3D integration culminated in the use of the *Three.js* library for the browser-related features in order to display and work with 3D models.

Initially, the goal consisted of creating an interface for users to be able to import and visualize tridimensional objects prior to actually adding them to the video scene as annotation elements. Figure 3.5 displays a visual representation of the application’s window where those and other actions are possible. Immediately at the top of interface, there is a slider element containing the thumbnails for all the previously uploaded 3D models. The thumbnail images serve as visual cues for the user to know which model will be displayed upon selection. Below the 3D visualization area at the center of the interface, there are four buttons responsible for: adding the 3D model as an annotation,

visualizing it on the 3D Models Manager window, deleting it, and uploading the model to the server (left to right, respectively).

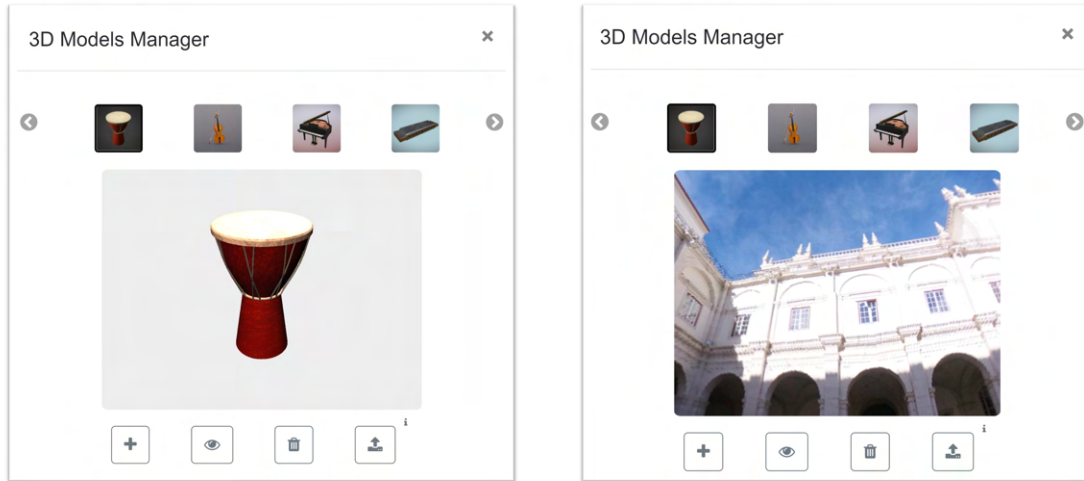


Figure 3.5: Interface for the 3D model visualizer containing neutral (left) and 360° (right) interactive backgrounds.

Intuitively, prior to the initial loading of the interface described above, elements such as the (empty) slider, the central canvas, and the four buttons are created statically to make up the structure that supports functionalities within the 3D Models Manager window. However, following that, two key operations are executed to have a functional UI: filling the slider element with all the thumbnails that represent existing 3D models - previously uploaded by the user - and triggering the setup of the Three.js environment for 3D object visualization. While the first resorts to a simple call to the server to receive the models' image data, the latter creates two initial instances of the Three.js virtual environment for both the user's personal 3D visualization area and the *3D Models Provider* containing existing models from our project's partner Arctur (fig. 3.8). The following critical elements are thus instantiated:

- **Camera** - Describes the frustum dimensions (fig. 3.6) where the scene will be rendered.
- **Scene** - Defines the elements (e.g., objects and lighting) to rendered.
- **Renderer** - Responsible for displaying the scene onto the canvas HTML element.
- **OrbitControls Plugin** - Allows the camera to controlled (e.g., zoom in and out).
- **Cube Map** - Sets up the surrounding environment for the selected 3D model.

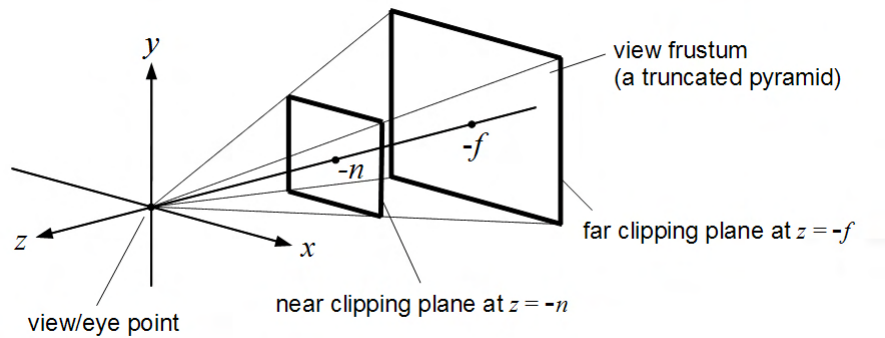


Figure 3.6: Illustration of the frustum view in eye coordinates by Martin Kraus.

On closer inspection, the camera uses perspective projection due to it being capable of mimicking natural human vision. Consequently, the FoV (field of view), aspect ratio, and frustum dimensions are the necessary parameters used to define the camera's base structure. In this context, despite the possibility of multiple scenes existing, only one scene is necessary in order to hold the visualization data: the selected 3D model, lighting conditions, and surrounding environment. The latter comprises six images representing all sides of a cube within which the 3D object will be placed, creating the enveloping cube map scenario, neutral by default (fig. 3.5). This elements are then displayed on the browser using the WebGL renderer to process the global scene and respective camera. Finally, after selecting one of the available 3D models, the system adds it to the Three.js scene to be directly observed.

In short, to upload a tridimensional object for subsequent use in their personal area, users must choose a valid file format either in the form of a *.gltf* or *.glb* and a thumbnail image representing the model along with possible additional data (e.g., textures). For example, Sketchfab⁶ is a widely used option for users to obtain existing 3D models across a vast genre of categories which also supports these popular formats. Even so, the internal format in the MotionNotes system is glTF by default due to its previously referred compatibility with the Three.js framework and efficiency regarding lightweight transmission within the web environment. Therefore, other formats are pre-processed after an upload is completed in order to maintain consistency within the server.

For models to be available in the user's personal area, the upload must contain a zip file containing the model's tridimensional data plus a thumbnail image. Afterwards, enabling the selected model's visualization requires user's to either double-click on the object's respective thumbnail image or trigger the view action (fig. 3.5 - eye button). Consequently, the selected object is added to the existing Three.js tridimensional area that was setup earlier on. In this case, that space is comprised by a simple Cube Map environment containing six grey images that enclose the object in this immersing neutral

⁶<https://sketchfab.com/>

background. The subsequent interactions are then possible through the use of the OrbitControls plugin, allowing users the illusion of controlling the selected object. This is achieved by altering the positioning of the camera when actions such as zooming in and out are applied (e.g., using a touchscreen, mouse, or touchpad). For instance, in this case, the camera simply moves closer and away from the object respectively to display this affect. This behavior is especially relevant when interacting with large models, as is often the case with the Arctur's Model Manager 3D objects since moving the camera relative to the object is significantly more efficient than having to re-render a potentially enormous amount of points that make up such a model. Intuitively, this approach also avoids conflicts, such as users trying to position the object outside the Cube Map's volume. However, there is still a problem regarding the initial size of the imported 3D model since it can have an arbitrarily small or large volume possibly hindering visualization (e.g., initial size exceeds the Cube Map's dimensions).

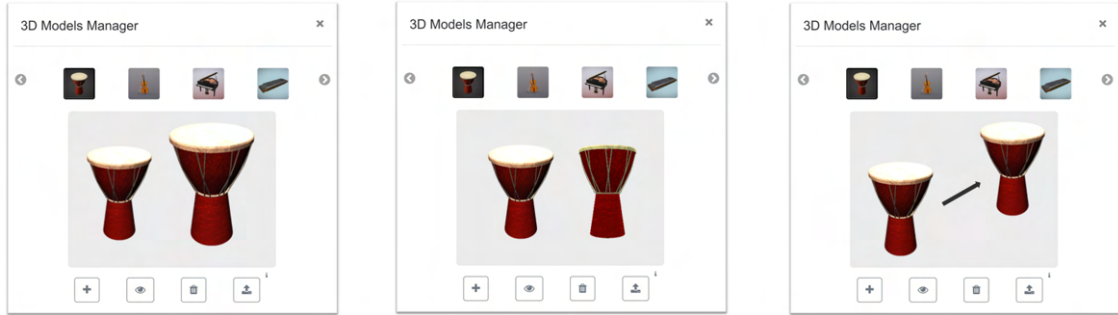


Figure 3.7: Simple interactions over 3D elements on the 3D Model Manager interface.

Adjusting the initial size of the retrieved model is thus achieved by resizing the model to fit the 2D canvas dimensions properly (fig. 3.5 - central area). Evidently, except for the particular case of the model exceeding the dimensions of the surrounding Cube Map, users can always zoom in and out to adapt the camera proximity to the model regardless of whether it is excessively tiny or large. However, automatically resizing the 3D object allows users to visualize and interact with the model directly, which greatly simplifies their initial interaction with the interface. Therefore, re-calculating the scale factor for the new model's dimensions, for instance, considering it exceeds the proper size for direct visualization can be computed as: $scaleFactor = \alpha / \max(Dx, Dy, Dz)$. In which α is a standard constant derived from empirical observations over the canvas dimensions and Dx , Dy , Dz represent the length of the 3D object along the axis x , y and z respectively. In this case, these values are obtained using a native Three.js function that captures the object's bounding box when first loading the model on the target canvas.

Upon enabling the model to be displayed on the viewer as a result of uploading and selecting its respective thumbnail, there are three main interactions done over the canvas. Intuitively, users can expect to be able to rotate, translate and scale their selected model even though behind the scenes these actions are abstracted through the camera's

movement over a neutral background. Figure 3.7 presents a summary of these possible interactions using a musical instrument inserted into this standard interface - 3D Models Manager. Similarly, these same interactions can also be conducted over a similar window with some visible differences further discussed ahead. Due to the nature of the international project where this thesis is inserted, one of the goals that was established among the existing parties was having viable connections between the different developed tools. While some fall outside the scope of this thesis, one particularly relevant one is deeply related to the concepts and developments conducted over the 3D components: Arctur's 3D Weaver⁷.

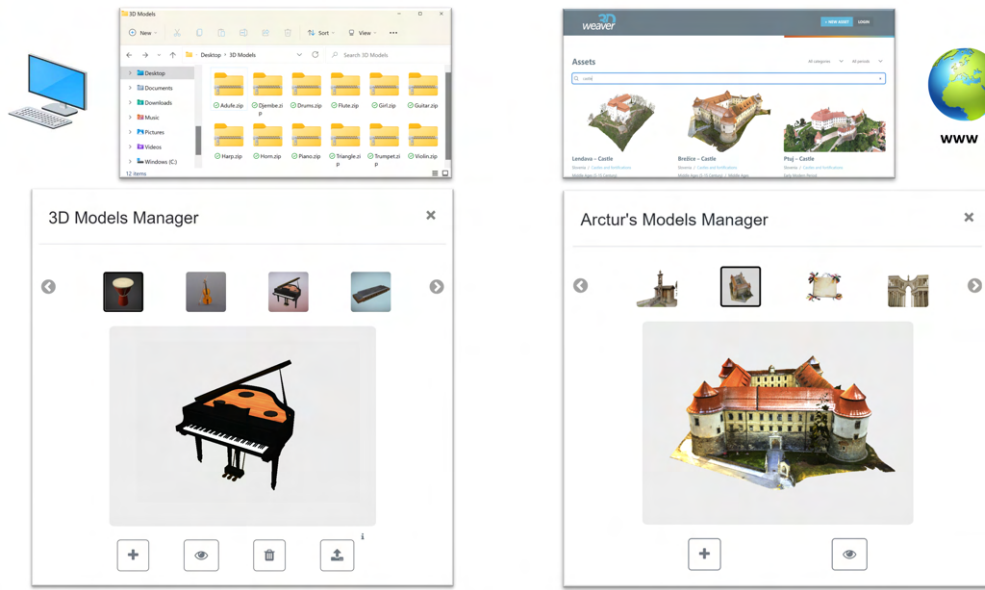


Figure 3.8: Interfaces for 3D model usage: personal (left) and public 3D models (right).

Identically to the explored default interface where users can upload and visualize 3D models, the *Arctur's Model Manager* complementary window displays a structure similar to the previously discussed one. However, the key distinction lies in the model's origin. While the *3D Models Manager* requires users to submit their models, in this interface, users can choose from a variety of additional models, which are publicly available through the Arctur's API (fig. 3.8). In this case, the model's thumbnails displayed in the slider element are retrieved through two successive GET calls that fetch the 2D images from the existing 3D objects on the 3D Weaver system. Afterwards, activating the model's visualization on the central canvas triggers another API call that transmits the 3D model in the glTF format back to the client. Consequently, since these tridimensional items are external, actions such as importing and deleting 3D models become unnecessary, thereby justifying the absence of their respective buttons in this interface.

Regardless of whether the models are selected from previously existing models or

⁷<https://weave-3dviewer.com/>

directly uploaded by the user, the annotation work over the video may begin upon properly loading and visualizing a given 3D model. For that purpose, it becomes essential to have at least three visualization worlds within the Three.js framework: one for each pre-visualization interface and another to display the 3D objects over the video. However, before diving further into the annotation functionalities, it is first necessary to explore how the models are converted inside the server, as well as the importance of further visualization features.

3.2.2.1 3D Formats and Environment

Initially, the *glTF* format was used as a base format with the purpose of using a popular format that could power the needed functionalities whilst supporting the structure of the tridimensional rendering performed by Three.js and subsequently WebGL. The previous subsection approached how 3D models can be interacted with and visualized. Nevertheless, the fact is that, at most, only the select model is visible on either pre-visualization window. Moreover, when a different model is selected for visualization, the previous one is removed from that space before adding the new one to the graph scene. As a result, the possibility of placing too much weight on the system can only derive from utilizing significantly large models as opposed to simply rendering many models at once. However, the latter should still be taken into consideration since users can annotate using multiple 3D models at a given moment. Thus, while having models with notable dimensions can be controlled by establishing a cap over the size of the uploaded 3D models, for this second one, the format used can impact the system's performance.

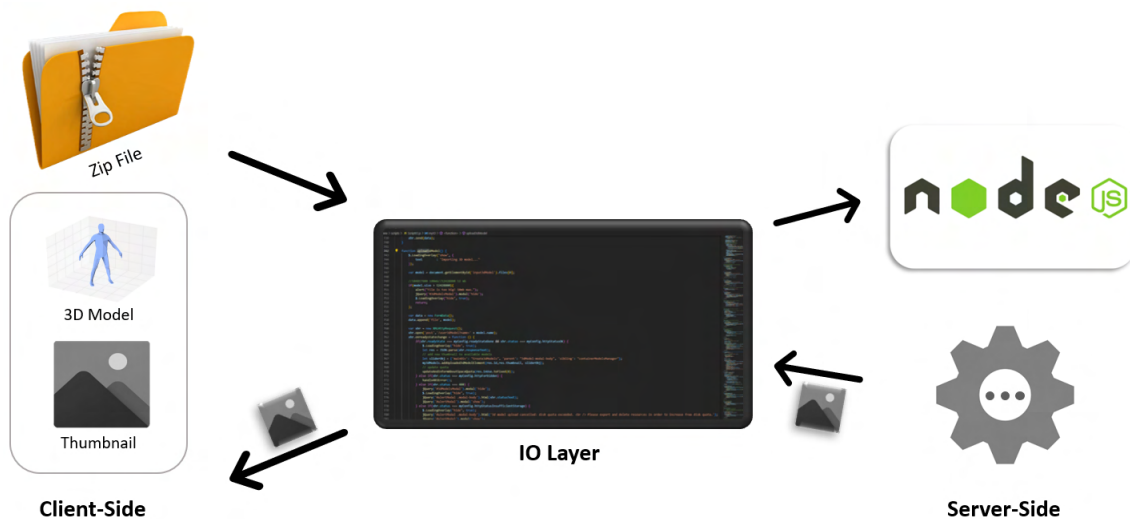


Figure 3.9: Uploading 3D models operations execution flow.

Hence, users should have the possibility of adding models with their preferred format (e.g., *.gltf* and *.glb*) as long as they are within the range of the accepted formats in Motion-Notes. Thereafter, the models are uniformly converted to the *gLTF* format to guarantee

consistency inside the system. Thus, the correct conversion of models requires users to provide an acceptable input containing a zip file that includes both the 3D model as well as its respective thumbnail image. However, the server side must then be responsible for identifying all cases that pose a problem before proceeding to convert the model. In this case, the default transformation outputs the provided 3D object into the glTF embedded format, meaning that the textures applied to the object are directly integrated into the .gltf file. Consequently, even though the input may already contain a glTF model, if the textures are placed in a separate folder, the system will transform the object in order to integrate the textures within the original .gltf file. Moreover, this particular example also represents a special case since it is essential to handle the distinction between the textures and the thumbnail received, which often come in the form of an identical format (e.g., .jpg or .png). Therefore, with the help of packages such as the *gltf-pipeline* available at the npm⁸ registry, the following two cases are covered:

- **Wrong Format** - The application displays an error message to indicate that either there was an incorrect number of thumbnail images (e.g., 0 or 2) or the model provided contains the wrong format.
- **Model successfully converted** - The items inserted in the zip file are acceptable triggering the conversion process by calling forward an external process to be executed with the following structure: `<package_name> -i <input_path> -o <output_path>`, in which, the *package_name* represents the npm package used to convert the model and the *input_path* and *output_path* define the given 3D model's input path as well as where to place the outputted file, respectively.

Afterwards, some additional actions are executed in the background to organize the stored models, such as deleting the external textures and preparing the thumbnail image to return back to the client. Figure 3.9 portrays a simple overview regarding this process leading up to filling the slider element with the model's image, which enables the model to be ready for visualization and subsequent use in the annotation work. In the next system access, the slider will already contain that thumbnail element obtained in the initial setup of the interface by executing a GET call to the */models* endpoint.

Even though the present version of the MotionNotes system currently supports these functionalities, future work may contemplate compressing the 3D models, which is important when working with multiple or considerable large 3D objects just as previously mentioned. The most suitable library in this context will in all probability integrate Draco compression⁹ due to its widespread use as well as efficiency in regards to improving the storage and transmission of 3D graphics. Another possible route of further development which will be discussed in a later chapter is expanding the integration of traditional monocular recordings to 360 videos given the potential behind using 360° in

⁸<https://www.npmjs.com/>

⁹<https://google.github.io/draco/>

both images and video. This reasoning partially motivated the creation of mechanisms capable of altering the background of the visualization window from a neutral setting to an interactive panoramic background (fig. 3.5). The 3D objects are thus loaded into a virtual environment surrounded by a specified Cube Map. In this context, Cube Maps are defined as a collection of six images that represent the bounding box enclosing the selected 3D element. By default, the Cube Map used contains only grey images which are loaded as textures to be rendered by the Three.js framework.

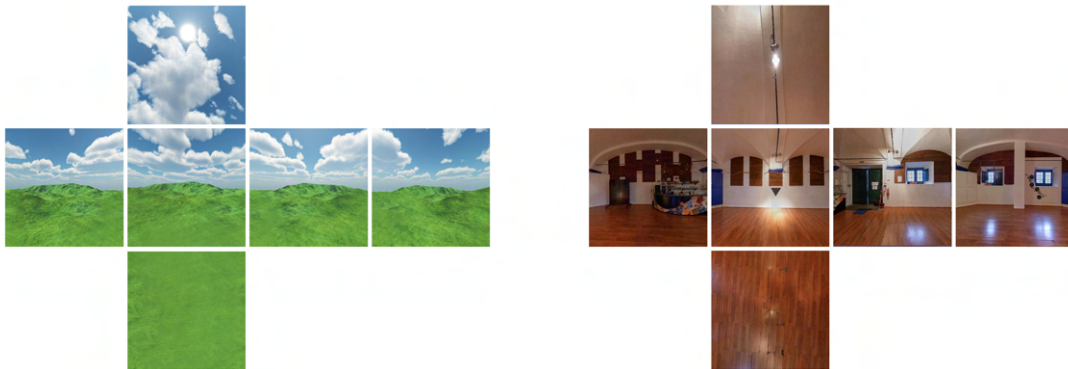


Figure 3.10: Cube map for a general skybox and Pé de Xumbo's studio environments.

There are many possible applications that benefit from the use of 360° environments for object visualization. Besides providing an extra layer of interaction with 3D elements that suit tridimensional visualization, it can allow users the option of further navigating the object's characteristics and explore how it might interact with a certain virtual surrounding. Consequently, it can thus serve as a prior preparation to feel out how the selected model may look when inserted in the video (e.g., simulate lighting conditions before the video). Moreover, having the possibility of positioning 3D models immersed in a known environment can also be an excellent means of optimizing future activities that might occur in it. For example, in the case of the performing arts, an interesting topic approached by traditional dance experts - Chapter 4 - revealed how troublesome it can be to have limited contact with the final performance scenery before the actual performance (e.g., dance studio). In those scenarios, figuring out how certain instruments or clothing items mesh with their surroundings are some of the typical concerns. Therefore, some of the scheduled rehearsal time is usually filled by initial preparations to find the most suitable locations for different items (e.g., props and instruments) as well as examining that space to better potentiate its use. While it is obvious that some details are better analysed with a real experience inside that scenery, a technological solution that provides prior visualization of objects integrated into that space's 360° environment can greatly benefit these artists by ultimately saving up rehearsal time. In this case, the implemented 360° features rely upon the use of Cube Maps to present a straightforward solution for rendering and storing encompassing surroundings. Hence, environments such as Pé de

Xumbo (fig. 3.10 - right image) can be loaded and directly visualized on the extensively discussed 3D viewer interfaces (fig. 3.5 - right image).

3.2.2.2 3D Annotations

By attaining the ability to visualize and interact with the 3D models available, the focus shifted to using these objects as annotation elements. Consequently, new challenges emerged since now the interactions occur in relation to the video and possibly other annotation elements instead of the controlled environment provided by the visualization window (fig. 3.5). Furthermore, besides having 3D models saved and selected, there are evident requirements and dependencies in the working system in order to annotate successfully, including having the video loaded and creating timestamps in the 3D annotation track upon annotating, respectively.



Figure 3.11: Interface to add annotations (center) and respective annotation tracks (highlighted).

Previously, the mechanisms behind uploading and previewing 3D elements were introduced in order to later use them as annotation types in the video itself. Regardless of whether their origin comes from personal or existing models (Artur), one of the main goals derived from inserting these 3D models as annotation types is to make such an integration as seamless as possible. With that in mind, ideally, their structure within the system aims to inherit most behaviors from existing annotation types (e.g., text, drawing, and mark annotations). Thus, similarly to what happens to other types of annotations, the application displays a temporal representation of a given 3D annotation upon adding it to the video. For that purpose, the area containing the existing annotation tracks now also incorporates another track to visualize the timestamps for the 3D annotations. Here, further interactions with the timestamp bars (fig. 3.11 - red and green rectangles) enable users to modify details regarding the annotation's duration and start time directly. As a result, two common actions are used to customize this 3D component: dragging the

annotation bar on the annotation track to adjust the start of the time the annotation is visible and pulling its edge to extend the annotation's duration. Additionally, in annotation types such as drawing and text annotations, altering the color of the annotation besides altering its respective color on the annotation track also affects the color of the annotation in the video itself (e.g., drawn line and text font color).

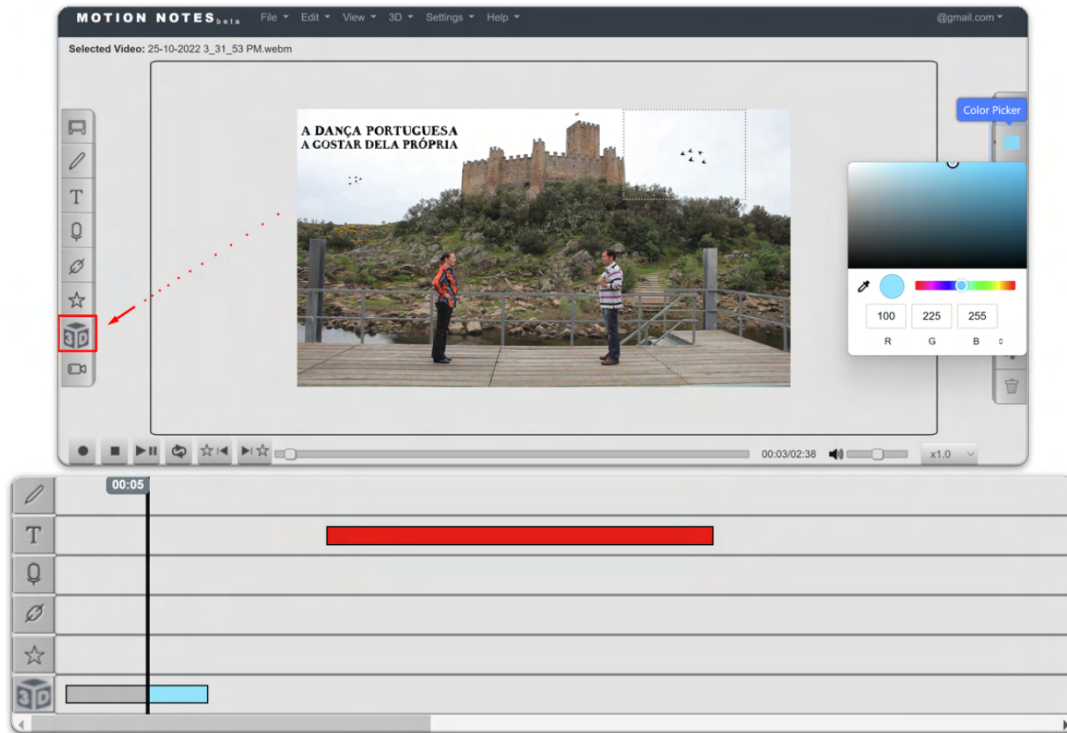


Figure 3.12: Adding and customizing 3D annotations.

In the case of 3D annotations, the color characteristics of the imported model are not intended to change according to its respective annotation bar, however the option of changing the visual representation on the annotation track is enabled all the same due to the potential to provide a more versatile use. For instance, if multiple 3D annotations are added, having different colors for each of them in the 3D annotation track significantly facilitates the process of repositioning their order throughout the video. Consequently, the possibility of altering the color of the annotation's timestamps on the annotation tracks was extended to the 3D annotation type (fig. 3.12). Contrary to the standard interface for uploading and visualizing 3D objects where various backgrounds could encompass the loaded models (e.g., neutral and 360° environments) when placed as annotations over the selected video, these models have their surroundings set as transparent in order to blend-in as much as possible with the video's visual elements. Moreover, in the side annotation triggers, a quick action button is available to aid in selecting recurrent 3D elements. This way, instead of users accessing one of the interfaces back and forth to add the annotations repeatedly, they can use them more efficiently. Figure 3.12 displays a simulated example of this scenario where two instances of a bird's 3D model are added to the sky portion of

the video in quick succession using the side triggers. Each of these added models has its own personalized orientation and dimensions, which are then easily distinguishable in the annotation track due to their different colors.

The figure displays three 'Edit Annotation' modal windows, each with a close button (X) in the top right corner.

- Model3d Modal:**
 - Annotation type: **Model3d**
 - Start time (sec): Duration (sec):
 - Width (px): Height (px):
 - Camera (position): X: Y: Z:
 - Camera (rotation): X: Y: Z:
 - Buttons: **Reset Camera**, **Save**, **Cancel**
- Text Modal:**
 - Annotation type: **Text**
 - Start time (sec): Duration (sec):
 - Text:
 - Buttons: **Save**, **Cancel**
- Draw Modal:**
 - Annotation type: **Draw**
 - Start time (sec): Duration (sec):
 - Buttons: **Save**, **Cancel**

Figure 3.13: Settings interface for annotation-specific options.

Further customizations are also possible by accessing the settings modal relative to the selected annotation simply by right-clicking on the targeted annotation bar (fig. 3.13). Similarly to other annotation types, modifying the temporal representation of a given annotation can either be achieved directly on the respective annotation track or by interacting with this complementary interface. Moreover, 3D annotations can be adapted further by changing the dimensions of the canvas that contains that added model or adjusting the camera's position. The latter can be especially relevant due to the interactions users are capable of conducting over 3D elements. For example, users might excessively zoom out of the object, making the model no longer visible. A simple solution to this problem resides in resetting the camera to its initial position in order to allow users to interact with the object just like when it was first placed over the video.

Lastly, an important element concerning the inherent nature of the video content is the existing annotation's timeline throughout the video. In the traditional annotation mechanisms, the behavior of the annotation track as a whole is quite simple: while the video is playing, a vertical line (fig. 3.12 - 00:05 seconds) traverses the annotation tracks proportionately to the video's duration and stops at the respective time of that video's timestamp when the video stops. For instance, if the video stops at the third second of a ten-second video, the vertical line will stand at the 2/10 position of the track's length. Thus, if an annotation is added at the X second of the video, the next time the video reaches that temporal mark, the annotation will be loaded onto the video and disappear immediately after hitting the end of its respective duration. However, in the case of the

previously existing annotation types, this behavior perfectly suits the system's normal functioning. For example, upon reaching the timestamp containing the beginning of a text annotation, the vertical line representing the current video time can resume its horizontal movement right after loading the aforementioned annotation. In this instance, the process of creating the text element to be displayed on the application is almost instantaneous, as is the case for the other traditional annotations (e.g., drawing and mark types). Nevertheless, 3D models take substantially longer to load before being ready to be displayed on the video. This would naturally lead to undesirable waiting times every time the video reached a 3D-based annotation timestamp. In order to solve this, the system pre-loads all the tridimensional models associated with the video's annotations. As a result, when the video is first loaded, all types of previously created annotations are traversed and added to create their visual representation in the annotation tracks. Simultaneously, for each of the annotations, if they are of the 3D annotation type, then they are loaded onto the video and set to be non-visible. Finally, whenever the video hits the 3D annotation's timestamp, the model is set to be visible yet again, thus creating uniform processing times across the loading of annotation for all existing annotation types.

3.2.3 Pose Estimation

Chapter 2 approached some of the challenges regarding the use of different pose estimation libraries, their inherent characteristics, and respective behaviors. Using the automatic detection of keypoints (e.g., knees, hips, and shoulders) can enable users to visualize simplistic skeletal images directly, thus potentially enhancing the study of the quality of people's movements. In this system, pose estimation is aimed at augmenting the annotation work in a different manner than traditional annotation types that, in essence, represent different possibilities of annotating over a video (e.g., using drawings, text, and images). Instead, the focus is centered on the human components commonly present in multimedia content to display visual information about people's positioning and posture automatically. Overall, such algorithms can significantly complement this annotation system by seamlessly highlighting relevant details regarding a person's body parts. However, in the context of this project, partners such as Pé de Xumbo are especially keen on this integration due to its applicability in the traditional dance world. Nevertheless, this concept of approximating the structure of people's poses to simplified figures can be applied across different fields of work. For instance, from a more ubiquitous standpoint, it can be utilized in the health sector to study and contribute positively to people's well-being, for example, by correcting a person's posture when sitting in front of the computer for too long. Similarly, in the sports world, understanding an athlete's motion patterns can improve performance, potentially prevent injuries, and help adjust a player's training load based on movement differences over time. Consequently, the sports area motivated the creation of the second preliminary study using basketball as a case

study which is further detailed in the next chapter.

Prior developments in the MotionNotes system integrated PoseNet as the pose estimation to infer people's keypoints throughout the video for subsequent analysis. The initial integration of this pose estimation technique allowed base keypoint inference to be directly displayed over people's bodies, with quick computation times being the main advantage of using this computer vision model. However, the lightweight nature of PoseNet results in some of the inconveniences mentioned before, the most visible being the lack of accuracy when applied to various people throughout a video (fig. 3.14).



Figure 3.14: PoseNet (top) vs OpenPose (bottom) examples of keypoint estimation.

Opting to integrate OpenPose as the library to expand pose estimation features allowed the system to overcome these limitations and thus expand the pose estimation functionalities using a more stable keypoint inference as its base. In fact, without further developments conducted over OpenPose's pose estimation, it visibly surpasses PoseNet's pose estimation stability. However, the pose estimation area of computer vision is rather recent and is continuously being updated by the scientific community. As a result, coupled with wanting to preserve this initial integration, the MotionNotes system makes both options available yet restricts the complementary features described ahead to the

OpenPose's alternative.

3.2.3.1 OpenPose Integration

For the OpenPose integration into the system, contrary to what happened with PoseNet, which ran directly on the client-side, the pose estimation based on the OpenPose library must run on the server-side in order to take full advantage of the capabilities of the specialized (GPU-powered) hardware. Therefore, the system relies upon the use of the OpenPose binaries in order to execute the necessary commands. There is also the option of using a compiled version from the available source code, thus allowing the possibility of making some alterations to create flexibility if needed. Even so, given that this project's primary requirements mainly aim at accurately displaying pose estimation, the current integration utilizes these binaries at the center of the pose estimation computations due to them being optimized to run the primary pose estimation functionalities (e.g., contains cuDNN to optimize memory use).

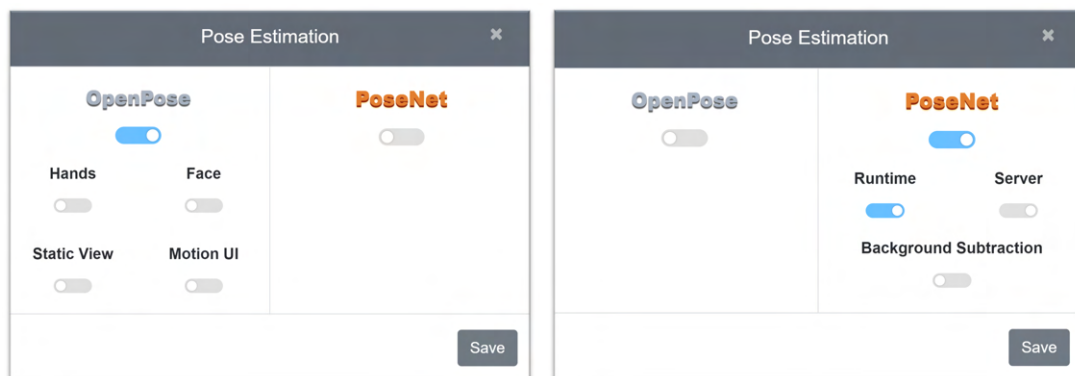


Figure 3.15: Settings interface for extra pose estimation options.

Available at a dedicated tab in the top area of the application, users can toggle the visualization of an interface containing the customization settings for both pose estimation alternatives displayed in the figure above. Briefly, the existing PoseNet technique allows users to either run the model at runtime or re-utilize previous computations in case there is prior pose estimation data regarding the selected video. Moreover, users can also remove the video's background so as to exclusively display the identified skeletal images. On the novel side, the OpenPose's pose estimation contains four main customization possibilities:

- **Hands keypoint identification** - Attaches further keypoint information for both identified hands.
- **Face keypoint identification** - Appends additional face keypoint data.
- **Static View** - Displays the video with pose estimation data directly embedded into the source file.

- **Motion UI** - Reveals an interface for further pose estimation analysis.

The first two options of adding extra keypoint information to the hands or face exist only to provide the users with additional visualization data. However, they do take a toll on the resulting computation times since the keypoint inference must also consider more keypoints for each person in each frame throughout the whole duration of the video. Even so, these options open up interesting possibilities for future developments beyond visualization (e.g., ceramics and other craft arts movement analysis). The other two customization settings affect the manner in which base pose estimation (keypoint identification and visualization) are displayed. Similarly to the system's traditional annotation mechanisms (e.g., drawing), to activate pose estimation, users must trigger the intended actions using the application's interactive interface. In this case, pressing the corresponding pose button sends a request to the server, which will process the video in order to extract pose estimation data and respond accordingly. This behavior is quite similar to the existing features that often use the side triggers to activate the annotation functionalities, thus striving for consistency within the system.

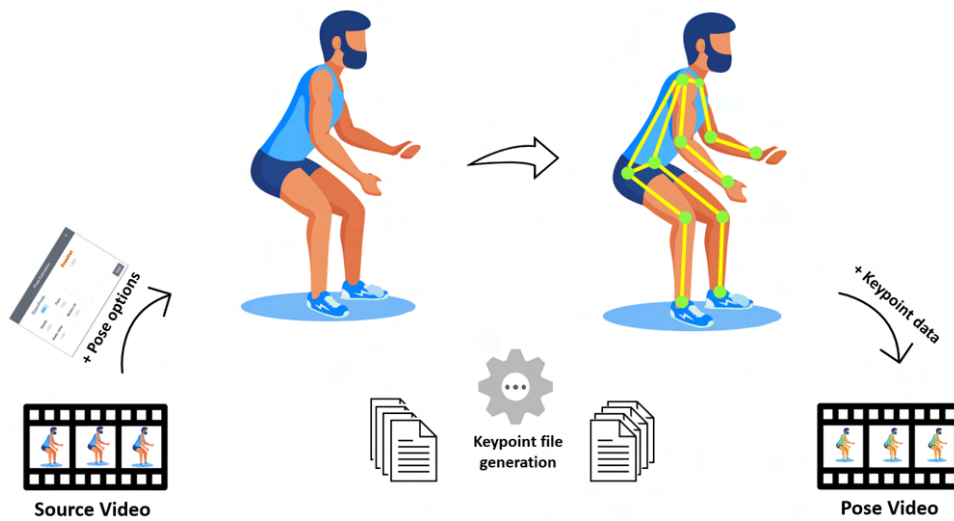


Figure 3.16: Pose estimation client-server's communication overview.

After activating pose estimation, the client sends a request to the server-side of the application that receives both the id of the selected video as well as the toggled options to begin the pose estimation execution. Afterwards, using a native node.js call, the OpenPose binaries containing the already provided models run the inference algorithm successively over each video frame. In the OpenPose documentation, several flags are available to customize the type of processing done to the video (e.g., `-net_resolution` to adjust the speed-accuracy ratio) and respective output results. Therefore, options such as tracking the hands or face keypoints are executed as additional arguments added to the original function call that runs the OpenPose binaries. For example, a standard request to display base keypoint visualization results in the creation of a command that begins with

the OpenPose's binaries executable (e.g., <executable_name>.exe) followed by flag values that define the intended output. The table below describes some of the most relevant flags in this system's context:

Flag	Status	Description
-video	Mandatory	Defines the input file path containing the video to be processed.
-display	Mandatory	Hides the visual real-time processing over the selected video.
-write_json	Mandatory	Creates output files for each video frame containing keypoint data.
-write_video	Mandatory	Creates a video file with embedded keypoint data.
-hand	Optional	Adds additional hands keypoint data.
-face	Optional	Adds additional face keypoint data.
-net_resolution	Optional	Adjusts the intended resolution used to compute pose estimation.
-tracking	Optional	Optimizes consistent person tracking over multiple video frames.

Table 3.2: OpenPose's most prominent flags.

The default execution of OpenPose creates an auxiliary window to visualize the detected keypoints on the video throughout the algorithm's computation at the maximum frame-rate possible. Despite possibly being useful for external testing, for instance, to test the algorithm's accuracy according to the specified input, on the server-side it is irrelevant since keypoint visualization only occurs on the client side. Consequently, the display flag's value is always set to 0 in order to prevent an unnecessary load on the system. Figure 3.16 displays a simple overview of the overall processing done over a standard server call to receive pose estimation data. The resulting computation outputs several json files for each video frame containing, among additional data, the necessary keypoint data to process and display pose estimation.

With the purpose of simplifying the communication within the application, subsequent processing is done to merge each of the outputs into a single json file to return back to the client. This is accomplished using the -write_json flag described above. Similarly, the -write_video is used to provide flexibility to the system. Web browsers are known to struggle with reliably streaming or displaying multimedia elements such as videos. For instance, Youtube is arguably the largest video content distributor used regularly to upload and watch a video. Even then, dropping frames¹⁰ is quite common, especially at higher resolutions. On the programming side, synchronizing the video's frames and the drawing rate of keypoints is tricky since there is no guarantee that frame drops will not occur or pause/play synchronicity will be stable. As a result, instead of returning the keypoints in the form of a json file for the client-side to process and draw them, there is also the option of directly displaying the outputted video containing embedded keypoints (fig. 3.15 - Static View). Moreover, at a later stage of development, this option might also allow users to directly download the pose estimation embedded version of the video for personal usage. The temporal and spatial complexity of the respective computations remains constant, however, the raw outputted video consumes considerably

¹⁰In this context, frame drops occur to compensate in order to keep up with the connection's bitrate.

more memory than the original source video that does not include keypoint data - about three times the original size. As a result, an intermediary processing step converts and significantly compresses the raw outputted video to the standard system format for video storage (.webm). This process is achieved using the FFMPEG¹¹ framework. At closer observation, the computation steps required to retrieve pose estimation data back to the client side of the application must be synchronized in order to avoid errors. For instance, although the OpenPose's execution concludes by storing all the json files - frame keypoint data - and embedded video, the server must wait for the file merging and video compression completion to respond with the necessary data. It is then the client side that, depending on the Static View status, either displays the OpenPose's video or makes use of the received keypoint data to render and connect the identified keypoints for each person present in every frame of the video. Intuitively, the latter iterates through every frame field in the json object and, for each identified person, analyses the values of the identified keypoints.

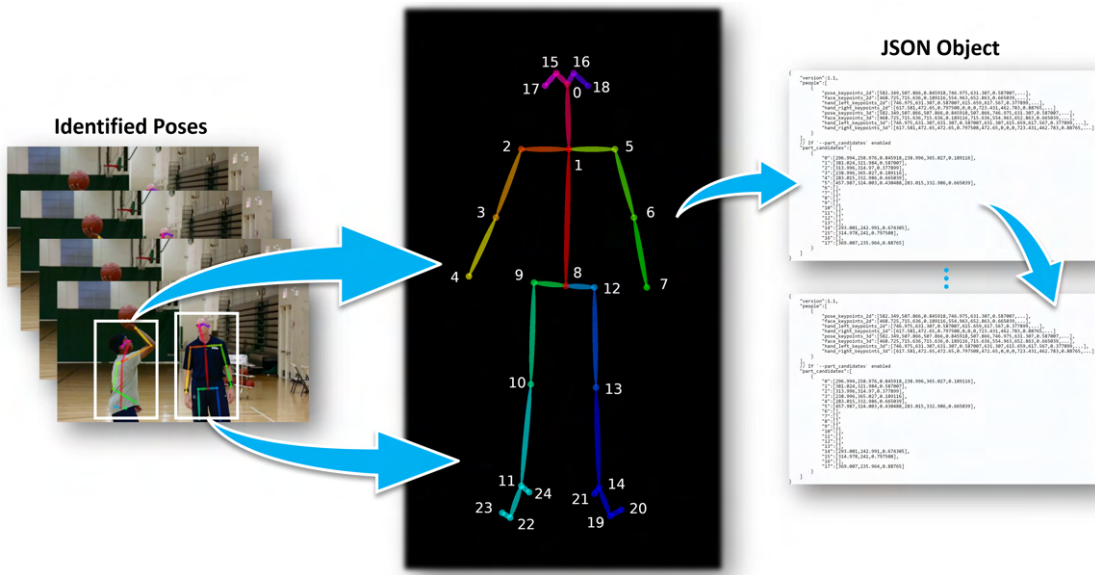


Figure 3.17: Pose estimation keypoint data generation.

In Fig. 3.17, at the center section of the image, there is a visual representation of the default 25-keypoint structure used by the OpenPose's algorithm to match people's poses. When a person is identified, the respective object field containing their keypoint data is available in the following format:

"pose_keypoints_2d" : [x0,y0,c0,x1,y1,c1,x2,y2,c2...x24,y24,c24]

In which the pose_keypoints_2d fields contain the coordinates for all of the twenty-five keypoints relative to the video's dimensions. Each keypoint is represented by chunks

¹¹FFMPEG is widely available and reliable open-source software for handling multimedia content. Available at <https://ffmpeg.org/>.

of three values that define the keypoint's position as well as its confidence score ([0,1] interval with 1 being the optimal value). For example, if the keypoints' arrays for a given person at a specific frame begin with the values [1070, 670, 0.9, (...)], it means that the nose keypoint was found at coordinates (1070, 670) with a degree of confidence of 0.9 (fig. 3.17 - 0 keypoint). Evidently, these coordinates are attributed based on the video's initial resolution and thereby converted to be properly displayed in the application's video section. Thus, the application is then able to draw the keypoints and connect them throughout the video to allow users visualize base pose estimation.

3.2.3.2 Pose Estimation and Annotation

Similarly to the integrated 3D elements, the developed pose estimation functionalities aim for other goals beyond fulfilling the necessary functional needs, in this case, achieving the automatic detection and visualization of people's poses. In fact, a central goal regarding the development of pose estimation features is to integrate components without adding unnecessary complexity to the system while efficiently complementing the existing annotation mechanisms. Therefore, behaviors such as using the side triggers to activate pose estimation visualization (fig. 3.18 - right) are used yet again to be coherent with the previous incremental behaviors and their respective functionalities. Moreover, since the overall goal of the annotation tool is to augment annotation work conducted over a video using different annotations, there is potential in the idea of linking both concepts together. This thought process partially motivated the creation of further annotation mechanisms connected to pose estimation.

The goal of having annotations coexist with the pose opened doors for practical applications such as explicitly and unequivocally referring to a specific body part (e.g., directing drawing annotations toward a person's knee). However, more than static annotations might be required to link human motion and note-making together properly when analyzing human motion. There are several valuable practical examples concerning fields of work such as education as well as areas that often must resort to presentations to an audience where keeping people's attention is key. In this context, annotations are often used to, among other things, highlight observations relative to human behavior and movement (e.g., correcting overall posture, improving effective public speaking, and highlighting dance movement patterns). These factors motivated the development of the first complementary pose estimation feature: dynamic annotations.

Traditional annotations, for instance, drawing and text annotations, have a well-defined behavior regarding their positioning in the video. Throughout their duration, unless users directly change their location, they will remain in the same place they were last placed in. However, when associated with human motion, linking a specific motion to its respective annotation that moves accordingly to that body part can greatly benefit both the system and its users. On the system's side, it explicitly interconnects the existing annotation functionalities with the novel pose estimation features. On the other hand, it

helps to reinforce and clearly point out an observation of a person's motion throughout the annotation's duration. In the basketball case study described further ahead, dynamic annotations also proved valuable in improving athletes' attention spans by resorting to the annotation's dynamic visual cue as opposed to standard static annotations.

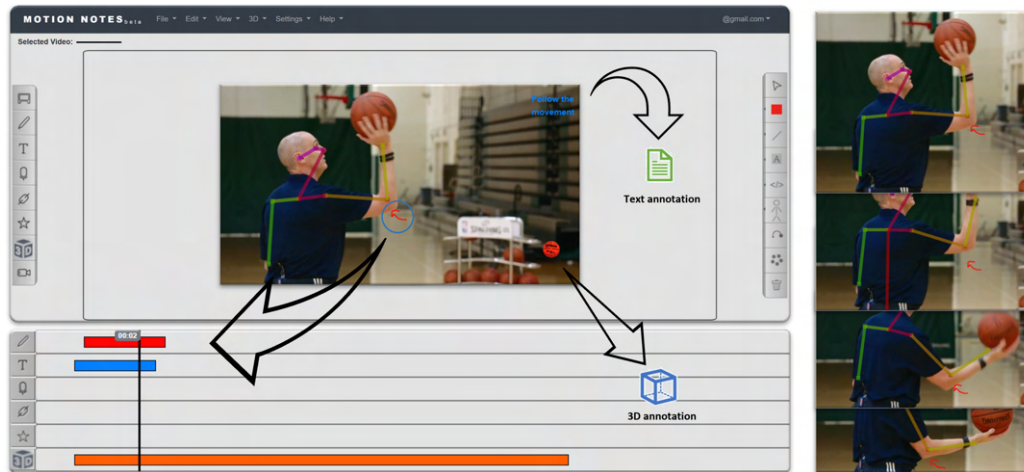


Figure 3.18: Example of a dynamic annotation (arrow - elbow).

With all aspects considered, the resulting developments lead to the creation of mechanisms to handle the direct association between a person's keypoint (e.g., elbow, knee) to any annotation type such as drawing, text, or 3D annotation as portrayed in fig. 3.18. It is worth to note that despite the OpenPose's accuracy in determining a person's keypoints, it is inevitable that in variable environment conditions they will fail to correctly identify a person's pose. Thus, if a dynamic annotation is added to a person's body part in such a time period, the annotation might present erroneous behaviors (e.g., follow a miss-identified keypoint). Nevertheless, OpenPose's algorithm is sufficiently reliable to accurately compute pose estimation in the vast majority of video recordings. Moreover, from a developmental standpoint, if in the future another pose estimation framework is used to improve overall performance, dynamic annotations can easily be reincorporated.

3.2.3.3 Pose Estimation Expansion

Beyond the relationship between the existing annotation mechanisms and simple keypoint visualization, to answer the question, "Can pose estimation-based features be a valuable complement to the MotionNotes annotation system?" further development phases are required. The reason being that the automatic highlighting of people's poses as a means to aid annotation work and the analysis of human motion might be insufficient to tackle some of the following identified needs:

1. Increasing the possibilities of annotation work directly linked to pose estimation.
2. Individually narrowing down pose analysis in a controlled environment.

3. Automatically obtain information regarding the execution of tabled movements and gestures.

The first thought-about requirement derives from the implicit goal of having a cohesive system that seamlessly integrates the new functionalities without creating unnecessary complexity. Evidently, the system still retains the ability to add different types of annotations over the video while pose estimation is active. However, having dynamic pose estimation establishes a direct link from the annotation process to pose estimation components (keypoints), thereby improving the direct association between the two.

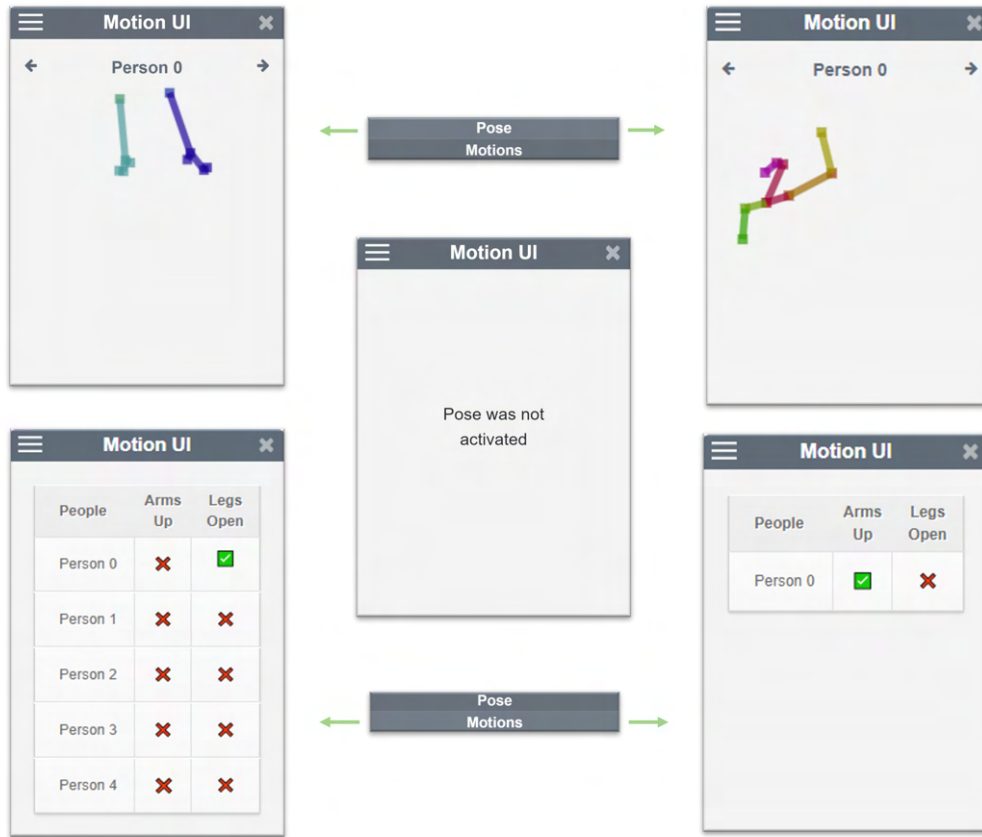


Figure 3.19: Motion UI interface examples.

The other two points solely focus on the potential of pose estimation applied to the human component present in a video. In the 3D annotation details, the decision to keep color customization options regarding their respective annotation bar in the annotation tracks derived from the need to distinguish between several 3D annotations in the same video simply by associating each annotation to a respective color. Similarly, analysing multiple people and their movements simultaneously can prove inefficient and unnecessarily challenging due to the multitude of movements happening at the same time. With the purpose of solving this problem, an interface was created to expand pose estimation with related functionalities: Motion UI (Fig. 3.19). The goal is to have a separate window

where users can specify which person's pose they want to visualize at a given moment depending on the video frame they are currently on (Fig. 3.19 - top interfaces). As a result, when pose estimation is displayed over the video, it also duplicates its respective drawing call to this interface, and it filters the person to be displayed based on the currently selected person id (Fig. 3.19 - "Person 0"). Additionally, further interactions can be explored, such as using this interface to directly compare people's postures and movements by placing them side by side, which was often mentioned in the basketball case study. Accordingly, specific rule-based actions and gestures can be inferred using accurate pose estimation data that do not place too much weight on the system's resources as opposed to adding another layer of machine learning. For instance, in basketball, identifying if an athlete's arms are raised when challenging an opponent's shots can be easily determined using pose data and then applied to practice and game sessions. Similarly, some traditional Portuguese dances emphasize the importance of the lateral movement of people's legs, such as in the "Dança do Pézinho" dance. As a result, to explore the potential of automatically identifying certain movement patterns, there is a complementary tab in the Motion UI interface to directly visualize such information. In Fig. 3.19 - bottom interfaces - there are practical examples of these same use cases that apply the ArmsUp and LegsOpen gestures. Intuitively, the json object containing keypoint data for that respective video is initially processed to immediately determine for each frame which of the tabled movements are being executed and by whom. If successful, that person's movement is labeled with a symbol or with a red cross otherwise (e.g., *Person 0* - left-hand side). Thus, the object is iterated over each frame and identified person in order to create a new object to check gesture recognition at a specific frame. That way, each time a frame change is detected, the object is accessed at that frame's field and evaluates the values available for each person and their tabled movements.

Despite presenting a fairly simple execution flow, several details require attention. OpenPose's inference is not hundred percent accurate; therefore, some miss-classifications might directly impact movement identification. Moreover, people often move and have their body parts partially or entirely occluded by objects or humans. Consequently, the criteria used to recognize a gesture should have computational redundancy whenever possible, meaning that different options should be covered to correctly identify movements even in unstable and ambiguous scenarios. For instance, to correctly identify the "LegsOpen" movement, chosen as an exploratory example for this pose estimation functionality, one could extrapolate whether the movement is being executed at a given moment simply by comparing the length between a person's shoulder and comparing it the distance between that person's toes. However, despite being a valid approach to estimating this movement's execution, it will often fall short whenever keypoint data is missing regarding any of the involved keypoints (shoulders and toes). In order to identify as many positive cases as possible, the two other redundant criteria are used: comparing the length between both hips and opposite toes/heels and, lastly, the angles from each knee to its respective lateral foot.

Perhaps the most complex from a developmental standpoint regarding these complementary interface's features: the accurate tracking of people. OpenPose makes no assurances regarding if a person receives the same id in successive video frames, even if that person barely moves from one frame to another. In terms of overall usability, this is perfectly acceptable in cases where there is a sudden shift in environments and people in the same video (e.g., changing from an indoor to an outdoor scene). However, it becomes ineffective in most stable scenarios in order to have dynamic annotations, individual pose estimation in the Motion UI interface, and motion recognition. The reason being that people identification will be incoherent, leading to a person having multiple ids in a short span of time (e.g., Person 0 attributed alternatively to two different statically-placed people for ten seconds). OpenPose tried to implement a person-tracking version available through the use of the `-tracking` flag described in table 3.2 based on an additional layer of machine learning. However, due to its subsequent complexity, it had to limit pose estimation to a single person in order to perform person tracking.

Nevertheless, this option is currently released as an experimental version, and besides frequently failing to track someone accurately, it sacrifices pose estimation's multi-person inference to a great extent making it impractical. The subsequent conceived solution thus focuses its approach on being both lightweight and creating consistency for people identification in successive video frames, especially for stable scenarios. Stable scenarios are defined in this context as having the following characteristics: accurate keypoint inference, constant human quantity, and consistent backgrounds. This means that as long as pose estimation correctly identifies people's keypoints and these same people are moving in a well-defined trajectory within the same space, then person tracking must work. As a result, scenarios with volatile amounts of people and sudden changes in spaces (and consequently unexpected keypoint repositioning) make no guarantees of accurate people tracking.

In video content, people can move freely with variable background areas, lighting conditions, and object interactions. Therefore, to properly track people using existing keypoint data, it is essential to look at the relationship between the identified keypoints across each frame. The algorithm begins executing immediately after all the received keypoint files - one file for each frame - are merged into a single keypoint object where the tracking changes will occur (e.g., reordering people's ids in the initial frames). Consequently, the modifications resulting from the tracking algorithm will be persistent for future pose estimation function calls. This person-tracking algorithm behaves as follows:

1. For each video frame, gather the previous X frames' respective keypoint data. In which X symbolizes how many prior frames should be considered.
2. In each comparison, compute the differences between the actual frame's identified keypoints and its current predecessor.

3. A map representing the likelihood of a given person (A) in a frame Y being the same as every identified person for each $Y-\beta$ ($0 < \beta < X$) frame is created using these differences.
4. Using this mapping, people might change ids in the current frame based on the similarity among previous identifications.

The algorithm's structure is relatively straightforward: each frame is iterated and looks back a set number of frames to find the previous people's ids (e.g., Person 0 and Person 1) and respective differences to each person present in the current frame. Then, based on the differences between ids in the current frame and prior ones, people's ids might be reordered to be consistent with their positioning throughout the video (fig. 3.20). However, several aspects need consideration across each of these steps. For instance, the computed differences between current and prior keypoints are only to be used for tracking if they are valid. An example of this scenario can happen if a given Person 0 has a valid left shoulder keypoint in frame X and invalid (e.g., confidence score of 0) in frame X-1. In this case, the left shoulder keypoint would be disregarded to compute to differences between Person 0 in the current and previous frames.

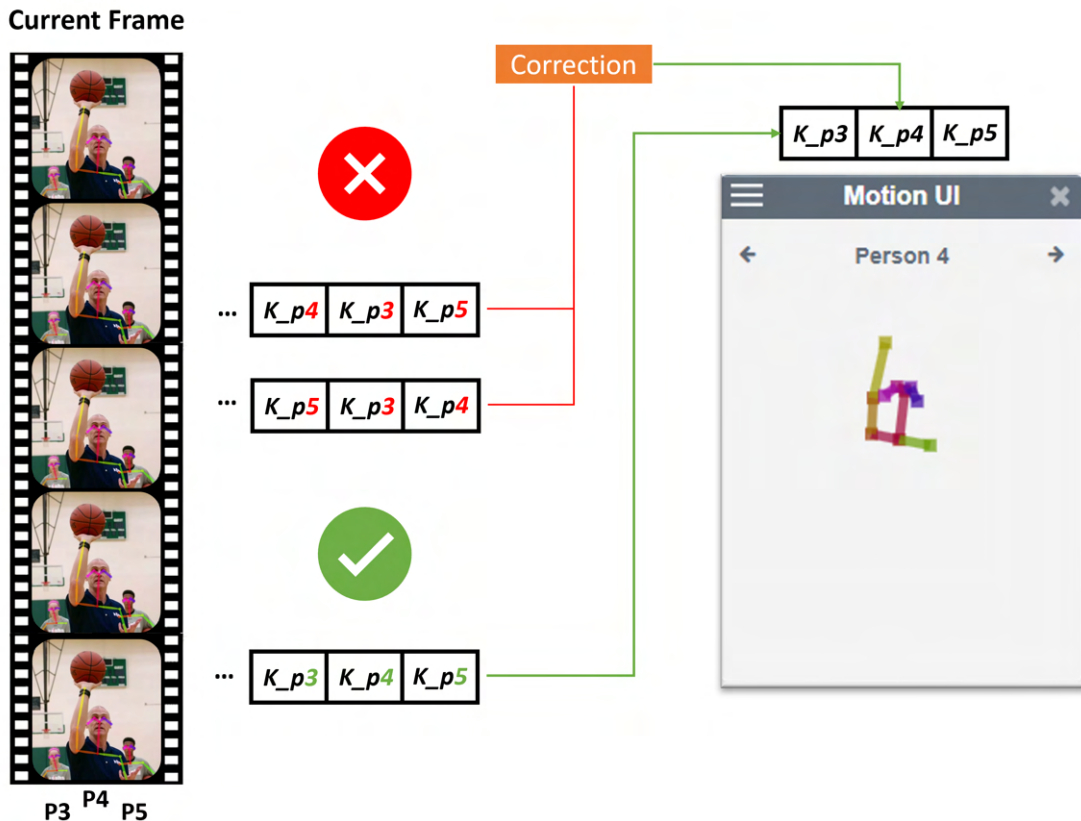


Figure 3.20: Keypoint tracking across multiple frames.

Fig. 3.20 displays an example containing three people identified from left to right as Person 3, Person 4, and Person 5 (P3, P4, and P5), each with their own respective keypoints

(K_p3, K_p4, and K_p5). Since a consistent keypoint order was established in previous frames, the most recent frames compare their keypoints with previous ones and rearrange their position to match their order. Therefore, for instance, when Person 4 is displayed in the Motion UI without changing between different people's keypoints. Nevertheless, there are fallible cases, such as when people occlude each other, thus leading to changes in the number of people in the video. Tracking becomes unstable in such scenarios due to the different possibilities (e.g., a new person appeared or a previously existing person was hidden). Additionally, when limited keypoints are present in a video (e.g., only the arm keypoints), OpenPose is liable to erroneously conclude that there are two people each with one arm raised when there is only one raising both arms. While the latter results in an additional pre-processing case to merge keypoint data if necessary (fig. 3.19 - top left), the first falls outside what was set as a stable scenario. However, since this behavior is quite common in people's recorded interactions, a prototype was developed to extract 3D coordinates in monocular videos using the previously mentioned MiDas model (fig. 3.21 - right image).

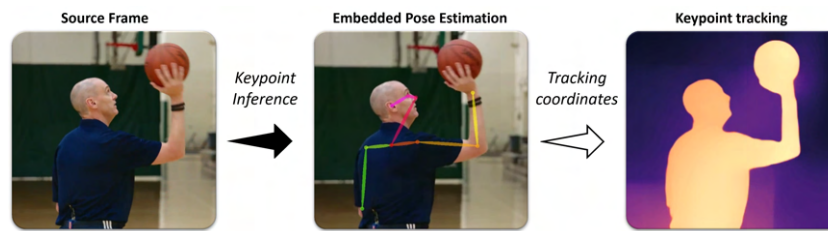


Figure 3.21: Pose Estimation 3D tracking layer.

This tracking version is regarded as a prototype due to the fact that depth estimation techniques (e.g., MiDaS, and LeRes) have associated weighty time and computational complexity since there are significant inference times to extract 3D coordinates. However, it works as a valuable route towards solving occlusion cases by objects and humans by tracking people using defined 3D keypoints (e.g., re-identifying someone upon moving out of a person's back).

EVALUATION AND RESULTS

This chapter presents the preliminary and final evaluation methodologies used to validate and study the annotation system containing the newly integrated functionalities. In this first section, the feedback and results received in two independent workshop sessions summarize the semi-structured interviews conducted in order to evaluate the system at an earlier stage. Notably, the feedback received from both led the way for two scientific papers. While the first is published¹ as a poster paper presented at the *IMX*² conference the second is currently accepted and soon to be published at the *MUM*³ international conference in the same format. Finally, the concluding section evaluates the system using the *SUS* usability questionnaire complemented by some additional relevant questions.

4.1 Preliminary User Tests

There is a clear contrast between the initial evaluation conducted with the developed system and the final one since the main goal in these preliminary user tests was to acquire valuable feedback, gather possible use cases, and guide future developments. Moreover, the semi-structured format of the interviews, further described below encouraged users to discuss how they felt about each presented functionality and brainstorm different ideas. Interestingly, this triggered two very distinct debates in each of the sessions fitting to the circumstances since the earlier sessions focused on the 3D features while the latter explored the pose estimation functionalities.

4.1.1 Case Study: Traditional Dances

In this preliminary case study, there was an opportunity to organize a workshop session with experts in the performing arts world, more specifically in Portuguese traditional dance. Notably, these participants are also the directors of the previously mentioned PédeXumbo association and are currently documenting and studying choreographies

¹<https://dl.acm.org/doi/abs/10.1145/3505284.3532972>

²<https://imx.acm.org/2022/>

³<https://www.mum-conf.org/2022/>

that are mostly no longer performed or taught. The intent of the WEAVE project to protect and contribute to preserving cultural heritage in European communities makes the PédeXumbo foundation a noteworthy partner. Consequently, given the experience of these participants, the workshop aimed to receive feedback regarding the new 3D functionalities in the context of traditional dance forms and cultural heritage in general.

At an earlier stage, the specialists were given a brief demonstration of the MotionNotes system, thereby introducing the basic annotation mechanisms through the use of simple annotation types (e.g., text and drawing annotations). Afterwards, the focus shifted towards the 3D-based elements by centering the attention on the following two primary interfaces:

- **3D Models Manager** - In this interface, the key interactions fall upon three distinct actions. Firstly, users are capable of importing 3D models to have available while working on the annotation system (fig. 3.5). Then, by selecting from the accessible models, the 3D objects can be displayed at the center of the interface and interacted with. Additionally, there are various 360° backgrounds for enhanced object visualization, which can be selected to replace the standard neutral background.
- **Video Canvas** - Upon selecting from the available 3D models, users can thus place them anywhere on top of the video as an annotation element. Moreover, the same interactions with these objects are possible either through moving, rotating, or scaling actions and subsequently, the position and respective timestamps are stored.

The participants were then given an initial demonstration using smaller-scale 3D models of musical instruments on a 2D traditional dance video (fig. 4.1). The primary objective of using these annotations was to enhance the perception of the dance's musical accompaniment as well as to allow users to analyze each particular instrument in a 3D setting to better understand its characteristics. Interestingly, the specialists gave positive feedback regarding this experiment. Besides being able to conceive multiple scenarios in which video-integrated 3D elements can be helpful, this case, in particular, was received as especially justified due to the fact that the relationship between dance and music in this traditional context can be complex at times.

Suggestions regarding the use of three-dimensional objects as historical and cultural references were also a subject of discussion. The experts elaborated further using examples such as having 3D models to illustrate the historical background of the dances being performed, for instance, by representing the attire used at the time. Moreover, using cultural items to enhance the video's information can potentially aid an audience in situating themselves geographically in the context of the dance's origin. From their perspective, such scenarios can be especially relevant due to the fact that some traditional dances are fading over time.

On another note, when presented with a customizable 360° background feature, the participants noted the possible applications in their field of work. Dance practitioners

and teachers often struggle to optimize the limited rehearsal space and time available to improve and learn. Consequently, being able to externally visualize the environment they practice in can help them prepare and enhance their creative process. Furthermore, the specialists also hinted at the possibility of developing these functionalities further (e.g., measuring within the 360° space, including 360° videos) to allow for more flexibility when interacting inside the selected tridimensional space.

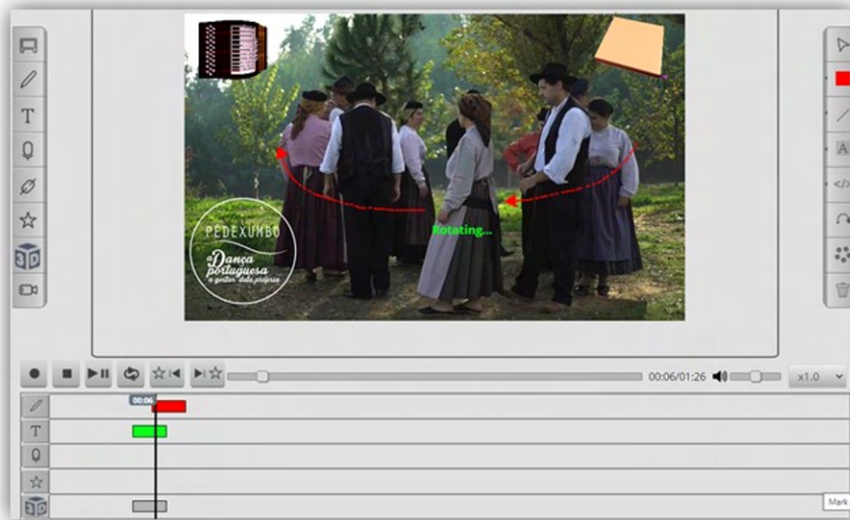


Figure 4.1: Sample annotation scheme used to introduce 3D functionalities in the described workshop.

In this preliminary study, the experts greatly supported the relevancy of the presented 3D features integrated into the MotionNotes annotation system. Through discussing examples they found pertinent in their professional context, their feedback revealed that these 3D functionalities are valuable when applied to traditional dance research and practices. Using detailed objects to highlight unnoticed information or reproduce intended circumstances, such as simulating where an object would be placed in a real-world scenario, can be critical. Additionally, given the flexibility of the 3D object's nature, such capabilities are likely to be convenient across different areas (e.g., sports, education, and health). Therefore, this workshop session motivated the development of the mentioned scientific paper (later-breaking work) recently submitted at the *IMX* (Interactive Media Experiences) conference, where these topics are further detailed [80].

4.1.2 Case Study: Basketball

The main objective when conducting this study was to gather feedback on the utility of the pose estimation features in the sports world by conducting a controlled hands-on discussion with knowledgeable basketball people. Hence, participants with relevant competitive experience were invited and evenly divided among two groups: athletes and coaches. The idea of having people with different federate⁴ experiences within the same setting were deliberate in order to acquire a complete understanding of how such features could be useful from both a player's and coach's perspective.

From a statistical standpoint, of the ten subjects, one was female, and the other nine were males, where half of the participants were athletes, with the other half being coaches with an average of 9.2 and 23.6 years of experience in their national basketball federation, respectively. Regarding their ages, players ranged from 16 to 24 (mean: 20.6, standard deviation: 3.1), while coaches were distributed between 25 to 48 (mean: 39.8, standard deviation: 8.6).



Figure 4.2: Sample frames selected in the workshop to demonstrate a player's shooting motion.

The use of video-based analysis in practice sessions is quite common in the world of sports, whether to highlight subtle details that previously occurred in a competition or enhance athletes' overall performance. In the basketball context, research has demonstrated the importance of how factors such as the height and angle of a shot can impact the success rate of a player's shooting motion [81]. As a result, utilizing an annotation tool can be a viable way to effectively study and point out specific details both to athletes and instructors. The workshop was thus organized to include an open discussion with each participant regarding each pose estimation feature, further summarized in appendix C.

⁴In this context, federate refers to being formally affiliated to the Portuguese Federation of basketball either as player or as a coach.

Even though the test subjects were given the option of importing and using other materials, a previously selected video⁵ recorded during a basketball practice was handpicked to be used throughout the workshop (fig. 4.2). This pre-selected video was selected due to containing several visual demonstrations common to standard basketball exercises. Moreover, there is also an emphasis on the basketball shot motion, which is often worked on during basketball drills. Thus, participants could visualize familiar aspects that are frequently explored during practice sessions.

Independently of their role as either athletes or coaches, all participants possess advanced technical and tactical knowledge within the sport, thereby allowing them to provide more thorough feedback in this setting than the typical user. Early on, participants were asked to answer basic questions about their personal information and briefly describe their competitive basketball experience. Then, to contextualize the general use of the annotation tool, users were given a demonstration, including the initial annotation features (e.g., annotation types, annotation timestamps) and the system as a whole.

The following interview format followed a standard structure regardless of the discussed subject matter. For each discussed topic, the participants observed a short presentation regarding each pose estimation feature. Afterwards, coaches and players interacted with the system using guidance if needed. This approach focused on steering the conversation towards answering if the given feature was relevant to them in their roles as athletes or coaches as a means to enhance their performance. Consequently, the first functionality participants experienced was the base estimation feature visible on the MotionNotes screen through the connected keypoints (e.g., elbows, shoulders) drawn on top of each identified person throughout the video (fig. 4.2).

Among other aspects, it is quickly perceptible that having pose estimation over an athlete's movements allows for an intuitive visualization of angles between a person's joints. This is especially relevant in the context of sports, in this case, basketball, due to the impact the quality of a movement can have on the player and team's performance (e.g., shot mechanics, defensive slide, jumping technique). A simplistic view of an athlete's pose and posture can thus lead to different practical applications that were objectively sought after in discussion with both types of test users. Following their interaction and subsequent discussion, participants engaged with dynamic annotations where notes are placed and directly linked to a specified keypoint to move according to it. Regarding this feature, one previously thought about advantage inherent to dynamic annotations as opposed to static annotations was being capable of stimulating a person's attention span by having a captivating visual cue. As a result, since most coaches have experience with young children, it was also intended to discover how relevant this functionality was to these types of participants. Finally, athletes and coaches alike examined the two tabs available in the Motion UI window (fig. 4.3). Here, the two main concerns were addressing the usefulness of having both a neutral background to display selective pose

⁵<https://www.youtube.com/watch?v=SpjsZA0kq1s>

estimation, i.e., choosing whose person's pose to watch at a given time, as well as a simple gesture recognition table. For the latter one, the *ArmsUp* and *LegsOpen* tabled movements aim to simulate two essential components in basketball athletes' movement patterns: shooting motion and defensive slide. As a result, in the individual interviews of coaches and players, practical examples such as improving the shooting technique and having a stable defensive slide were subjects of discussion with all participants regarding this last feature.

A summary of the participant's feedback as described in the submitted paper is as follows:

[Feature 1] - Human keypoint visualization via OpenPose's pose estimation

The feedback received for this initial feature was overwhelmingly positive, with all players and coaches alike agreeing on the usefulness of visualizing a presented person's posture through their drawn skeletal image in the sports context. Interestingly, every participant also mentioned its applicability in studying and improving a player's shot mechanics, for instance, by highlighting the angle between the arm and forearm throughout the shooting motion. Even though the feedback was very uniform, two of the five coaches commented on their experience in trying to simulate something similar while in a team setting. However, they noted how such a tool would prove beneficial in those instances by making details (e.g., arm position) that otherwise might go unnoticed much clearer.

[Feature 2] - Dynamic annotations linked to the specified keypoint

The possibility of creating annotations such as drawings that would move accordingly to the player's motion when associated with a given keypoint (e.g., knee) was classified by all as useful regardless of the role they fulfilled in a team. However, perhaps due to the discrepancy between an athlete vs. a coach's perspective, overall, players viewed the application of this feature from the standpoint of receiving more comprehensible feedback from their respective coaching staff as opposed to using it themselves. On the other hand, four of the five coaches noted that such dynamic mechanisms would allow athletes to have a larger attention span when given observations instead of losing focus with their surroundings as often.

[Feature 3] - Individual pose estimation on neutral background

The participants posed several concerns regarding the possible limitations of this feature, mainly on the coach's side. All recognized its potential, however, three main comments were proposed on possible improvements to make the feature more complete.

1. Displaying two athletes at once in the same window.
2. Selecting which skeletal images to display on the video itself.
3. Observing all players in a frame with a neutral background.

Additionally, coaches also highlighted its pertinence when comparing players. Athletes also made this same observation where the unanimous view was that this feature was helpful in their role, for instance, by removing the visual noise from the video and allowing them to focus on their posture and use other players' poses as reference. Unsurprisingly, the importance of having the video present while analyzing the individual postures was emphasized by both coaches and players alike.

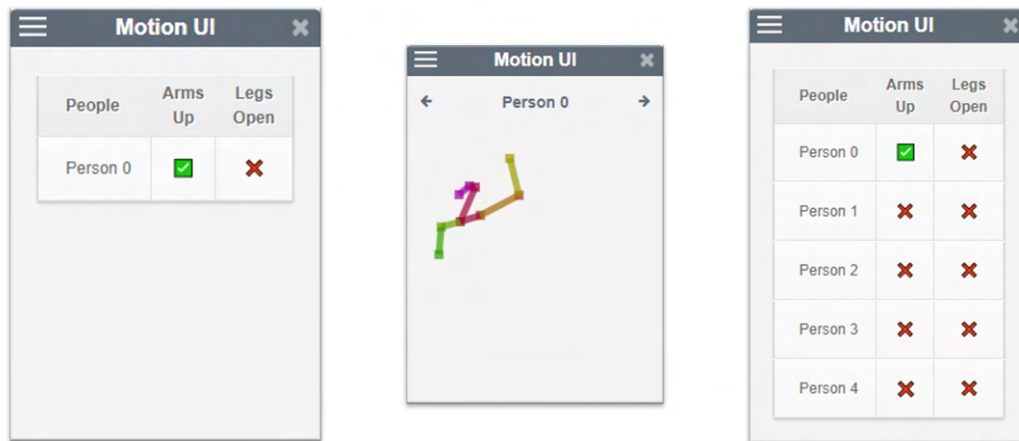


Figure 4.3: MotionUI images portraying the executed basketball shot motion (*Person 0*).

[Feature 4] - Automatic detection of technical gestures

All subjects demonstrated interest throughout the demonstration and hands-on experience in this interview section. "Coach 1" provided a practical use case where a player could observe if the gestures were identified and discuss it with the instructor afterwards to find possible improvements. Moreover, coaches also mentioned the helpfulness of self-coaching mechanisms that might derive from this feature by having athletes check if their movements matched a given motion pattern ideal model. One player even commented on how valuable it would be to understand which players were executing a gesture correctly (e.g., only one of ten players did not execute the "Legs Open" motion).

The research and developments made on the annotation tool by integrating pose estimation features using state-of-the-art technology are encouraging, given the preliminary results on the basketball sport. Furthermore, it is worth mentioning that four of the five interviewed coaches have added experience in the sports world, either as P.E. instructors or physiotherapists. In addition, more than half commented positively on the possibility of expanding the use of the demonstrated pose components to other sports. Particularly, 20% of the coaches mentioned the relevancy of the features applied to an individual sport such as tennis, since their technical gestures contribute directly to their competitive success without relying on other factors inherent to team sports (e.g., basketball's help defense strategy).

4.2 Final User Tests

This section covers the procedures used to plan and conduct the evaluation of the annotation system from an usability standpoint. It is important to note that the system itself offers a lot of possible interactions using previously implemented features such as the mentioned annotation types and using the embedded functionality to share the annotation work to others. As a result, despite focusing this final system evaluation on the newly implemented features - 3D annotation type and pose estimation -, it is important to note that questions regarding their integration with the formerly existing system will inevitably present some bias in relation to the system as whole. Moreover, the feedback received, derives from users with short experience with the annotation tool as opposed to others that might otherwise frequently use it.

This rationale deeply influenced the workshop structure that was created to be presented at an appointed Weave project session since the system was discussed in its totality. Therefore, since the scope of the presentation that preceded the actual hands-on section of the workshop focused on different aspects beyond the newly implemented features, the subsequent questions were created to be visibly divided. The objective was presenting the system usability scale questionnaire applied repeatedly to both developed components (3D-based annotations and pose estimation) in order to have a clear distinction between them while mitigating the importance of other pre-existing functionalities (appendix A). Unfortunately, the amount of feedback received was mostly verbal and while it did prove valuable to understand what users might feel when presented with the system from professionals in the area, the bias of seeing the system as whole hindered the statistics that might be drawn from their participation. Thus, a final workshop focused mainly in the 3D-based annotation and pose estimation features.

In this final user evaluation, the target of analysis was entirely centered around the 3D-based features and pose estimation functionalities following a short system introduction regarding basic annotation mechanisms. Initially, participants receive a brief demonstration of loading a video and creating simple drawing and text annotations over it. Afterwards, the user is asked to enter the system, play/pause a loaded video and begin interacting with 3D models: uploading, visualizing and interacting with a 3D object, changing backgrounds, as well as adding and customizing models as annotations. Finally, participants move on to the pose estimation section, where they are asked to interact with the different possible modalities, from base pose estimation visualization to dynamic annotations and movement recognition.

Following the hands-on section of the workshop, the volunteers engage with the questionnaire containing questions regarding the two types of features they had just interacted with. For each of the specific sets of questions relative to either 3D-based components or pose estimation, they are encouraged to offer suggestions and feedback regarding their experience with the tool.

A summary of the interview's structure is available in guide D.

4.2.1 Participants and Evaluation Method

A total of 30 users with ages from 18 to 52 years old (approximated mean: 26.17, approximated standard deviation: 7.78), engaged with the annotation tool with 18 being male and 12 females participants. Aside from two participants, none of the users had previous contact with the system and its functionalities. Additionally, the selected volunteers have different professional and academic backgrounds and all reported to frequently use web-browsers as well as having a reasonable comfort level with video annotation and editing (fig. 4.4).

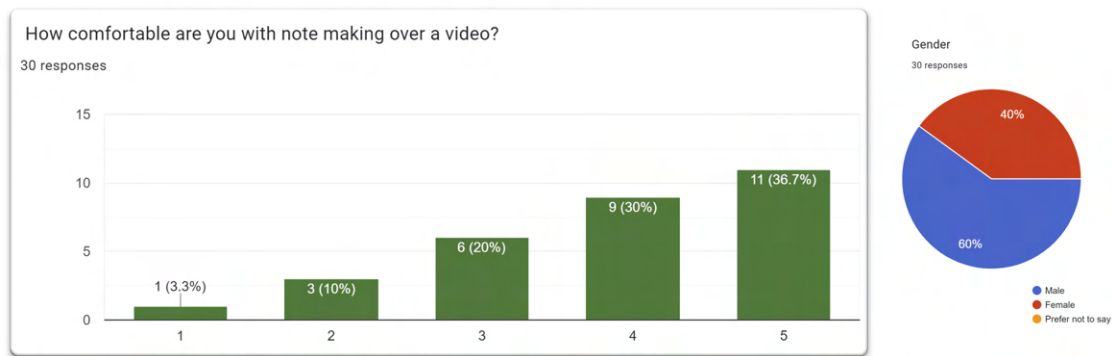


Figure 4.4: Answers for the comfort level using video annotations.

The structure of the questionnaire is divided into four sections (appendix F). Despite signing a consent form prior to the actual interview, the first part asks users for their consent and contains questions regarding participant's personal data (e.g., age and gender) as well as others to retrieve information about their technological experience with web-browsers. The second section contains standard SUS questions to draw statistics from an usability standpoint based on user feedback. Finally, the last two sections target the 3D-based features and pose estimation functionalities respectively to acquire a more detailed feedback from users.

4.2.2 Usability Scores

Initially presented by Brooke in 1995 [82], the system usability scale (SUS) is a category of questionnaires that requests the respondent to determine their degree of agreement with ten affirmations about their perception of the usability of a system. For each question, the respondents are solicited to choose between a five point Likert scale, each taking a value between 1 and 5. The range of numerical values from 1 to 5 directly corresponds to responses between "completely disagree with this statement" and "completely agree with this statement".

The SUS questionnaire is often used to obtain a straightforward assessment over a system's overall usability and thus usually not applied in a bisected manner, such as was intended to be in the WEAVE session's initial version. Nevertheless, users were briefed

with instructions to answer the system-related topics of the questionnaire without giving too much weight to the previously existing functionalities.

Calculating the system usability scores⁶ is done through the following formula:

$$((Q1 - 1) + (5 - Q2) + (Q3 - 1) + (5 - Q4) \dots + (Q9 - 1) + (5 - Q10)) * 2.5$$

In which, Q_i is the answer's average across all participants regarding question i ($i \in [1,10]$). The system usability scale results show that the developed system - related to both the 3D-based-annotations and pose estimation - had the following statistical values:

Mean	Median	Standard Deviation
83.6	85.0	9.37

Table 4.1: SUS scores.

Bangor et al. judges SUS scores using these metrics, describing scores below 50 as a "(...) *cause for significant concern*" as opposed to 100 which would be the best score imaginable. From the presented reasoning, these empirical values show that the user acceptance was very reasonable with a near excellent acceptability score. Moreover, some of the default SUS questions aim to infer the participant's perception in regards to the system's complexity and overall ease of use. These are crucial aspects considering the nature of this thesis as it pertains to the integration of new features into an existing intricate system. The volunteer's evaluation concerning these factors, as observed in figure 4.5, reveal that the efforts made towards a smooth integration of the developed features were perceived as rather successful.

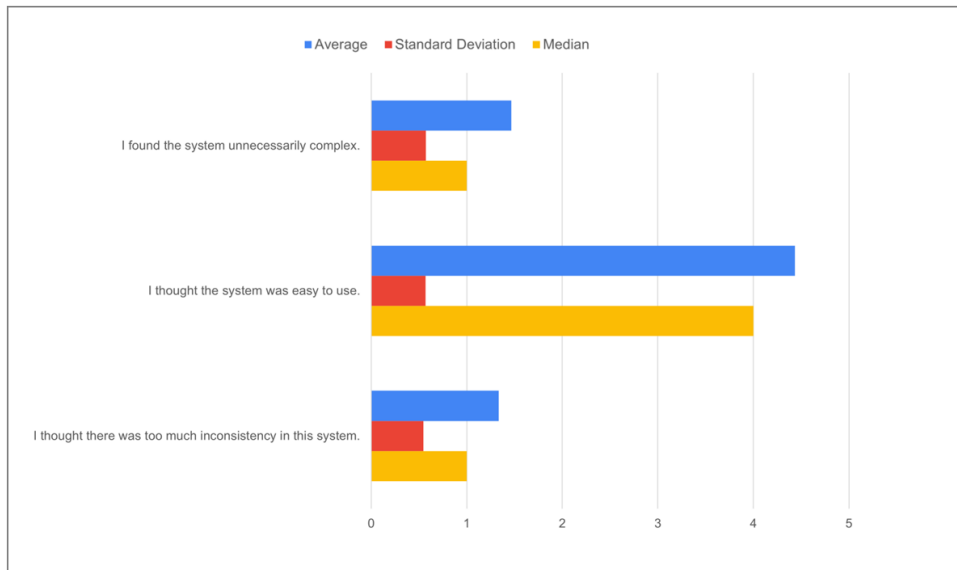


Figure 4.5: SUS questions regarding system complexity.

⁶Additional details can be found at [83].

From a different perspective, the questions in which the best possible value is '1' and the questions where '5' is the highest score had a mean and median value below two and above four, respectively. Thus, these values further indicate that the results regarding the system's usability were good.

4.2.3 Results and Discussion

The results of the final evaluation's questionnaire are summarized in figure 4.6. Aside from the standard SUS questions, the respondents were also prompted to answer more specific questions regarding each of the 3D and pose estimation features which are divided into their own two respective sections. To retrieve immediate feedback relative to how users perceived these two different modalities as an addition to the MotionNotes system, each section begins by questioning users about the value of integrating either 3D-based components or pose estimation. Similarly to the SUS standard set of questions, the devised questions can range between '1' or '5' depending on the volunteer's level of agreement with the presented statements.

For the 3D-based component to import, visualize and use 3D models as annotations, there are two distinguishable phases to add 3D annotations over the selected video. The first is mainly through the interactions with *3D Models Manager* interface, where 3D objects are initially imported with the possibility of pre-visualizing and interacting with them prior to having them behave as actual annotations. Following that, the selected model will be associated with its respective annotation bar on the 3D annotation track containing other annotation-specific data (e.g., timestamp, position, and size) as soon as they are added to the video. Considering this bisected nature in the annotation process using 3D objects, the section following the SUS questionnaire contains four questions for each of these two parts.

For the interface-related topics, this section approaches the following factors: user perception of the interface's intuitiveness, the action buttons, 3D interactions, and 360° backgrounds (statements 14, 15, 16, and 17, respectively). The purpose of these questions is to have more specific feedback for each of the presented features in order to understand what could be improved as well as how participants viewed each component. For instance, statements 14 and 16 reveal how users reportedly found the pre-visualizing interface and overall object interaction to be straightforward and intuitive. Consequently, despite all statements revealing averages very near the optimal value ('5'), the results for numbers 15 and 17 show a mean value of 4.1 and 4.8, which display the most significant distance regarding user acceptance for these four statements. Moreover, some suggestions reinforce this need for clearer action buttons, with one user noting, "The button to show the 3D model could be more intuitive". In contrast, the possibility of selecting between different 360° backgrounds as a very appealing feature was a common remark made by several users, which justifies the excellent results regarding this interface's functionality.

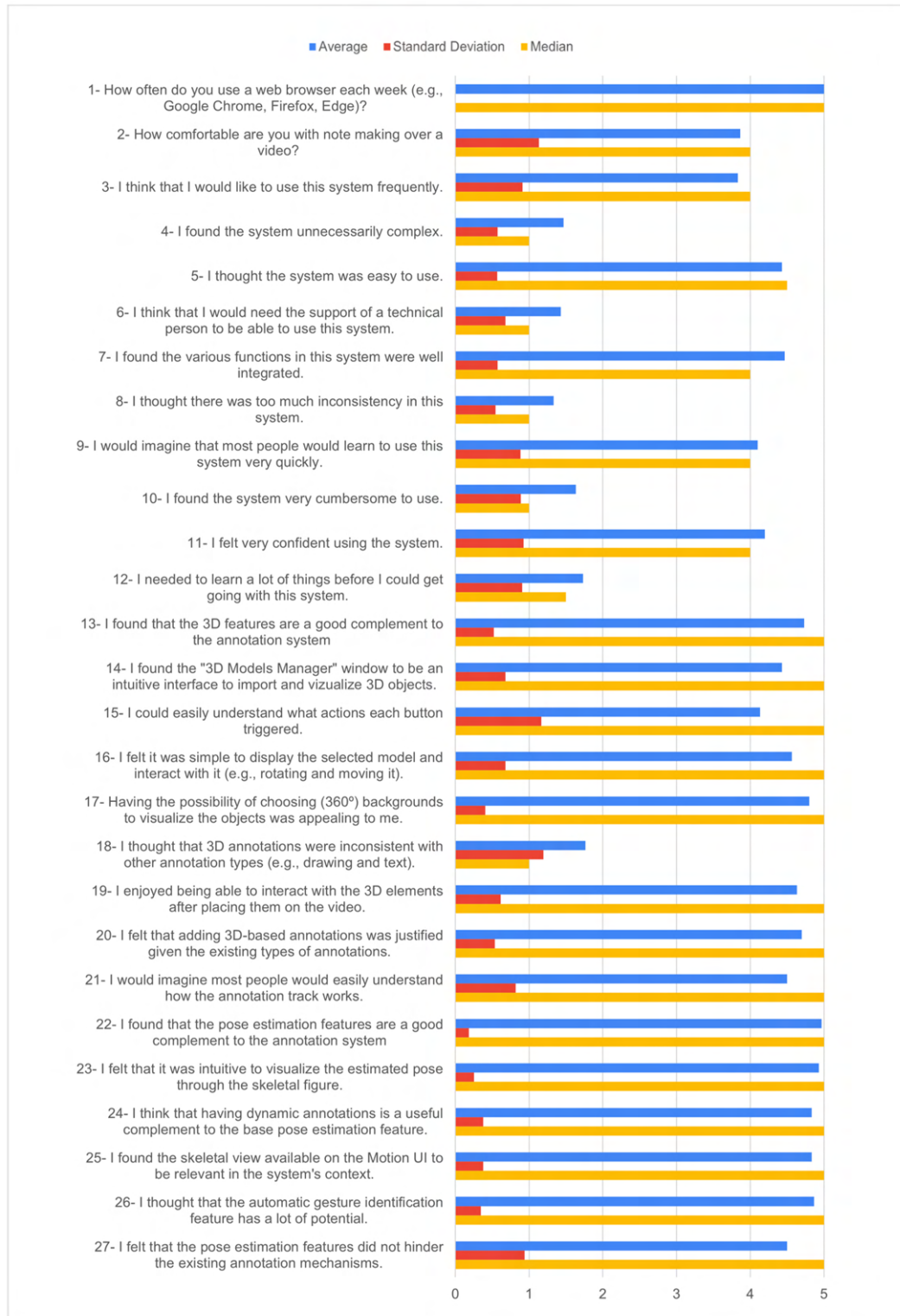


Figure 4.6: Final user tests results.

For the 3D-based annotation, the statements aim to provide user feedback regarding three key aspects: viability and consistency with previous annotation types, annotation tracks, and 3D interaction within video content (fig. 4.7). Since one of the goals of this thesis is the testing how viable 3D annotations are in the context of the MotionNotes annotation system, it is important to understand the participants' feelings regarding this integration. With that in mind, statements 18 and 20 are purposely redundant in order to properly grasp the volunteer's feelings regarding the validity of the 3D-based annotation functionality. Therefore, the positive results indicate that users found the 3D annotation to be successfully integrated into the system. Additionally, two other relevant factors inherent to creating 3D annotations are the ability to interact with objects while in the video, and their respective annotation timestamp, to which users attributed a positive evaluation (numbers 19 and 21).



Figure 4.7: Results from statements 18 to 21 - 3D annotation.

The last section of the questionnaire focuses on the pose estimation functionalities. Contrasting with the previous part's initial statement: "I found that the 3D features are a good complement to the annotation system", the pose estimation equivalent "I found that the pose estimation features are a good complement to the annotation system" received an overwhelming twenty-nine out of thirty answers scoring the maximum value '5 - Strongly Agree'. In a similar fashion to the 3D-specific section, there is also a statement aimed at detecting if users found pose estimation to hinder prior functionalities (fig. 4.8). Since the results for this statement's scores are rather positive, having an approximate average near the optimal value '1', there is further empirical evidence that suggests there was a successful integration the pose estimation components thus fulfilling the general goal of attempting to seamlessly integrate these functionalities into an existing system.

The following set of statements aim to cover the existing pose estimation functionalities, in short: pose estimation visualization, dynamic annotations and selective skeletal visualization as well as gesture recognition (Motion UI). The scores directly obtained from

these statements reveal an overall positive user satisfaction with the mean and median values scored above the '4' mark for all four statements (fig. 4.8 - left to right, respectively). Nevertheless, it is worth to note that despite the good results, users still commented on possible ideas and provided some useful suggestions, namely, the possibility of automatically generating statistics using a timeline of accurately executed movements for a given video.

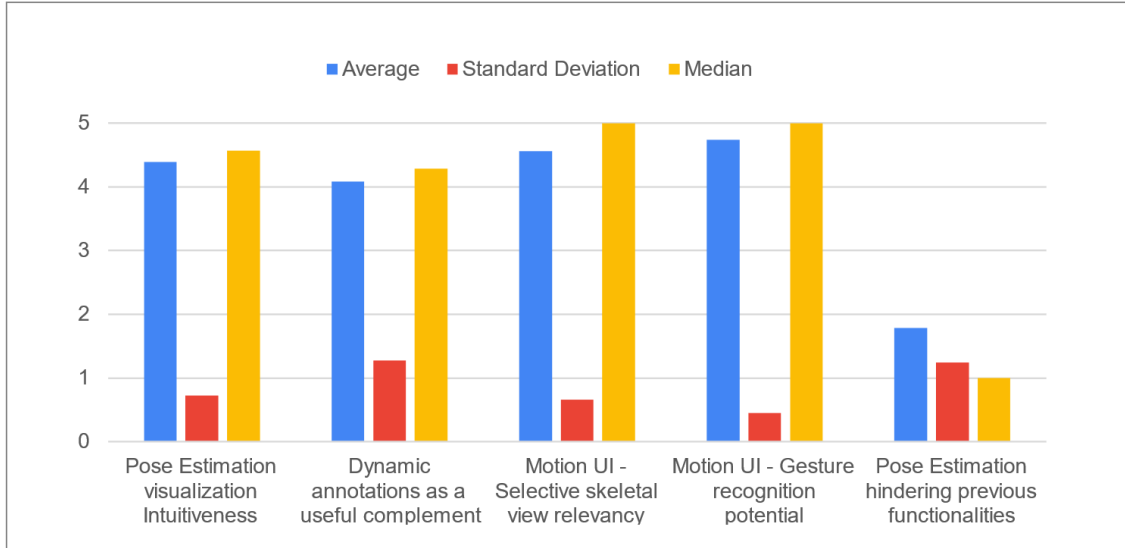


Figure 4.8: Results from statements 23 to 27 (left to right, respectively) - Pose Estimation.

Similarly to the SUS section, the results for these last two section show a positive acceptance rate, as most answers were in average very near to the best possible result - '1' or '5'. Still, there is a slight difference between the overall result averages between the 3D and pose estimation sections which suggests a higher acceptance of the latter's features in comparison to their 3D counterpart. Additionally, for each of these sections, participants were encouraged to add personal comments on their experience. Interestingly, the resulting optional suggestions mainly targeted the 3D components and not as much the pose estimation functionalities which seem to corroborate the user's preference towards these keypoint-based features. Some relevant impressions of the system are explicit below⁷:

- "Have the system inherently provide pre-defined shape models" (*3D Models Manager*).
- "The icon to add model as annotation might be misleading" (*3D Models Manager*).
- "The icon to view a model could be more intuitive" (*3D Models Manager*).
- "Adapting lighting conditions would be a nice complement to the interface" (*3D Models Manager*).

⁷Disclaimer: Some suggestions were altered using different wording to make them more readable.

- "Automatically generate movement reports" (*Motion UI*).
- "Make real-time pose estimation possible" (*Motion UI*).
- "Having the 3D objects animated according to a giving movement" (*Video*).

Given the emphasis made by different participants on these topics, several of the aforementioned items should definitely be considered for future developmental iterations of the MotionNotes system such as improving the icon choice for the action buttons and adding lighting conditions. Nevertheless, even though one user mentioned that the eye button could be more intuitive, it should not be considered critical as a large sample size of users were quickly able to trigger a selected model's visualization by either double-clicking the respective model's thumbnail or through the action button. It is understandable, however, that users might initially think the button to add annotations will load the model into the canvas just as another participant mentioned. As a result, despite posing a minor inconvenience at first it should still be counted as a target for improvement to avoid future hesitation in user's actions.

The presented functional requirements for displaying and interacting with 3D models in the context of an annotation system resulted in the creation of a pre-visualization interface for uploading, viewing and interacting with dimensional object before adding them to the video. Subsequently, to employ these models as annotation mechanisms, the system also integrates further interactive elements to the main MotionNotes interface (e.g., 3D annotation track, side trigger to add last selected 3D model). The relevancy of 3D-based objects in the web context, the usual means to obtain them as well as how they can be utilized in the target system is explored in detail from Chapter 1 to Chapter 2. For the evaluation phase, this project's partner Pé de Xumbo provided vital resources in regards to Portuguese cultural heritage by providing traditional dance content. Moreover, their contribution to the preliminary study regarding 3D-based annotations in Portuguese traditional dances also provided relevant insight into these features potential and inspired the creation of the first published paper at the *IMX* international conference.

Similarly, the pose estimation requirements to augment annotation work by automatically highlighting human pose and postures was fulfilled through the integration of the OpenPose library in order to extract keypoint data (e.g., shoulder and knee's position) belonging to people present in a given video. However, in order to make video analysis more complete when using pose estimation's skeletal imagery, base pose estimation visualization was expanded. Thus, besides the novelty of dynamic annotations to follow a specified keypoint during a video (e.g., elbow), there is also the addition of selective pose estimation and gesture recognition at a dedicated interface (*Motion UI*).

Regarding the pose estimation, both the verbal as well as written feedback indicate that users find it interesting to create mechanisms for report generation with the purpose of summarizing what tabled movements were executed throughout a video for each identified person. Interestingly, this comment was also discussed and quite commonly

brought up by participants at the Basketball case study in the context of sport's practice and competition. On another note, making pose estimation run in real-time, for instance, for live demonstrations and presentations is yet another attainable goal for future developments which a volunteer showed interest about. Lastly, one user suggested that pose estimation and 3D annotations could be directly connected through the creation of automatic movement animations for an object associated with a person's body part.

CONCLUSIONS AND FUTURE WORK

This final chapter presents the conclusions resulting from the developments made throughout this thesis and its evaluation while also discussing routes of possible work to improve the system in the future.

5.1 Conclusions

This thesis aims to explore the potential behind the integration of 3D models to enhance existing annotation mechanisms as well as how pose estimation-based features may present a viable component to augment annotation work by automatically highlighting people's poses and postures. Since these functionalities are integrated into an existing annotation system, another inherent goal of these developments is to make this integration as smooth as possible to avoid increasing unnecessary complexity and hindering overall usability. Due to the inherent nature of this thesis development within the context of the WEAVE international project, some of the work and design decisions result directly from targeted goals of connections between other partner tools (e.g., Arctur's 3D Weaver).

The presented functional requirements for displaying and interacting with 3D models in the context of an annotation system resulted in creating a pre-visualization interface for uploading, viewing, and interacting with tridimensional objects before adding them to the video. Subsequently, to employ these models as annotation mechanisms, the system also integrates further interactive elements to the main MotionNotes interface (e.g., 3D annotation track, side trigger to add last selected 3D model). The relevancy of 3D-based objects in the web context, the usual means to obtain them as well as how they can be utilized in the target system are explored in detail from Chapter 1 to Chapter 2 along with the pose estimation functionalities. For the evaluation phase, this project's partner Pé de Xumbo provided invaluable resources regarding Portuguese cultural heritage by providing traditional dance content. Moreover, their contribution to the preliminary study regarding 3D-based annotations in traditional Portuguese dances also provided relevant insight into these features' potential and inspired the development of the first

research article presented earlier.

Similarly, the pose estimation requirements to augment annotation work by automatically highlighting human pose and postures were fulfilled through the integration of the OpenPose library in order to extract keypoint data (e.g., shoulder and knee's position) belonging to people present in a given video. However, to make video analysis more complete when using pose estimation's skeletal imagery, base pose estimation visualization was expanded. Thus, besides the novelty of dynamic annotations to follow a specified keypoint during a video (e.g., elbow), there is also the addition of selective pose estimation and gesture recognition at a dedicated interface (Motion UI). These functionalities were then explored in the sports context, culminating in the creation of the second preliminary evaluation: a Basketball case study. Here, participants provided feedback for the presented features resulting in an overall positive evaluation from a usability standpoint further complemented by specific use cases for their ubiquitous basketball environments. Therefore, the pose estimation features, interview structure, and applicability in the Basketball context were transcribed into the second accepted poster at the *MUM* Conference.

The results received from the evaluation process, from preliminary to final user tests, provide empirical evidence regarding the successful integration of the two modalities of features: 3D objects and annotation and pose estimation and tracking. Despite leaving some room for improvements in the existing system as well as its respective functionalities, the targeted developmental goals were entirely fulfilled, and new future challenges can now be explored.

5.2 Future Work

The future of the annotation tool beyond the existing functionalities can be approached through different perspectives. From a general point of view, the overall system visual representation is a possible target for improvements since, especially in the *HCI* research area, it is important to properly invest in an application's visual structure as it has a direct effect over users ability to interact with the system. Conversely, when focusing on this project's explored features (3D elements and pose estimation), there are several distinct ways conceivable of being further developed. On the one hand, there is room to grow regarding the possible 3D components since there will certainly be more frameworks like Threejs besides its own progression through community feedback and independent work. Although 3D models can be interacted with through simple translation, rotation, or scaling, creating adaptable lighting conditions as well as possibly modifying object characteristics (e.g., color and mesh structure) are possible paths of development.

Moreover, the use of both 360° videos and entirely virtual scenarios is interesting in the context of this annotation tool and was incentivized by performing arts specialists in the first preliminary study (Chapter 4). Similarly, in the final user evaluation, participants suggested that the interface to pre-visualize and interact with 3D models could

be somewhat modified, for instance, by making the action buttons more intuitive. Along with these adjustments, adding a search component to the interface for sorting through available 3D objects is also worth exploring.

On the pose estimation side, some immediate concerns must be addressed in the future, namely the computation time associated with state-of-the-art pose estimation models such as in the integrated OpenPose library. Fortunately, in the future, this issue will be mitigated due to the upgrade to the application's server capabilities on the graphics card side and since the pose estimation is rather new in the computer vision area, there is room for improvement in upcoming releases as well as in other future alternatives. Accordingly, one participant at the final evaluation even commented on how advantageous having real-time pose estimation can be in self-teaching environments.

The overwhelmingly positive feedback received in the usability tests reveals the potential behind the MotionNotes system and its features even when applied across different fields of work. For instance, as it pertains to the basketball case study, participants showed enthusiasm regarding the possibility of expanding pose estimation and tracking functionalities beyond their primary sport (e.g., football, volleyball, handball, and gymnastics). These comments are of significant relevancy as most of the interviewed coaches also work professionally as PE instructors. Here, participants expanded the discussion towards having automatically generated reports for the executed movements throughout a video, which proved to be a common remark made both in this study and final workshop and thus should also be considered as a possible route of development. Additionally, expanding the rule-based movements for the gesture component of pose estimation is another viable developmental option, for instance, by including hands keypoint data to be applied in fields of work such as ceramics and carpentry.

Lastly, submitting a scientific paper (full paper) describing the work conducted for the pose estimation functionalities and its application to the performing arts world is currently in progress.

BIBLIOGRAPHY

- [1] J. M. Lourenço. *The NOVAthesis L^AT_EX Template User's Manual*. NOVA University Lisbon. 2021. URL: <https://github.com/joaomlourenco/novathesis/raw/master/template.pdf> (cit. on p. vii).
- [2] E. S. Veinott et al. "Video matters! When communication ability is stressed, video helps". In: *Conference on Human Factors in Computing Systems - Proceedings 22-27-March-1997* (Mar. 1997), pp. 315–316. DOI: 10.1145/1120212.1120411 (cit. on p. 2).
- [3] V. V. Wassenhove, K. W. Grant, and D. Poeppel. "Visual speech speeds up the neural processing of auditory speech". In: *Proceedings of the National Academy of Sciences of the United States of America* 102 (4 Jan. 2005), pp. 1181–1186. ISSN: 00278424. DOI: 10.1073/PNAS.0408949102. URL: https://www.researchgate.net/publication/8084799_Visual_speech_speeds_up_the_neural_processing_of_auditory_speech (cit. on p. 2).
- [4] K. Ito et al. "Development of pen-based note-taking system for blind people". In: *Second International Conference on Innovative Computing, Information and Control, ICICIC 2007* (2007). DOI: 10.1109/ICICIC.2007.263 (cit. on pp. 2, 32).
- [5] D. S. Hayden et al. "Note-taker 2.0: The next step toward enabling students who are legally blind to take notes in class". In: *ASSETS'10 - Proceedings of the 12th International ACM SIGACCESS Conference on Computers and Accessibility* (2010), pp. 131–137. DOI: 10.1145/1878803.1878828 (cit. on pp. 2, 32).
- [6] T. Baji. "Evolution of the GPU Device widely used in AI and Massive Parallel Processing". In: *2018 IEEE Electron Devices Technology and Manufacturing Conference, EDTM 2018 - Proceedings* (July 2018), pp. 7–9. DOI: 10.1109/EDTM.2018.8421507 (cit. on pp. 2, 13).
- [7] P. Grocott et al. "Clinical note-making and patient outcome measures using TELER®". In: *Wounds International* 2 (3 Oct. 2011) (cit. on pp. 7, 32).

- [8] “Learning from YouTube: An analysis of information literacy in user discourse”. In: *ACM International Conference Proceeding Series* (2011), pp. 640–642. DOI: 10.1145/1940761.1940851. URL: <http://www.clex.org.uk> (cit. on p. 8).
- [9] R. Pea, R. Lindgren, and J. Rosen. “Computer-Supported Collaborative Video Analysis”. In: *Proceedings of the 7th International Conference on Learning Sciences. ICLS '06*. Bloomington, Indiana: International Society of the Learning Sciences, 2006, pp. 516–521. ISBN: 0805861742 (cit. on p. 8).
- [10] B. Xiao et al. “Modeling Multimodal Integration Patterns and Performance in Seniors: Toward Adaptive Processing of Individual Differences”. In: *Proceedings of the 5th international conference on Multimodal interfaces - ICMI '03* (2003). DOI: 10.1145/958432 (cit. on p. 8).
- [11] “Perceptual user interfaces: multimodal interfaces that process what comes naturally”. In: *Communications of the ACM* 43 (3 Mar. 2000), pp. 45–53. ISSN: 0001-0782. DOI: 10.1145/330534.330538. URL: <https://dl.acm.org/doi/abs/10.1145/330534.330538> (cit. on p. 8).
- [12] V. V. Wassenhove, K. W. Grant, and D. Poeppel. “Visual speech speeds up the neural processing of auditory speech”. In: *Proceedings of the National Academy of Sciences of the United States of America* 102 (4 Jan. 2005), pp. 1181–1186. ISSN: 00278424. DOI: 10.1073/PNAS.0408949102. URL: https://www.researchgate.net/publication/8084799_Visual_speech_speeds_up_the_neural_processing_of_auditory_speech (cit. on p. 8).
- [13] B. Dumas, D. Lalanne, and S. Oviatt. “Multimodal Interfaces: A Survey of Principles, Models and Frameworks”. In: *Lecture Notes in Computer Science (including sub-series Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5440 LNCS (2009), pp. 3–26. ISSN: 03029743. DOI: 10.1007/978-3-642-00437-7_1. URL: https://link.springer.com/chapter/10.1007/978-3-642-00437-7_1 (cit. on p. 8).
- [14] M. Turk. “Review Article”. In: *Pattern Recognition Letters* 36 (1 Jan. 2014), pp. 189–195. ISSN: 01678655. DOI: 10.1016/J.PATREC.2013.07.003. URL: <https://dl.acm.org/doi/abs/10.1016/j.patrec.2013.07.003> (cit. on p. 8).
- [15] R. Prasad. “Alexa Everywhere: AI for Daily Convenience”. In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* 19 (2019). DOI: 10.1145/3289600. URL: <https://doi.org/10.1145/3289600.3291377> (cit. on p. 8).
- [16] J. C. Kim, T. H. Laine, and C. Åhlund. “Multimodal Interaction Systems Based on Internet of Things and Augmented Reality: A Systematic Literature Review”. In: *Applied Sciences* 11 (4 Feb. 2021), pp. 1–33. ISSN: 20763417. DOI: 10.3390/APP11041738 (cit. on p. 8).

-
- [17] S. Oviatt. “Ten myths of multimodal interaction”. In: *Communications of the ACM* 42 (11 Nov. 1999), pp. 74–81. ISSN: 00010782. DOI: 10.1145/319382.319398. URL: <https://dl.acm.org/doi/abs/10.1145/319382.319398> (cit. on p. 8).
 - [18] Rui Rodrigues, Rui Neves Madeira and N. Correia. “Studying Natural User Interfaces for Smart Video Annotation towards Ubiquitous Environments”. In: *20th International Conference on Mobile and Ubiquitous Multimedia (MUM 2021)*. MUM 2021. Leuven, Belgium: ACM, New York, NY, USA, 2021, pp. 1–18 (cit. on p. 9).
 - [19] N. C. C. Lam and H. Habil. “The Use of Video Annotation in Education: A Review”. In: *Asian Journal of University Education* 17 (4 Nov. 2021), p. 84. ISSN: 1823-7797. DOI: 10.24191/AJUE.V17I4.16208 (cit. on pp. 9, 32).
 - [20] J. Grudin and D. Barger. “Multimedia Annotation: An Unsuccessful Tool Becomes a Successful Framework”. In: (2017). URL: <https://www.microsoft.com/en-us/research/wp-content/uploads/2017/01/evolutionchapter.pdf> (cit. on pp. 9, 11, 32).
 - [21] “The collaborative lecture annotation system (CLAS): A new TOOL for distributed learning”. In: *IEEE Transactions on Learning Technologies* 6 (1 2013), pp. 4–13. ISSN: 19391382. DOI: 10.1109/TLT.2012.15 (cit. on pp. 9, 32).
 - [22] “Identifying learning strategies associated with active use of video annotation software”. In: *ACM International Conference Proceeding Series* 16-20-March-2015 (Mar. 2015), pp. 255–259. DOI: 10.1145/2723576.2723611. URL: <http://dx.doi.org/10.1145/2723576.2723611> (cit. on p. 9).
 - [23] A. Pless et al. “Using self and peer video annotations of simulated patient encounters in communication training to facilitate the reflection of communication skills: An implementation study”. In: *GMS Journal for Medical Education* 38 (3 2021). ISSN: 23665017. DOI: 10.3205/ZMA001451 (cit. on p. 10).
 - [24] M. Bakopoulos et al. “Mobile video annotation for enhanced rich media communication during emergency handling”. In: *ACM International Conference Proceeding Series* (2011). DOI: 10.1145/2093698.2093730 (cit. on pp. 10–12).
 - [25] V. Singh et al. “The Choreographer’s Notebook-A Video Annotation System for Dancers and Choreographers”. In: *Proceedings of the 8th ACM conference on Creativity and cognition - CC ’11* (2011). DOI: 10.1145/2069618 (cit. on pp. 10, 11).
 - [26] D. Gašević, N. Mirriahi, and S. Dawson. “Analytics of the effects of video use and instruction to support reflective learning”. In: *ACM International Conference Proceeding Series* (2014), pp. 123–132. DOI: 10.1145/2567574.2567590. URL: <http://dx.doi.org/10.1145/2567574.2567590> (cit. on p. 10).
 - [27] D. Geer. “Vendors Upgrade Their Physics Processing to Improve Gaming”. In: *Computer* 39.8 (2006), pp. 22–24. DOI: 10.1109/MC.2006.284 (cit. on p. 13).

- [28] “Gaming Graphics: The Road to Revolution”. In: *Queue* 2 (2 Apr. 2004), pp. 62–71. ISSN: 15427749. DOI: 10.1145/988392.988409. URL: <https://dl.acm.org/doi/abs/10.1145/988392.988409> (cit. on p. 13).
- [29] A. Jones et al. “The effects of virtual reality, augmented reality, and motion parallax on egocentric depth perception”. In: *Proceedings - IEEE Virtual Reality* (2008), pp. 267–268. DOI: 10.1109/VR.2008.4480794 (cit. on p. 14).
- [30] M. Krichenbauer et al. “Augmented Reality versus Virtual Reality for 3D Object Manipulation”. In: *IEEE Transactions on Visualization and Computer Graphics* 24 (2 Feb. 2018), pp. 1038–1048. ISSN: 10772626. DOI: 10.1109/TVCG.2017.2658570 (cit. on p. 14).
- [31] L. Ma. “Application of AR in 3D Model”. In: CCRIS’21. Qingdao, China: Association for Computing Machinery, 2021, pp. 261–265. ISBN: 9781450390453. DOI: 10.1145/3483845.3483891. URL: <https://doi.org/10.1145/3483845.3483891> (cit. on p. 14).
- [32] “Crossmedia integration of 3D contents for cultural communication”. In: *3DTV-Conference* (2014). ISSN: 2161203X. DOI: 10.1109/3DTV.2014.6874732 (cit. on p. 14).
- [33] H. Southall, L. Beever, and P. Butcher. “Traversing social networks in the virtual dance hall: Visualizing history in VR”. In: *Proceedings - 2017 International Conference on Cyberworlds, CW 2017 - in cooperation with: Eurographics Association International Federation for Information Processing ACM SIGGRAPH 2017-January* (Nov. 2017), pp. 249–252. DOI: 10.1109/CW.2017.48 (cit. on p. 14).
- [34] “A Virtual Exhibition on the History of Hungarian Ballet”. In: *10th IEEE International Conference on Cognitive Infocommunications, CogInfoCom 2019 - Proceedings* (Oct. 2019), pp. 431–432. DOI: 10.1109/COGINFocom47531.2019.9089890 (cit. on p. 14).
- [35] P. Pellegretti et al. “A clinical experience of a prototype automated breast ultrasound system combining transmission and reflection 3D imaging”. In: *IEEE International Ultrasonics Symposium, IUS* (2011), pp. 1407–1410. ISSN: 19485719. DOI: 10.1109/ULTSYM.2011.0348 (cit. on p. 14).
- [36] K. M. Lewis et al. “Fast Learning-based Registration of Sparse 3D Clinical Images”. In: *Proceedings of the ACM Conference on Health, Inference, and Learning* (2020). DOI: 10.1145/3368555. URL: <http://voxelmorph.mit.edu/>. (cit. on p. 14).
- [37] F. Wang and T. Chen. “Constructing a 3D Virtual World for Traditional Culture Education”. In: *ACM International Conference Proceeding Series* (Feb. 2020), pp. 149–152. DOI: 10.1145/3383923.3383956. URL: <https://doi.org/10.1145/3383923.3383956> (cit. on p. 14).

-
- [38] O. Ha and N. Fang. "Development of interactive 3D tangible models as teaching aids to improve students' spatial ability in STEM education". In: *Proceedings - Frontiers in Education Conference, FIE* (2013), pp. 1302–1304. ISSN: 15394565. DOI: 10.1109/FIE.2013.6685043 (cit. on p. 14).
 - [39] G. Ruan et al. "High performance photogrammetry for academic research". In: *ACM International Conference Proceeding Series* (July 2018). DOI: 10.1145/3219104.3219148. URL: <https://doi.org/10.1145/3219104.3219148> (cit. on p. 15).
 - [40] D. Ionescu et al. "An infrared-based depth camera for gesture-based control of virtual environments". In: *2013 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications, CIVEMSA 2013 - Proceedings* (2013), pp. 13–18. DOI: 10.1109/CIVEMSA.2013.6617388 (cit. on p. 15).
 - [41] J. Ye and K. A. Hua. "Exploiting depth camera for 3D spatial relationship interpretation". In: *Proceedings of the 4th ACM Multimedia Systems Conference, MMSys 2013* (2013), pp. 151–161. DOI: 10.1145/2483977.2483998 (cit. on p. 15).
 - [42] A. Kanezaki and T. Harada. "3D Selective Search for obtaining object candidates". In: *IEEE International Conference on Intelligent Robots and Systems 2015-December* (Dec. 2015), pp. 82–87. ISSN: 21530866. DOI: 10.1109/IRoS.2015.7353358 (cit. on p. 15).
 - [43] C. Zhang et al. "MeshStereo: A Global Stereo Model with Mesh Alignment Regularization for View Interpolation". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 2057–2065. DOI: 10.1109/ICCV.2015.238 (cit. on p. 16).
 - [44] H. Lim et al. "Putting real-world objects into virtual world: Fast automatic creation of animatable 3D models with a consumer depth camera". In: *Proceedings - 2012 International Symposium on Ubiquitous Virtual Reality, ISUVR 2012* (2012), pp. 38–41. DOI: 10.1109/ISUVR.2012.12 (cit. on p. 16).
 - [45] H. Lim, J. Kang, and S. C. Ahn. "Rapid 3D avatar creation system using a single depth camera". In: *26th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2019 - Proceedings* (Mar. 2019), pp. 1329–1330. DOI: 10.1109/VR.2019.8798353 (cit. on p. 16).
 - [46] S. Ding et al. "MBD Based 3D CAD Model Automatic Feature Recognition and Similarity Evaluation". In: *IEEE Access* 9 (2021), pp. 150403–150425. ISSN: 21693536. DOI: 10.1109/ACCESS.2021.3126333 (cit. on p. 16).
 - [47] D. Bartonek and M. Buday. "Problems of Creation and Usage of 3D Model of Structures and Theirs Possible Solution". In: *Symmetry* 2020, Vol. 12, Page 181 12 (1 Jan. 2020), p. 181. ISSN: 20738994. DOI: 10.3390/SYM12010181. URL: <https://www.mdpi.com/2073-8994/12/1/181/htm%20https://www.mdpi.com/2073-8994/12/1/181> (cit. on p. 16).

- [48] Y. Zhang et al. “3D CAD Modeling and Visualization of the Tunnel Construction Process in a Distributed Simulation Environment”. In: *Proceedings of the Winter Simulation Conference*. WSC '10. Baltimore, Maryland: Winter Simulation Conference, 2010, pp. 3189–3200. ISBN: 9781424498642 (cit. on p. 16).
- [49] A. Guidazzoli et al. “Blender: A Framework for Cross-Media Cultural Heritage Applications”. In: *Proceedings of the 2016 Virtual Reality International Conference*. VRIC '16. Laval, France: Association for Computing Machinery, 2016. ISBN: 9781450341806. DOI: 10.1145/2927929.2927958. URL: <https://doi.org/10.1145/2927929.2927958> (cit. on p. 17).
- [50] C. Peng. “The research and design of 3D Web guide system based on WebGL”. In: *26th Chinese Control and Decision Conference, CCDC 2014* (2014), pp. 4052–4054. DOI: 10.1109/CCDC.2014.6852890 (cit. on p. 18).
- [51] E. Angel and E. Haines. “An interactive introduction to webgl and three.js”. In: *ACM SIGGRAPH 2017 Courses, SIGGRAPH 2017* (July 2017). DOI: 10.1145/3084873.3084875. URL: <http://webglstats.com/>. (cit. on p. 18).
- [52] U. Dey, P. K. Jana, and C. S. Kumar. “Modeling and Kinematic Analysis of Industrial Robots in WebGL Interface”. In: *Proceedings - IEEE 8th International Conference on Technology for Education, T4E 2016* (Jan. 2017), pp. 256–257. DOI: 10.1109/T4E.2016.067 (cit. on pp. 19, 20).
- [53] M. Li, C. Li, and M. Shi. “Movie Data Visualization Based on WebGL”. In: *Proceedings - 2021 21st ACIS International Semi-Virtual Winter Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD-Winter 2021* (Jan. 2021), pp. 74–78. DOI: 10.1109/SNPDWINTER52325.2021.00023 (cit. on p. 20).
- [54] R. Miao, J. Song, and Y. Zhu. “3D geographic scenes visualization based on WebGL”. In: *2017 6th International Conference on Agro-Geoinformatics, Agro-Geoinformatics 2017* (Sept. 2017). DOI: 10.1109/AGRO-GEOINFORMATICS.2017.8046999 (cit. on p. 20).
- [55] N. Eamnapha, S. Nuratch, and W. Lenwari. “The graphics and physics engines for rapid development of 3d web-based applications”. In: *Proceedings of the 16th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2019* (July 2019), pp. 89–92. DOI: 10.1109/ECTI-CON47248.2019.8955392 (cit. on p. 21).
- [56] R. Rodrigues et al. “Multimodal Web Based Video Annotator with Real-Time Human Pose Estimation”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11872 LNCS (Nov. 2019), pp. 23–30. ISSN: 16113349. DOI: 10.1007/978-3-030-33617-2_3. URL: https://link.springer.com/chapter/10.1007/978-3-030-33617-2_3 (cit. on pp. 21, 31).

- [57] A. Yamakawa, T. Ishikawa, and H. Watanabe. “Study on Improvement of Estimation Accuracy in Pose Estimation Model Using Time Series Correlation”. In: *2020 IEEE 9th Global Conference on Consumer Electronics, GCCE 2020* (Oct. 2020), pp. 409–412. DOI: 10.1109/GCCE50665.2020.9291962 (cit. on p. 22).
- [58] S. Feng et al. “Learning Joint Structure for Human Pose Estimation”. In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 16 (3 July 2020), p. 85. ISSN: 15516865. DOI: 10.1145/3392302. URL: <https://dl.acm.org/doi/abs/10.1145/3392302> (cit. on p. 22).
- [59] Y. Chen et al. “Cascaded Pyramid Network for Multi-person Pose Estimation”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Dec. 2018), pp. 7103–7112. ISSN: 10636919. DOI: 10.1109/CVPR.2018.00742 (cit. on p. 23).
- [60] K. Sun et al. “Deep high-resolution representation learning for human pose estimation”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2019-June* (June 2019), pp. 5686–5696. ISSN: 10636919. DOI: 10.1109/CVPR.2019.00584 (cit. on p. 23).
- [61] D. Shi et al. “InsPose: Instance-Aware Networks for Single-Stage Multi-Person Pose Estimation”. In: *Proceedings of the 29th ACM International Conference on Multimedia*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 3079–3087. ISBN: 9781450386517. URL: <https://doi.org/10.1145/3474085.3475447> (cit. on pp. 23, 24).
- [62] X. Nie et al. “Single-stage multi-person pose machines”. In: *Proceedings of the IEEE International Conference on Computer Vision 2019-October* (Oct. 2019), pp. 6950–6959. ISSN: 15505499. DOI: 10.1109/ICCV.2019.00705 (cit. on pp. 23, 24).
- [63] Z. Cao et al. “OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43 (1 Jan. 2021), pp. 172–186. ISSN: 19393539. DOI: 10.1109/TPAMI.2019.2929257 (cit. on pp. 23, 25).
- [64] T. L. Munea et al. “The Progress of Human Pose Estimation: A Survey and Taxonomy of Models Applied in 2D Human Pose Estimation”. In: *IEEE Access* 8 (2020), pp. 133330–133348. ISSN: 21693536. DOI: 10.1109/ACCESS.2020.3010248 (cit. on pp. 23, 24).
- [65] Y.-C. Li et al. “Baseball Swing Pose Estimation Using OpenPose”. In: (Aug. 2021), pp. 6–9. DOI: 10.1109/RAAI52226.2021.9507807 (cit. on pp. 23, 25).
- [66] S. Chang et al. “Towards Accurate Human Pose Estimation in Videos of Crowded Scenes”. In: *MM 2020 - Proceedings of the 28th ACM International Conference on Multimedia* 5 (20 Oct. 2020), pp. 4630–4634. DOI: 10.1145/3394171.3416299. URL: <https://doi.org/10.1145/3394171.3416299> (cit. on p. 23).

- [67] “DeepPose: Human pose estimation via deep neural networks”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Sept. 2014), pp. 1653–1660. ISSN: 10636919. DOI: 10.1109/CVPR.2014.214 (cit. on p. 24).
- [68] J. Han et al. “Enhanced computer vision with Microsoft Kinect sensor: A review”. In: *IEEE Transactions on Cybernetics* 43 (5 Oct. 2013), pp. 1318–1334. ISSN: 21682267. DOI: 10.1109/TCYB.2013.2265378 (cit. on p. 24).
- [69] T. Okumura et al. “Cooking activities recognition in egocentric videos using hand shape feature with OpenPose”. In: *ACM International Conference Proceeding Series* 18 (July 2018), pp. 42–45. DOI: 10.1145/3230519.3230591. URL: <https://doi.org/10.1145/3230519.3230591> (cit. on p. 25).
- [70] B. Lin et al. “Bradykinesia Recognition in Parkinson’s Disease via Single RGB Video”. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 14 (2 Feb. 2020). ISSN: 1556472X. DOI: 10.1145/3369438. URL: <https://dl.acm.org/doi/abs/10.1145/3369438> (cit. on pp. 25, 26).
- [71] K. Yamao and R. Kubota. “Development of Human Pose Recognition System by Using Raspberry Pi and PoseNet Model”. In: *Proceedings of ISCIT 2021: 2021 20th International Symposium on Communications and Information Technologies: Quest for Quality of Life and Smart City* (Oct. 2021), pp. 41–44. DOI: 10.1109/ISCIT52804.2021.9590593 (cit. on pp. 25, 26).
- [72] J. Zhang et al. “MobiPose: Real-time multi-person pose estimation on mobile devices”. In: *SenSys 2020 - Proceedings of the 2020 18th ACM Conference on Embedded Networked Sensor Systems* (Nov. 2020), pp. 136–149. DOI: 10.1145/3384419.3430726. URL: <https://doi.org/10.1145/3384419.3430726> (cit. on pp. 25, 26).
- [73] *Track human poses in real-time on Android with TensorFlow Lite — The TensorFlow Blog*. URL: <https://blog.tensorflow.org/2019/08/track-human-poses-in-real-time-on-android-tensorflow-lite.html> (cit. on p. 25).
- [74] T. Zhou and Y. Liu. “Long-Term Person Tracking for Unmanned Aerial Vehicle Based on Human-Machine Collaboration”. In: *IEEE Access* 9 (2021), pp. 161181–161193. DOI: 10.1109/ACCESS.2021.3132077 (cit. on p. 26).
- [75] S. Fang, S. Munir, and S. Nirjon. “Person Tracking and Identification Using Cameras and Wi-Fi Channel State Information (CSI) from Smartphones: Dataset”. In: *Proceedings of the Third Workshop on Data: Acquisition To Analysis. DATA ’20*. Virtual Event, Japan: Association for Computing Machinery, 2020, pp. 26–30. ISBN: 9781450381369. DOI: 10.1145/3419016.3431488. URL: <https://doi.org/10.1145/3419016.3431488> (cit. on p. 26).

- [76] F. Fang et al. "Real-time RGB-D based people detection and tracking system for mobile robots". In: *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*. 2017, pp. 1937–1941. DOI: 10.1109/ICMA.2017.8016114 (cit. on p. 26).
- [77] R. Ranftl et al. "Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.3 (2022), pp. 1623–1637. DOI: 10.1109/TPAMI.2020.3019967 (cit. on p. 26).
- [78] S. M. H. Miangoleh et al. "Boosting Monocular Depth Estimation Models to High-Resolution via Content-Adaptive Multi-Resolution Merging". In: 2021 (cit. on p. 26).
- [79] D. Cabral et al. "Evaluation of a Multimodal Video Annotator for Contemporary Dance". In: *Proceedings of the International Working Conference on Advanced Visual Interfaces*. AVI '12. Capri Island, Italy: Association for Computing Machinery, 2012, pp. 572–579. ISBN: 9781450312875. DOI: 10.1145/2254556.2254663. URL: <https://doi.org/10.1145/2254556.2254663> (cit. on p. 31).
- [80] R. Rodrigues et al. "Integrating 3D Objects in Multimodal Video Annotation". In: *ACM International Conference on Interactive Media Experiences*. IMX '22. Aveiro, JB, Portugal: Association for Computing Machinery, 2022, pp. 299–304. ISBN: 9781450392129. DOI: 10.1145/3505284.3532972. URL: <https://doi.org/10.1145/3505284.3532972> (cit. on p. 61).
- [81] W. Zhiwen et al. "Analysis of Influencing Factors of Shooting Rate Based on Trajectory Prediction of the Basketball". In: *2017 14th Web Information Systems and Applications Conference (WISA)*. 2017, pp. 176–180. DOI: 10.1109/WISA.2017.18 (cit. on p. 62).
- [82] J. B. Brooke. "SUS: A 'Quick and Dirty' Usability Scale". In: 1996 (cit. on p. 67).
- [83] "An Empirical Evaluation of the System Usability Scale". In: 24.6 (Aug. 2008), pp. 574–594. ISSN: 10447318. DOI: 10.1080/10447310802205776. URL: <https://doi.org/10.1080/10447310802205776> (cit. on p. 68).

| A

USABILITY QUESTIONNAIRE *WEAVE*
ONLINE

Motion Notes Questionnaire (WEAVE)

The following set of questions aims to gather user feedback using the Motion Notes annotation system in the context of the [WEAVE](#) European project. Firstly, you will interact with the recent contributions to the system, namely, the implemented 3D and pose estimation features based on the material provided. Following that, you may begin to answer the presented questionnaire.

Feel free to ask questions to any of Motion Notes collaborators.

***Required**

1. I accept the terms presented to me in the usability guide in order to participate in this workshop and answer the following questions. *

Note: Your participation must be voluntary. Refusing to participate in these tasks will not cause you harm or jeopardize you in any manner. By agreeing to participate in this study, you are granting permission to use its results anonymously for academic use, such as in oral class presentations or others, thereby contributing to the scientific community.

Mark only one oval.

☐ Yes

2. Age *

3. Gender *

Mark only one oval.

☐ Male

☐ Female

☐ Prefer not to say

4. How often do you use a web browser (e.g., Google Chrome, Firefox, Edge)?

Mark only one oval.

	1	2	3	4	5	
Almost Never	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Almost Always

5. How comfortable are you with video note making?

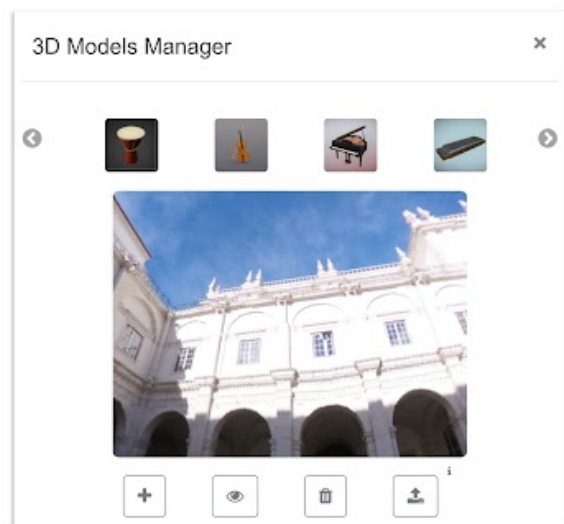
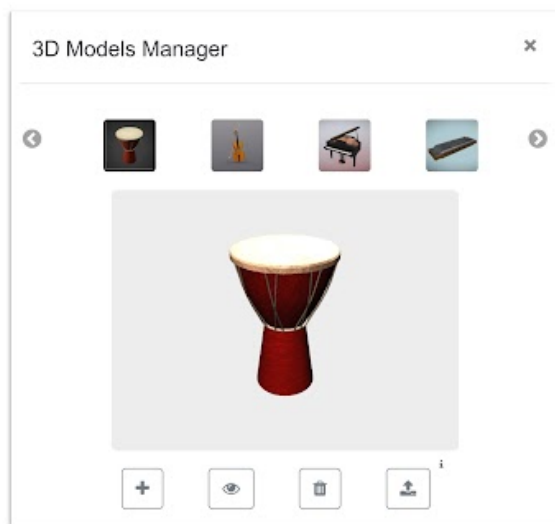
Mark only one oval.

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

Questionnaire (3D)

Only proceed to this section after concluding the workshop.

6. I found that the 3D features are a good complement to the annotation system



Mark only one oval.

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

7. I think that I would like to use the 3D features frequently.

Mark only one oval.

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

8. I found the 3D features unnecessarily complex.

Mark only one oval.

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

9. I thought the 3D features were easy to use.

Mark only one oval.

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

10. I think that I would need the support of a technical person to be able to visualize and use the 3D models.

Mark only one oval.

1 2 3 4 5

Strongly disagree ☐ ☐ ☐ ☐ ☐ Strongly agree

15. I felt very confident when interacting with the 3D features.

Mark only one oval.

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

16. I needed to learn a lot of things before I could get going with this 3D-based elements.

Mark only one oval.

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

17. Suggestions:

Questionnaire (Pose Estimation)

Only proceed to this section after concluding the workshop.

18. I found that the pose estimation features are a good complement to the annotation system



Mark only one oval.

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

19. I think that I would like to use the pose estimation features frequently.

Mark only one oval.

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

28. I needed to learn a lot of things before I could get going with this pose estimation features.

Mark only one oval.

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

29. Suggestions:

This content is neither created nor endorsed by Google.

Google Forms

USABILITY TEST GUIDE *WEAVE ONLINE*



Usability Test Guide (Weave Workshop)

NOVA LINCS, Departamento de Informática, Faculdade de Ciências e
Tecnologias, Universidade NOVA de Lisboa

Brief description

This document aims to assist both the participant and the researchers conducting the WEAVE's workshop following the presentation of the Motion Notes system. Before proceeding to the final questionnaire, complete the tasks described below with the help of a researcher if needed.

The newly developed features aim to enrich the annotation system's previously existing mechanisms by integrating both 3D elements as annotation types as well as pose estimation to provide innovative ways to highlight and analyze multimedia content. Additionally, to match the context of this project and its partners, the materials provided (e.g., videos and 3D models) are deeply related to cultural heritage and the overall theme of our esteemed WEAVE project and Europeana Foundation.

For your participation, make sure to have a stable internet connection and access to a working browser (e.g., Google Chrome).

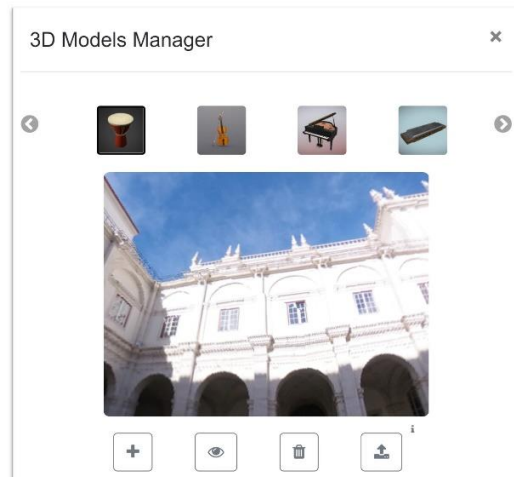
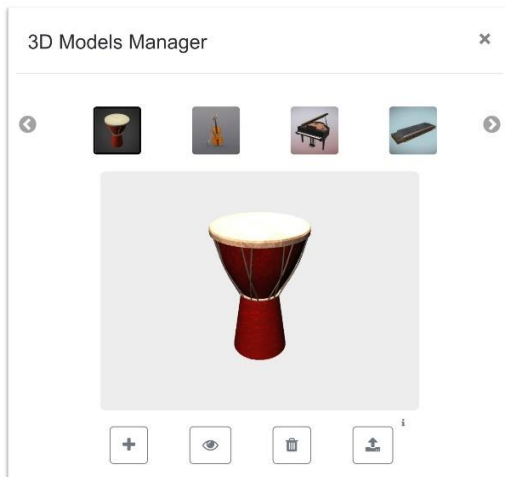
Please read each of the tasks described in the next section attentively while respecting the order assigned to each of them. All feedback provided either during the workshop or later is encouraged and appreciated by the development team, as new ideas and improvements may arise as a direct result of your participation.

To begin your workshop experience, enter: [Motion Notes](#)

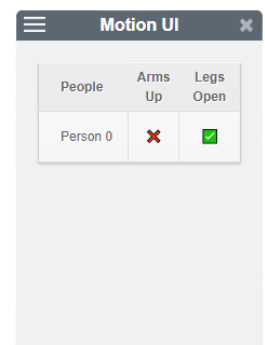
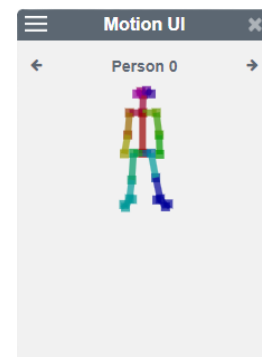
Tasks

1. Create an account
2. Login into *Motion Notes*
3. Import or use demo videos
 - i. File > Import Video
 - ii. File > Available Videos > Human Pose > Open
4. Play the selected video

5. Pause the current video
6. Choose one of the available 3D models
7. Visualize it in a neutral background
8. Choose a different background



9. Add the 3D annotation
10. Interact with it by moving, rotating and/or scaling the object
11. Change the duration of the created annotation
12. Refresh the web page (e.g., F5)
13. File > Available Videos > Human Pose > Open
14. Play the selected video
15. Pause
16. Start pose estimation (search)
17. Open Motion UI (Settings > Pose > toggle)
18. Play the selected video
19. Pause the selected video
20. Open and visualize both tabs on the Motion UI



SPORTS GUIDE *USE CASE: BASKETBALL*



Sports Workshop Guide

Case Study: Basketball

Brief description

This document aims to assist both the participant and the researchers in conducting the workshop session while presenting the Motion Notes system. Throughout the demonstration, each of the presented features are meant to be discussed from the participant's perspective, either as an athlete or coach. Both the demonstration video as well as the pose estimation functionalities are presented in the context of a simulated basketball practice. Initially, the Motion Notes annotation system is succinctly described before focusing on the pose estimation features. All feedback provided either during the workshop or later is encouraged and appreciated by the development team, as new ideas and improvements may arise as a direct result of your participation. Moreover, participants may directly interact with the system at any given time with the help of a researcher if needed. Below is a summary of the topics approached during the workshop:

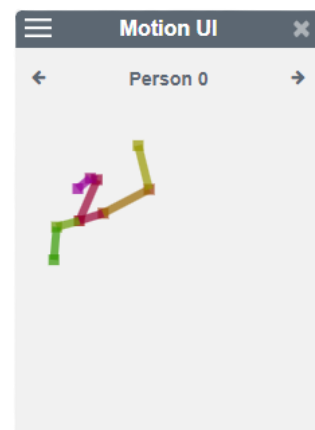
Topics

1. Request personal standard information and written consent
2. Basketball experience summary
 - i. Experience timeline as a coach and/or player
 - ii. Subjective feel over basketball concepts
3. Presenting the *Motion Notes* standard functionalities
 - i. Importing videos (internal or external links)
 - ii. Side annotation triggers
 - iii. Annotation tracks
 - iv. Possible annotation types (e.g., text, drawing)
4. Define Pose Estimation
 - i. Keypoint definition
 - ii. Final skeletal image

5. Discuss base Pose Estimation relevancy
 - i. General perspective as coach or athlete
 - ii. Practical examples
 - iii. Overall feeling regarding the feature



6. Discuss Dynamic Annotation's usefulness
 - i. General perspective as coach or athlete
 - ii. Practical examples
 - iii. Overall feeling regarding the feature
 - iv. Improve athlete's focus (?)

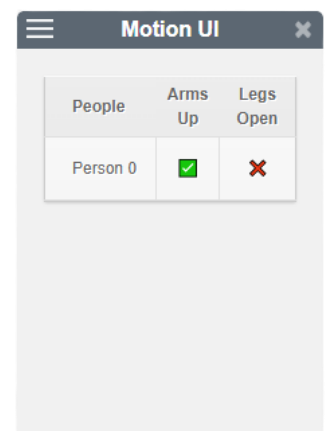


7. Discuss Skeletal View's value
 - i. General perspective as coach or athlete
 - ii. Practical examples
 - iii. Overall feeling regarding the feature
 - iv. Using pose as reference (?)

8. Discuss Automatic Gesture Recognition
 - i. General perspective as coach or athlete
 - ii. Practical examples
 - iii. Overall feeling regarding the feature
 - iv. Value as external feedback (?)

9. Possible application in other sports

10. Suggestions and Improvements



USABILITY TEST GUIDE *FINAL EVALUATION*



Usability Test Guide (Final Workshop)

NOVA LINCS, Departamento de Informática, Faculdade de Ciências e
Tecnologias, Universidade NOVA de Lisboa

Brief description

This document aims to assist both the participant and the researchers conducting the final in-person workshop following the presentation of the Motion Notes system. Before proceeding to the final questionnaire, complete the tasks described below with the help of a researcher if needed.

The newly developed features aim to enrich the annotation system's previously existing mechanisms by integrating both 3D elements as annotation types as well as pose estimation to provide innovative ways to highlight and analyze multimedia content. Additionally, to match the context of this project and its partners, some of the materials provided (e.g., videos and 3D models) are deeply related to cultural heritage and the overall theme of our esteemed WEAVE project and Europeana Foundation.

For your participation, make sure to have a stable internet connection and access to a working browser (e.g., Google Chrome).

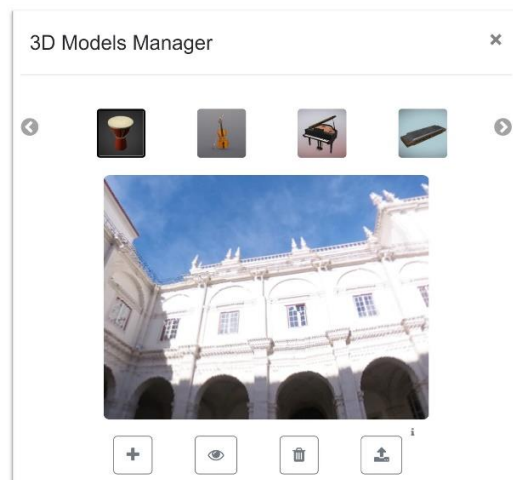
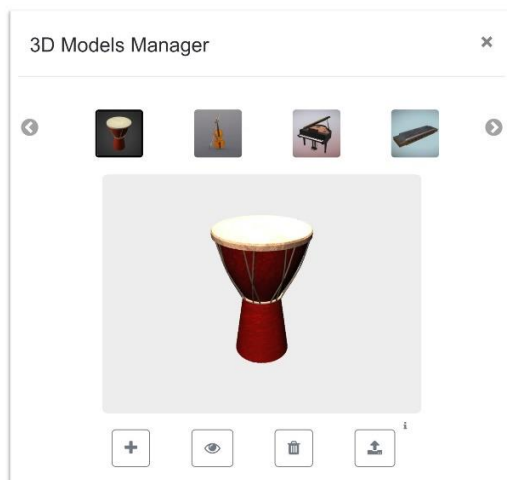
Please read each of the tasks described in the next section attentively while respecting the order assigned to each of them. All feedback provided either during the workshop or later is encouraged and appreciated by the development team, as new ideas and improvements may arise as a direct result of your participation.

To begin your workshop experience, enter: [Motion Notes](#)

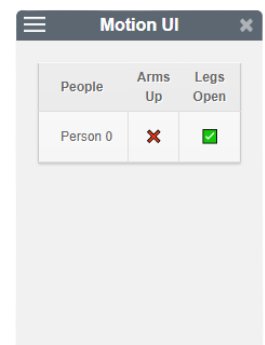
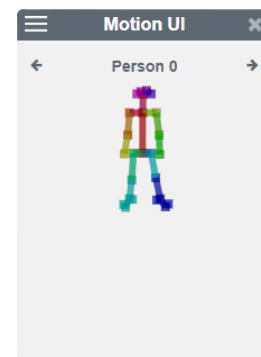
Tasks

1. Create an account
2. Login into *Motion Notes*
3. Import or use demo videos
 - i. File > Import Video
 - ii. File > Available Videos > Human Pose > Open

4. Play and pause the selected video
5. Enter [sketchfab](https://sketchfab.com) to download a 3D model
6. Choose from one of the available 3D models
7. Delete the model and import that same model



8. Visualize it in a neutral background (record double-click vs button)
9. Choose a different background
10. Add the 3D annotation
11. Interact with it by moving, rotating and/or scaling the object
12. Add a different annotation type
13. Change the duration of the created annotation
14. Refresh the web page (e.g., F5)
15. File > Available Videos > Human Pose > Open
16. Play and pause the selected video
17. Start pose estimation (search)
18. Associate annotation to a body part (keypoint)
19. Adjust the duration and play the video
20. Open Motion UI (Settings > Pose > toggle)
21. Play the selected video and pause
22. Open and visualize both tabs on the Motion UI



E

CONSENT FORM FOR USABILITY TEST *FINAL*
EVALUATION



Information and Consent Form for Usability Test

Theme: Integrating 3D objects and pose estimation for multimodal video annotations

Researchers: Prof. Nuno Correia, João Diogo

I'm a student for the Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, and I'm currently finishing my dissertation for my master's in Computer Science and Engineering. Summarily, the main goal of this thesis is to integrate 3D elements and pose estimation for an existing annotation tool: Motion Notes. In this workshop, you will conduct a series of experiments regarding the provided systems in order to subsequently provide valuable feedback.

There will be no monetary losses or gains as a result of your participation, nor will you receive any advantageous benefit for being a participant. However, this study will allow this project's researchers to gain a deeper understanding of the system's usability and possible weaknesses.

Your participation must be voluntary. Refusing to participate in these tasks will not cause you harm or jeopardize any benefits you may already have. The lead investigator might remove you from the study. In that case, you will not be penalized in any way as a direct consequence of doing so.

If you have any questions regarding this workshop, please reach out to any of the following contacts:

Professor: Nuno Correia

Institution: Departamento de Informática,
Faculdade Ciências e Tecnologia, UNL

Email: nmc@fct.unl.pt

Student: João Diogo

Institution: Faculdade Ciências e
Tecnologia, UNL

Email: jp.diogo@campus.fct.unl.pt

I've read this document completely. Therefore, I fully understand the nature of this study, and I agree to be a participant. The lead researcher and respective associates have my permission to use the results of the mentioned experiments for academic use, such as in oral class presentations or others, thereby contributing to the scientific community as long as my identity remains anonymized.

I allow the recording of my voice and image to authorized researchers only.

PARTICIPANT'S SIGNATURE

DATE (DD/MM/YY)

F

USABILITY QUESTIONNAIRE *FINAL*
EVALUATION

Motion Notes Questionnaire (In-Person)

The following set of questions aims to gather user feedback using the Motion Notes annotation system in the context of the [WEAVE](#) European project. Firstly, you will interact with the recent contributions to the system, namely, the implemented 3D and pose estimation features based on the material provided. Following that, you may begin to answer the presented questionnaire.

Feel free to ask questions to any of Motion Notes collaborators.

***Required**

1. I accept the terms presented to me in the usability guide in order to participate in this workshop and answer the following questions. *

Note: Your participation must be voluntary. Refusing to participate in these tasks will not cause you harm or jeopardize you in any manner. By agreeing to participate in this study, you are granting permission to use its results anonymously for academic use, such as in oral class presentations or others, thereby contributing to the scientific community.

Mark only one oval.

☐ Yes

2. Age *

3. Gender *

Mark only one oval.

☐ Male

☐ Female

☐ Prefer not to say

4. How often do you use a web browser (e.g., Google Chrome, Firefox, Edge)? *

Mark only one oval.

1 2 3 4 5

Almost Never ☐ ☐ ☐ ☐ ☐ Almost Always

5. How comfortable are you with note making over a video? *

Mark only one oval.

1 2 3 4 5

Really uncomfortable ☐ ☐ ☐ ☐ ☐ Really comfortable

Questionnaire (SUS)

Only proceed to this section after concluding the workshop.
Even though the system is composed of multiple annotation features, the following answers you give regarding the Motion Notes system should be mostly based on your experience using 3D and pose estimation functionalities.

6. I think that I would like to use this system frequently. *

Mark only one oval.

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

7. I found the system unnecessarily complex. *

Mark only one oval.

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

12. I would imagine that most people would learn to use this system very quickly. *

Mark only one oval.

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

13. I found the system very cumbersome to use. *

Mark only one oval.

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

14. I felt very confident using the system. *

Mark only one oval.

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

15. I needed to learn a lot of things before I could get going with this system. *

Mark only one oval.

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

Questionnaire (3D Features)

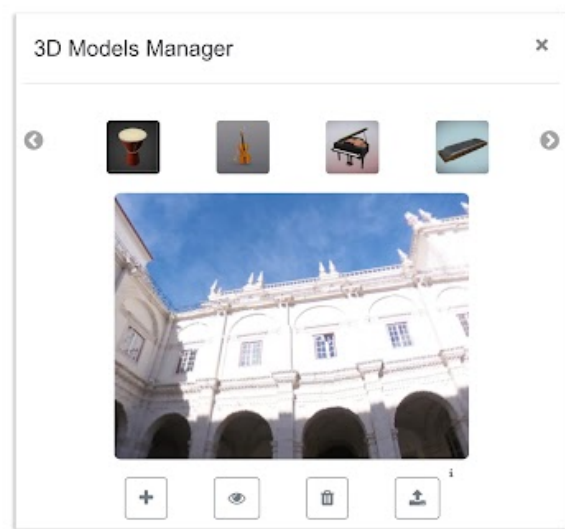
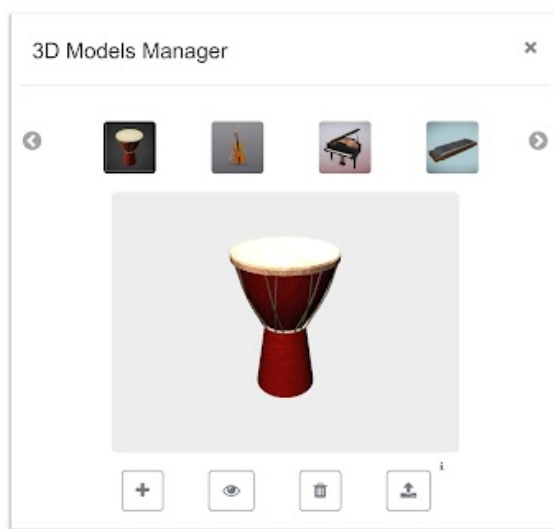
Only proceed to this section after concluding the workshop.

16. I found that the 3D features are a good complement to the annotation system

Mark only one oval.

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

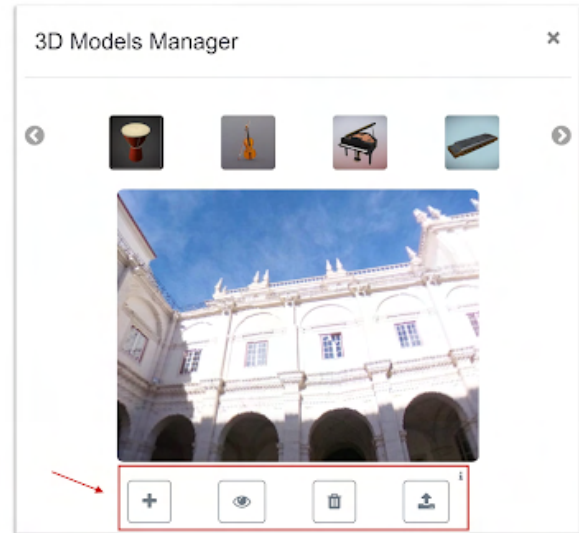
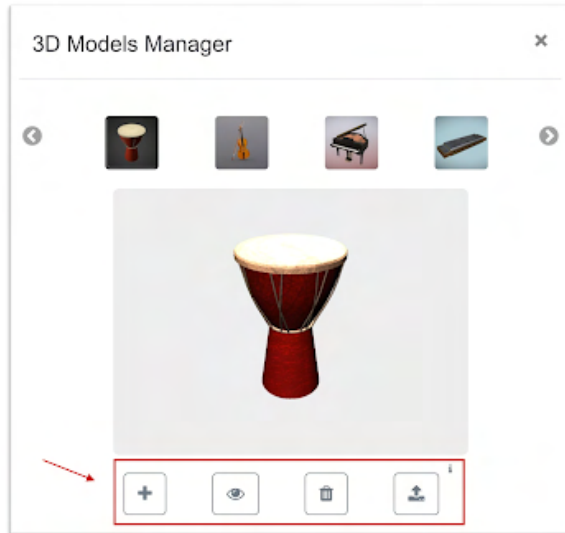
17. I found the "3D Models Manager" window to be an intuitive interface to import and visualize 3D objects.



Mark only one oval.

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

18. I could easily understand what actions each button triggered.

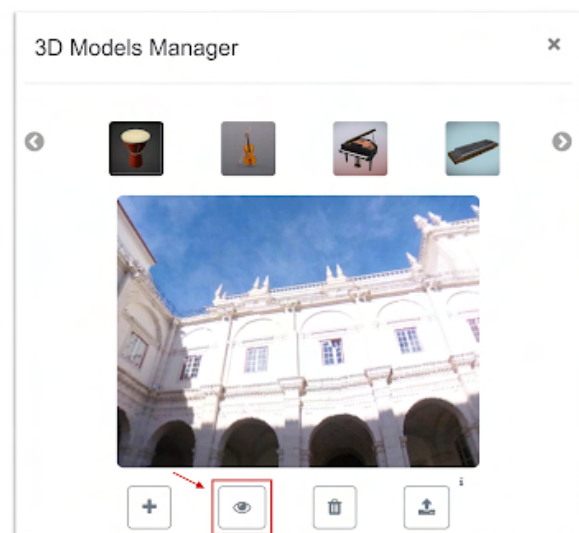
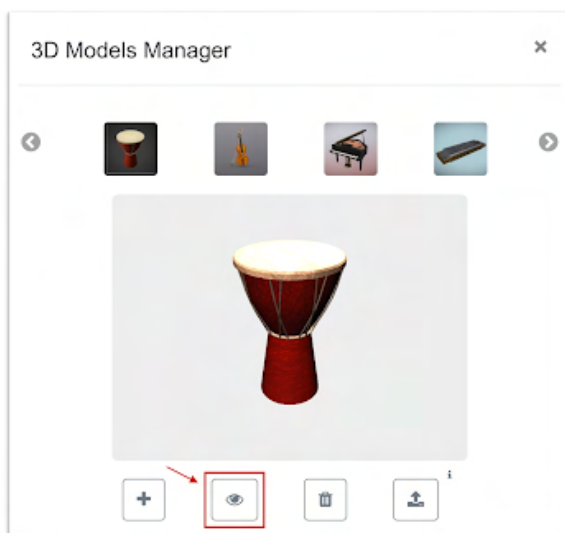


Mark only one oval.

1 2 3 4 5

Strongly disagree ☐ ☐ ☐ ☐ ☐ Strongly agree

19. I felt it was simple to display the selected model and interact with it (e.g., rotating and moving it).



Mark only one oval.

1 2 3 4 5

Strongly disagree ☐ ☐ ☐ ☐ ☐ Strongly agree

20. Having the possibility of choosing (360°) backgrounds to visualize the objects was appealing to me.

Mark only one oval.

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

21. I thought that 3D annotations were inconsistent with other annotation types (e.g., drawing and text).

Mark only one oval.

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

22. I enjoyed being able to interact with the 3D elements after placing them on the video.

Mark only one oval.

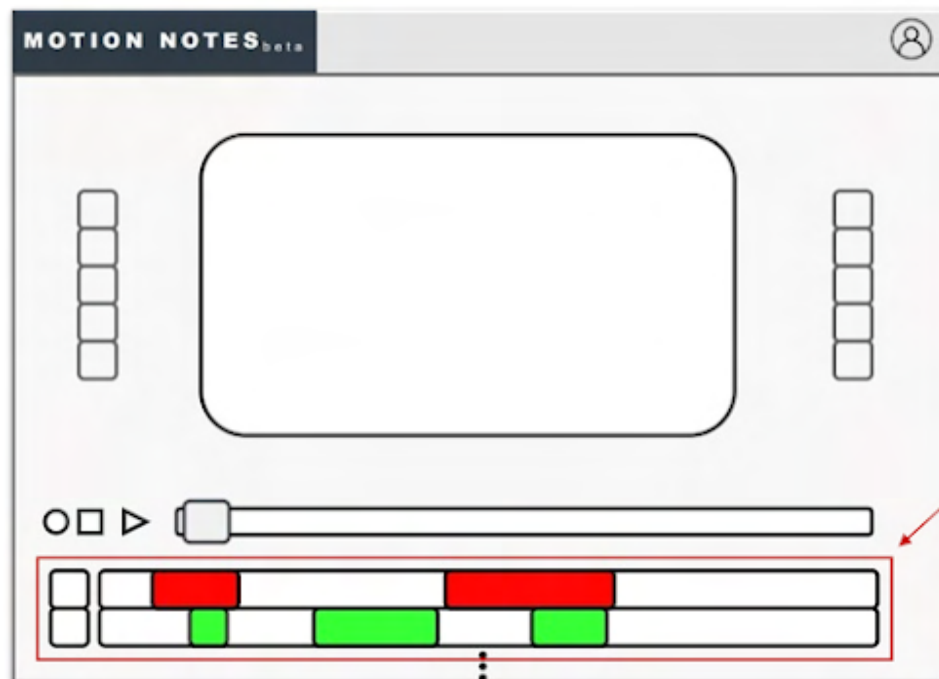
	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

23. I felt that adding 3D-based annotations was justified given the existing types of annotations.

Mark only one oval.

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

24. I would imagine most people would easily understand how the annotation track works.



Mark only one oval.

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

25. Suggestions (3D features):

Questionnaire (Pose Estimation)

Only proceed to this section after concluding the workshop.

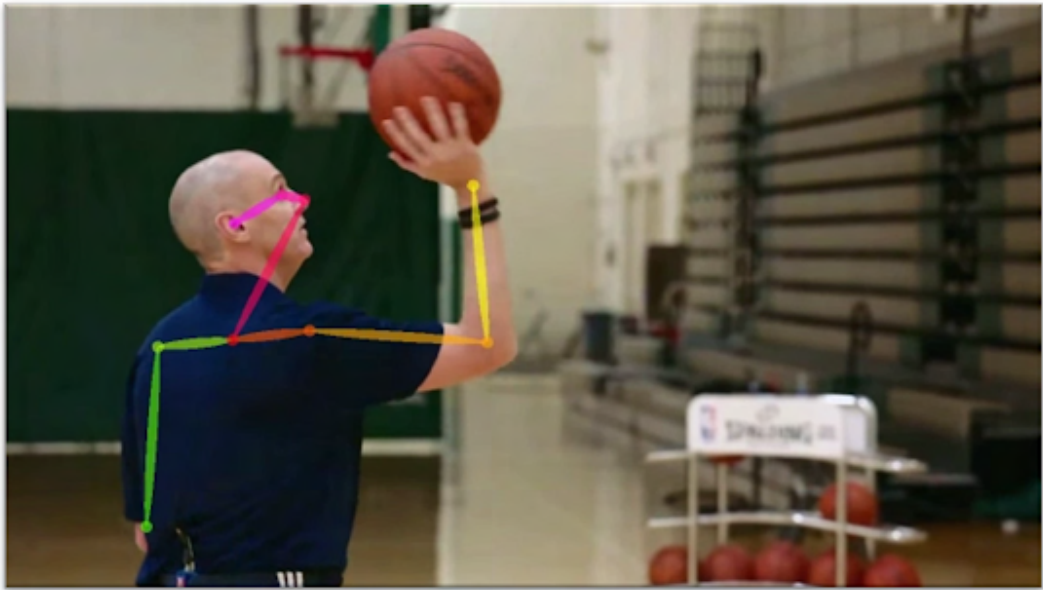
26. I found that the pose estimation features are a good complement to the annotation system *



Mark only one oval.

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

27. I felt that it was intuitive to visualize the estimated pose through the skeletal figure. *



Mark only one oval.

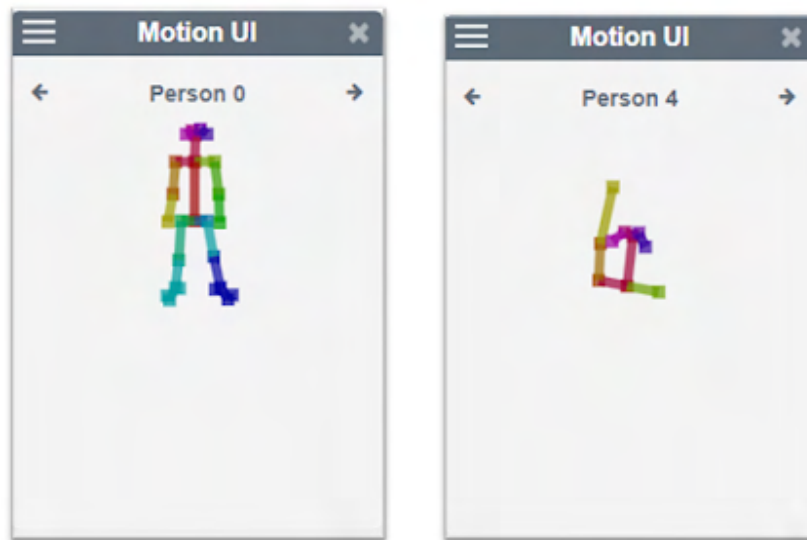
	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

28. I think that having dynamic annotations is a useful complement to the base pose estimation *
feature.

Mark only one oval.

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

29. I found the skeletal view available on the Motion UI to be relevant in the system's context.

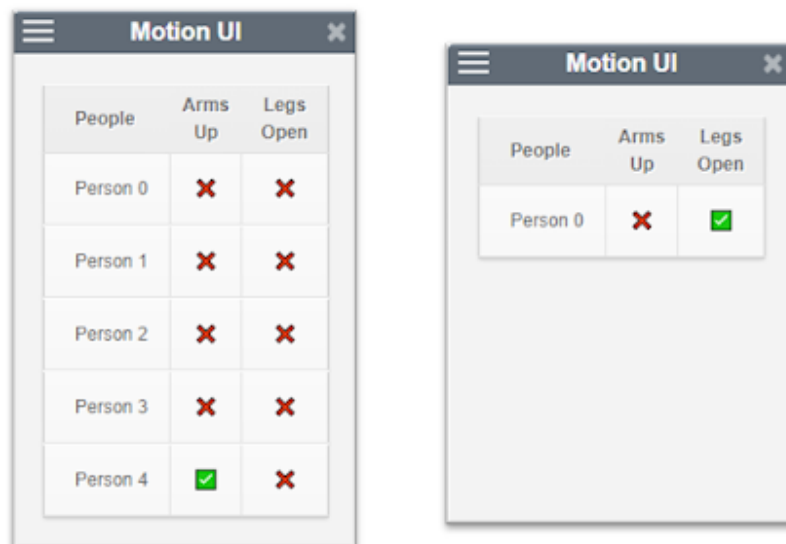


Mark only one oval.

1 2 3 4 5

Strongly disagree ☐ ☐ ☐ ☐ ☐ Strongly agree

30. I thought that the automatic gesture identification feature has a lot of potential.



Mark only one oval.

1 2 3 4 5

Strongly disagree ☐ ☐ ☐ ☐ ☐ Strongly agree

31. I felt that the pose estimation features did not hinder the existing annotation mechanisms.

Mark only one oval.

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

32. Suggestions:

This content is neither created nor endorsed by Google.

Google Forms

