UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Felipe Augusto Pedroso

ImmVis Framework: Establishing a Bridge between Immersive Applications and Data Analysis

ImmVis Framework: Construindo uma Ponte entre Aplicações Imersivas e Análise de Dados

Campinas

2021

Felipe Augusto Pedroso

ImmVis Framework: Establishing a Bridge between Immersive Applications and Data Analysis

ImmVis Framework: Construindo uma Ponte entre Aplicações Imersivas e Análise de Dados

Dissertation presented to the School of Electrical Engineering and Computer Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Electrical Engineering in the area of Computer Engineering.

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica, na Área de Engenharia de Computação.

Supervisor: Profa. Dra. Paula Dornhofer Paro Costa

Este exemplar corresponde à versão final da dissertação defendida pelo aluno Felipe Augusto Pedroso, e orientada pela Profa. Dra. Paula Dornhofer Paro Costa.

Campinas

2021

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

P343i Pedroso, Felipe Augusto, 1987-
ImmVis Framework : establishing a bridge between immersive applications and data analysis / Felipe Augusto Pedroso. – Campinas, SP : [s.n.], 2021.

Orientador: Paula Dornhofer Paro Costa.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Visualização da informação. 2. Análise de dados. 3. Realidade virtual. 4. Realidade aumentada. 5. Software gratuito. I. Costa, Paula Dornhofer Paro, 1978-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: ImmVis Framework : construindo uma ponte entre aplicações imersivas e análise de dados
Palavras-chave em inglês:
Information visualisation
Data analysis
Virtual reality
Augmented reality
Free software
Área de concentração: Engenharia de Computação
Titulação: Mestre em Engenharia Elétrica
Banca examinadora:
Paula Dornhofer Paro Costa [Orientador]
Letícia Rittner
André Santaché
Data de defesa: 02-06-2021
Programa de Pós-Graduação: Engenharia Elétrica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-2463-0459
- Currículo Lattes do autor: http://lattes.cnpq.br/0478777841950371

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Felipe Augusto Pedroso RA: 084995
Data de defensa: 02 de Junho de 2021
Titulo da Tese: "ImmVis Framework: Establishing a Bridge between Immersive Applications and Data Analysis"
"ImmVis Framework: Construindo uma Ponte entre Aplicações Imersivas e Análise de Dados"

Profa. Dra. Paula Dornhofer Paro Costa (Presidente)
Prof. Dr. André Santanché
Profa. Dra. Leticia Rittner

A Ata de Defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

Acknowledgements

I want to thank Prof. Dr Paula Dornhofer Paro Costa for all the support, patience and guidance during the whole process of my master's degree. I am grateful for the opportunity of having her as a supervisor and tutor.

I want to thank my friends and colleagues for the support and friendship during this process. Among them, I would like to give a special acknowledgement to Eduardo Carrara, Rodolfo Tonoli, and Caluã Pataca which helped with their reviews, knowledge, or simply being there to listen.

I want to thank my family for providing me support and love during my whole life. Without my parents efforts and sacrifices, I would not be able to achieve anything.

I want to thank my wife, Bruna Panaggio, for her immense care and support along the way. From reviewing texts to supporting me during the hard times, I hope one day to pay back everything that she did to make my life easier, warmer and happier during the last years.

Abstract

The Immersive Analytics (IA) field faces the same challenge as other fields related to data visualisation: choosing the appropriate tool to use on projects or experiments. To alleviate that, IA practitioners started using game engines and web technologies to implement their solutions, as they solve problems like rendering and user interaction. However, it is not common to find projects that, together with visualisation authoring and interaction, offer data analysis capabilities inside immersive environments. The present work proposes the creation of ImmVis, an open-source framework that empowers immersive applications with Python data analysis capabilities. The framework leverages the features from *gRPC*, a remote procedure call framework, to implement the communication between Python and the programming languages used by IA applications. This approach could potentially complement existing projects from the IA field or enable new types of interaction with data.

Keywords: Information visualisation, Data analysis, Virtual reality, Augmented reality, Free software.

Resumo

A área de pesquisa "Immersive Analytics" (IA) enfrenta um desafio comum de outras áreas relacionadas à visualização de dados: a escolha de uma ferramenta adequada para a condução de projetos ou experimentos. Para aliviar o problema, os praticantes de IA se utilizam de motores de jogos (do inglês, *game engines*) e tecnologias web para implementar suas soluções, já que as mesmas resolvem problemas como renderização e interação com o usuário. No entanto, não é comum encontrar projetos que, juntamente com ferramentas de autoria de visualizações e funcionalidades de interação, ofereçam recursos de análise de dados dentro de ambientes imersivos. O presente trabalho propõe a criação do framework de código aberto chamado ImmVis, que permite que aplicativos imersivos usem recursos de análise de dados presentes na linguagem de programação Python. Este framework emprega os recursos do *gRPC*, um framework de chamada de procedimento remoto, para implementar a comunicação entre a linguagem de programação Python e as linguagens de programação utilizadas na criação de aplicações imersivas. Essa abordagem pode complementar projetos existentes na área de IA ou permitir novos tipos de interação com dados.

Palavras-chaves: Visualização da informação, Análise de dados, Realidade virtual, Realidade aumentada, Software gratuito.

List of Figures

List of abbreviations and acronyms

API	Application Programming Interface
AR	Augmented Reality
CAVE	Cave Automatic Virtual Environment
gRPC	*gRPC* Remote Procedure Call (recursive acronym)
GUI	Graphical User Interface
IDL	Interface Definition Language
IA	Immersive Analytics
RPC	Remote Procedure Call
UDP	User Datagram Protocol
VR	Virtual Reality
VA	Visual Analytics

Contents

1 Introduction

The current production of data is continuously reaching unprecedented levels and, together with that, data complexity is also surging drastically (MARR, 2015; DONALEK et al., 2014). While the "data deluge" phenomenon brings new opportunities for science in general, it also challenges our ability to understand or utilise data in decision making (DWYER et al., 2018; DONALEK et al., 2014).

The greater access and computational capabilities of processing vast amounts of data gave the data science field a prominent role in scientific disciplines. They even created the perception that ample data availability could cause "the end of the theory" (KARPATNE et al., 2017). This perception can be attributed mainly to the emergence of black-box applications, typically present on *deep learning* solutions, that build actionable models without relying on the scientific theory behind them. While using these techniques is obtaining reasonable levels of success to solve practical problems, the humans' scientific knowledge from prior experience is still an important asset that can improve the knowledge discovery process (FEDERICO et al., 2017; KARPATNE et al., 2017).

A typical approach to include specialists in the data analysis process uses graphical thinking in exploratory studies, as it provides a straightforward and natural type of data processing for human beings. However, graphical thinking with massive amounts of data imposes challenges like visual noise, large image perception, information loss, high-performance requirements, and high rate of image change (GORODOV; GUBAREV, 2013).

Visual Analytics (VA) field proposes tackling these challenges through a combination of automated analysis techniques with interactive visualisations, supporting humans during the data analysis process and enabling a better understanding, reasoning, and decision making (KEIM; KOHLHAMMER; ELLIS, 2010). Immersive Analytics (IA) is a "derivative" field from VA that proposes to complement the VA approach with the usage of innovative interface technologies like Virtual Reality (VR) headsets, large touch surfaces and Cave Automatic Virtual Environment (CAVE).

Since IA is a relatively new field, researchers do not have many tools for creating and evaluating these interfaces in the data analysis context. To overcome that, many use game and web technologies to develop their solutions or tools and then perform the evaluations (CORDEIL et al., 2017; CORDEIL et al., 2019; SICAT et al., 2018; BUTCHER; JOHN; RITSOS, 2020). Most of these solutions present impressive results but give a more significant focus on implementing the visualisation or simplifying immersive visualisations creation process, not enabling automated data analysis operations. Another challenge is that each tool approaches data handling and representation differently, making interoperability more challenging.

The present work proposes ImmVis, a framework that provides data analysis services through the network, enabling IA systems created using different programming languages to benefit from the approach. Furthermore, our proposal aims to help researchers explore new ways of interacting with data inside IA applications, complementing existing projects or enabling new ones.

1.1 Objectives

The main objective of the present work was to propose a communication framework that enables data analysis functionalities inside IA applications, allowing visualisation and data analysis practitioners to explore how these functionalities could create new types of interaction with data inside immersive environments.

Specific objectives in this project were:

1. To investigate the state of the art of tools available for creating IA applications.
2. To propose a framework that exposes data analysis services through the network, enabling IA applications to integrate this type of functionality.
3. To create an open-source prototype of this framework using Python to implement the data analysis services and the Unity game engine to render the data and interact with users.
4. To implement a sample application that uses the framework functionalities to serve as documentation and enable other researchers to experiment with the framework.

1.2 Contributions

The main contributions of this work are:

- *Framework proposal*: in the present work, we document our journey to create the ImmVis framework, describing what was tried and establishing conceptual requirements for implementing it.
- *Implementation of a prototype*: we created an open-source prototype of the framework to help immersive application developers to integrate functionalities available on Python data analysis libraries. We built the network communication using the *gRPC* framework, enabling the data services to be consumed by different platforms and programming languages.

- *Case studies*: we implemented a sample application using the Unity game engine that is capable of plotting an interactive 3D scatterplot and performing the k-means analysis technique. We also explored a proof of concept of the framework inside Jupyter Notebooks, a tool that data analysis practitioners widely adopt.

The contributions of this work were also reported in the following publications:

- Felipe Augusto Pedroso, and Paula Dornhofer Paro Costa. 2021. ImmVis: Bridging Data Analytics and Immersive Visualisation. In Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 3: IVAPP, ISBN 978-989-758-488-6 ISSN 2184-4321, pages 181-187. DOI: 10.5220/0010256001810187

- Felipe Augusto Pedroso, and Paula Dornhofer Paro Costa. 2018. Improved Knowledge from Data: Building an Immersive Data Analysis Platform. In Virtual and Augmented Reality Graduate Workshop of 20th Symposium on Virtual and Augmented Reality (SVR 2018)

1.3 Text Organisation

This text is organised as follows. First, the core concepts and the related work are presented in Chapter 2. Then, in Chapter 3, we describe the ImmVis framework alongside its technical aspects and development context. Next, we present two case studies in Chapter 4: a sample application built with the framework and an experiment made to run ImmVis inside Jupyter Notebooks. Finally, Chapter 5 presents our final remarks and some discussions about future work.

2 Concepts and Related Work

This chapter presents an overview of Visual Analytics and Immersive Analytics, including the open problems that motivate the present work. Section 2.1 provides an introduction to the Visual Analytics (VA) field. Section 2.2 describes the Immersive Analytics (IA) field and review the current offer of tools available to create IA applications. The final section of this chapter discusses how the current work contributes to the IA field and its development tools.

2.1 Visual Analytics

Visual Analytics (VA) is a research field first described by Thomas and Cook (2006) that proposes using interactive visual interfaces to facilitate analytical reasoning. Keim, Kohlhammer and Ellis (2010) establishes that VA combines automated analysis techniques with interactive visualisations to enable a better understanding, reasoning, and decision making on enormous and complex data sets.

According to Keim, Kohlhammer and Ellis (2010), the VA process can be illustrated by the Figure 2.1, which shows the multiple interactive cycles used by visual data exploration to enable knowledge discovery. During these steps, analysts cycle between visual and automatic methods to find models and visualisations to obtain relevant knowledge. It is important to note that both types of methods "feed" each other in a loop. For example, users can use visualisation to refine the generated models and the models to suggest different types of visualization or improve the current ones.

The VA field is multidisciplinary and comprises the following focus areas: analytical reasoning techniques; visual representation and interaction techniques; data representation and transformations; methods to support production, presentation and dissemination of analytical results (THOMAS; COOK, 2006). The VA techniques and practices are being applied with success by different areas: emergency management; astronomy; monitoring climate and weather; environmental monitoring; security; scientific applications; biology and medicine; transport and logistics; business intelligence; personal information management; and fraud detection (KEIM; KOHLHAMMER; ELLIS, 2010; CHANDLER et al., 2015; PREIM; LAWONN, 2020). For example, Figure 2.2 shows the output of *DatAC*, a visual analytics platform created by Martorell-Marugán et al. (2021) to evaluate climate and air quality indicators and its association with COVID-19 cases in Spain.

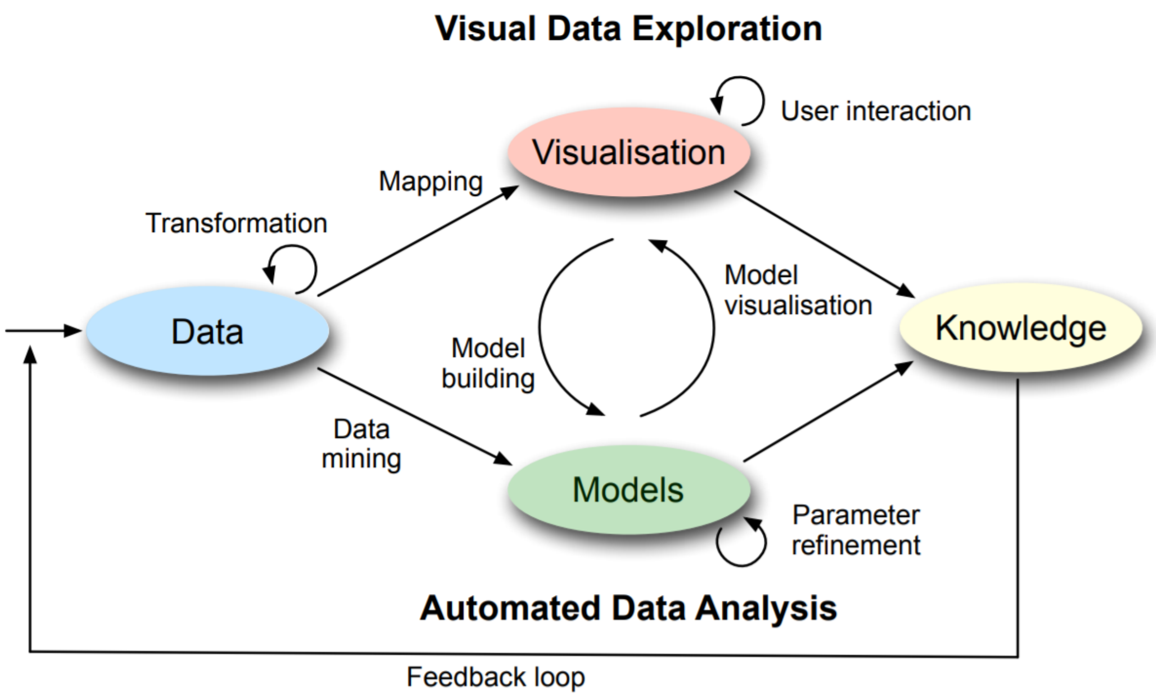


Figure 2.1 – The VA process proposed by Keim, Kohlhammer and Ellis (2010).

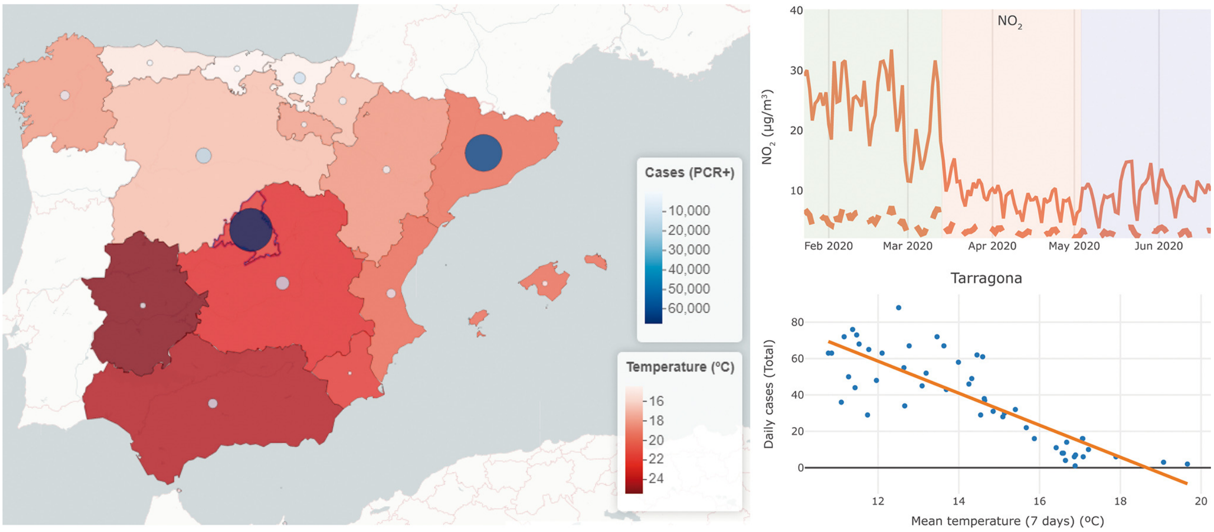


Figure 2.2 – A map created with the visual analytics platform *DatAC*, which shows the COVID-19 cases confirmed by PCR test and mean temperature in Spain on May 24th 2020 (MARTORELL-MARUGÁN et al., 2021).

2.2 Immersive Analytics

The definition of VA is agnostic regarding the interface devices used by its systems. Still, most VA applications base their interactions on devices available on a conventional desktop computer: a display, keyboard, and mouse, since these devices presented good results historically (CHANDLER et al., 2015; CHEN; LIN; YUAN, 2017; LIN et al., 2018; KWON et al., 2019; WANG et al., 2019; ZHAO et al., 2019; CUI, 2019). However, their relevance as computer interaction "language" is declining as their experience presents some limitations, and new technologies are becoming more affordable for end users (CHANDLER et al., 2015; DWYER et al., 2018; CUI, 2019).

The Immersive Analytics (IA) field comes to this space proposing the usage of innovative interfaces and devices to support analytical reasoning and decision making (CHANDLER et al., 2015). The origin of this field has a direct relationship with the unprecedented emergence of affordable display and interaction devices like the Oculus Quest[1], HTC Vive[2], and Microsoft HoloLens[3]. Although the term "immersive" suggests the use of virtual reality devices, the field also explores the usage of other types of interface technologies such as Augmented Reality (AR), large wall-mounted displays, touch devices, or wearable displays (CHANDLER et al., 2015; DWYER et al., 2018).

IA is a multidisciplinary field and shares many goals with VA, especially when it comes to deriving insights from data sets that are big and complex (CHANDLER et al., 2015). The main difference between them is that IA, together with the use of new technologies, also considers multi-sensory interaction to enable new ways of experiencing data and make the systems accessible for people with visual impairment (MCCORMACK et al., 2018). As stated by Dwyer et al. (2018), the main goal of IA is to remove the barriers between people, data, and analysis tools.

The usage of 3D for abstract data representation faces a cautious view from the data visualisation community, as it can present problems like occlusion, depth disparity, and foreshortening (MUNZNER, 2014). However, technologies explored by IA shine a new light on this view, as their advances help solve some of these problems. Additionally, user experience, collaboration and engagement are used to reframe the concept of effectiveness while evaluating IA applications (MARRIOTT et al., 2018).

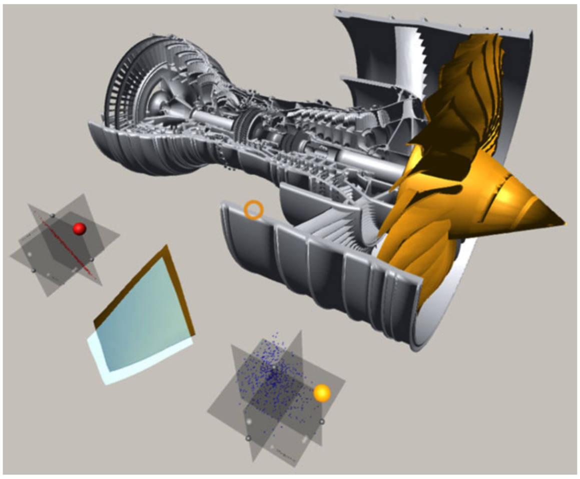
Despite being a relatively new research field and considering the knowledge about designing appropriate solutions is under development, some promising IA applications are emerging in multiple science fields. We list some of these applications below:

- In "AeroVR" (Figure 2.3a), Tadeja, Seshadri and Kristensson (2020) created an

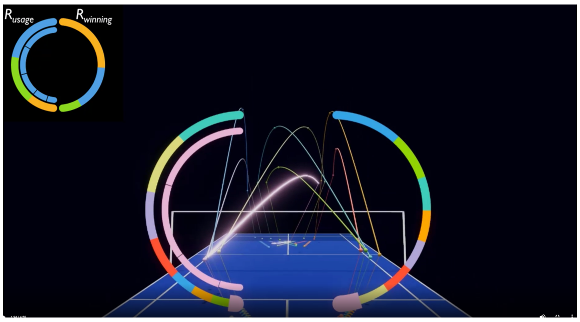
1 <https://www.oculus.com/quest/> (Accessed on July 22, 2021)
2 <https://www.vive.com/> (Accessed on July 22, 2021)
3 <https://www.microsoft.com/en-gb/hololens/> (Accessed on July 22, 2021)

immersive visualisation system for aerospace design and digital twinning in virtual reality to study a compressor blade of an engine.

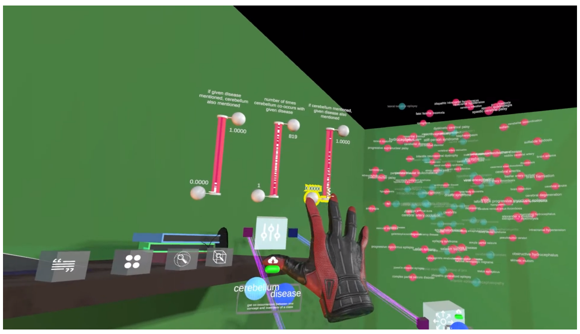
- Lin et al. (2020) defined the concept of SportsXR as the usage of IA focused on sports and described some user scenarios in training, coaching and fan experiences. In "ShuttleSpace" (Figure 2.3b), Ye et al. (2020) proposes a practical implementation of the concept with an application that support experts to analyse badminton trajectory data.
- "DatAR" (Figure 2.3c), a prototype created by Troost et al. (2020), uses an immersive environment to enable the analysis of large collections of neuroscience research papers.
- "DarkSky Halos" (Figure 2.3d) is an IA collaborative system that uses a CAVE environment to explore large scale cosmological simulation data where, for example, researchers could study aspects from dark matter formation (HANULA et al., 2019).



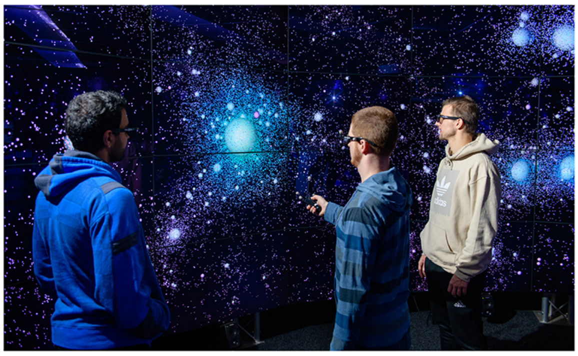
(a) AeroVR (TADEJA; SESHADRI; KRISTENSSON, 2020)



(b) ShuttleSpace (YE et al., 2020).



(c) DatAR (DATAR, 2020).



(d) DarkSky Halos (HANULA et al., 2019)

Figure 2.3 – Screenshots of AeroVR (2.3a), ShuttleSpace (2.3b), DatAR (2.3c) and DarkSky Halos (2.3d)

Other examples of fields exploring IA applications are biology, neuroscience, epidemiology, ecology, crowd simulation, product engineering, archaeology, seismology, and cultural heritage (CZAUDERNA et al., 2018; CHANDLER; MORGAN; KUHLEN, 2018; FULMER et al., 2019; LAWTON et al., 2020; KOEBEL; AGOTAI; ÇÖLTEKIN, 2020).

2.2.1 Available Tools for Creating Immersive Analytics Applications

One of the pioneer tools from the IA field, iViz, proposes using the Unity game engine to create a multi-platform visualisation tool. The main features of iViz are the ability to render significant amounts of data and a collaborative environment, which allows multiple users to visualise and interact with data at the same time (DONALEK et al., 2014). According to the authors, iViz was the precursor of *Virtualitics Immersive Platform*®(VIP), a commercial data analysis solution. VIP supports data exploration on desktop and VR environments and offers machine learning services to analyse data, recommend visualisations and improve the insights extraction process (VIRTUALITICS, 2015). Unfortunately, while there is a documentation portal[4], there is no easy way to use or test the tool since its source code is not available, there are no download links, and the authors did not answer our requests to try the tool.

Filonik et al. (2016) proposed Glance, a conceptual framework that, through the implementation of GPU-accelerated visualisations, enables the creation of data exploration applications on VR and Augmented Reality (AR) environments. The project is available as open-source on the author's Github page[5], but it does not present any documentation or signs that the project moved to a more concrete stage.

LookVR is an experimental tool that leverages the capabilities of Looker[6] commercial data analysis platform on an IA application (Figure 2.4). The tool allows the users to create visualisations using data from existing Looker accounts or embedded examples. LookVR explores different metaphors to visualise data. Among the metaphors, we highlight the "escalating big data", where users use VR controllers to climb a giant representation of the data, improving their perception of the magnitude of the data (GIESELER, 2017). Its source code is not available, but it is possible to download an executable for free on Steam[7].

The ImAxes system is an immersive open-source system developed using Unity that allows exploring multivariate data through the manipulation of objects in a virtual environment (CORDEIL et al., 2017). The system offers a Unity package that helps users create immersive visualisations using a Graphical User Interface (GUI) and an Application Programming Interface (API).

Proposed by Cordeil et al. (2019) and Sicat et al. (2018), "Immersive Analytics

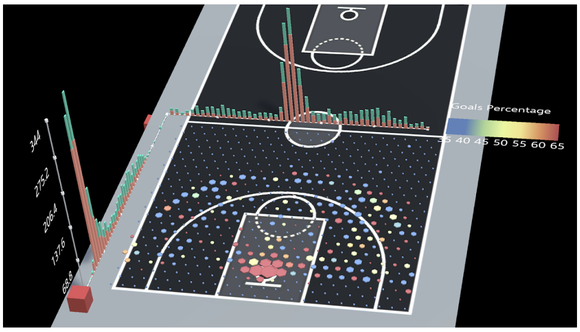
[4] <https://docs.virtualitics.com/> (Accessed on July 22, 2021)
[5] <https://github.com/filonik/glance> (Accessed on July 22, 2021)
[6] <https://looker.com/> (Accessed on July 22, 2021)
[7] <https://store.steampowered.com/app/595490/LookVR/> (Accessed on July 22, 2021)

Figure 2.4 – An screenshot from the LookVR demonstration video that shows an user analysing a scatterplot on an immersive environment (LOOKER, 2017).

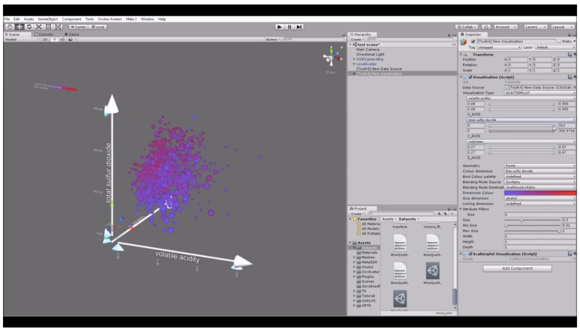
Toolkit" (IATK), and "Data visualisation applications for eXtended Reality" (DXR) are open-source toolkits enable users to author immersive visualisations inside the Unity game engine (Figure 2.5). Through them, users can create immersive visualisations using a Graphical User Interface (GUI) or an Application Programming Interface (API). The goal of offering both pathways is to help users with different levels of programming knowledge to create their visualisation applications. The IATK framework can be considered the successor of ImAxes and is optimised to render massive amounts of data (CORDEIL et al., 2019; REINA et al., 2020). DXR does not have the same optimisation but offers a visualisation grammar inspired on Vega-Lite[8] (SATYANARAYAN et al., 2017; SICAT et al., 2018). Both allow custom solutions through APIs, but their main focus is to enable users to create immersive visualisations without data analytics functionalities out of the box.

The Visualisation Toolkit (VTK) is a well-established tool among visualisation practitioners that also allows the development of immersive applications (Figure 2.6). The toolkit leverages the OpenVR standard to support devices like HTC Vive, Oculus Rift, and Windows Mixed Reality headsets (KITWARE, 2016). O'Leary et al. (2017) states that developers and researchers are exploring VTK usage in the immersive context for about two decades with varying results and also proposes new approaches to enable more consistent results. In addition, VTK provides support to some data analysis algorithms and also offers an integration with the Python programming language (QUAMMEN, 2015;

8 <https://vega.github.io/vega-lite/> (Accessed on July 22, 2021)



(a) An screenshot from an application that user DXR to embed sports data into a virtual basketball court (SICAT et al., 2018).



(b) An screenshot from an IATK demonstration video where the author demonstrate how to create a 3D scatterplot using Unity development environment (CORDEIL, 2019)

Figure 2.5 – Screenshots of DXR (2.5a) and IATK (2.5b)

SHARKEY; BENNETT; PEBAY, 2013). However, the documentation regarding both features is unclear about using them inside immersive applications.

Figure 2.6 – A VTK visualisation of aneurysm data running inside a CAVE environment (O'LEARY et al., 2017).

VRIA is an open-source web-based framework proposed by Butcher, John and Ritsos (2020) that enables the creation of IA applications through a declarative grammar described using JSON. The framework leverages open standards to run its applications on a browser, allowing it to reach multiple platforms. The framework also offers a prototyping tool and an API that enables developers to extend or integrate the framework capabilities. The project also contains necessary documentation about using the framework and a boilerplate application, allowing developers to explore its functionalities faster.

There also exist other tools with a minor relevance for the context of this work like AVAR, NiwViw, MRAT, and VR-Viz (MERINO et al., 2020; YIM et al., 2018; NEBELING et al., 2020; SAIFEE et al., 2018). AVAR and MRAT use Unity as the development environment, VR-Viz benefits from web technologies, and NiwViw does not present any technical aspect of the tool. Except for NiwViw, all of them are open source and provide some online documentation about its usage.

2.3 Discussion and Concluding Remarks

Our review shows that IA is an emerging field that offers new possibilities to data analysis practitioners. The tools to create IA applications are still in the dawn, but it is possible to find exciting approaches to create immersive visualisations, to provide a better user experience, and to perform essential data exploration. However, only a few of them offers tools or specific APIs to perform sophisticated data analysis.

For example, VTK is a tool traditionally adopted by the visualisation community and supports immersive technologies and data analysis techniques at some level. Still, the documentation regarding these features is difficult to comprehend, and it is not clear if they work together.

On the other hand, some promising tools, such as Virtualitics, are moving to a proprietary model, significantly impacting the evolution of the IA field by not providing access to individual researchers to test different visualisation paradigms with the tool.

In this context, our work aims to propose a framework to implement data analytics inside immersive applications, complementing the existing tools described in Section 2.2 or enabling researchers to consider this type of functionality when creating new applications. Additionally, we also aim to facilitate the adoption of our framework by considering some concerns regarding visualisation tools pointed by Reina et al. (2020):

- **Licensing model**: In order to succeed as viable visualisation software, the tools ideally would distribute its source code as open-source, empowering the visualisation community to contribute and extend its capabilities to meet their needs. Another alternative would be an extensible tool with an affordable licensing model. Everything we produced during the execution of this work is available on our Github page. Everything is published using the MIT License, enabling anyone to use, modify and redistribute according to their needs.
- **Available documentation**: The lack of documentation is one of the critical factors that can reduce the longevity and success of visualisation software. In the context

of the present work, we considered the following artefacts as essential pieces of documentation: user manuals, tutorials; articles; code samples; feature lists; blog posts, videos, and quick-start apps. We tried to cover the main parts of the framework with as much documentation as possible. Our code contains setup instructions, tutorials about adding new features, and sample code. Section 4.1 provides more details about two of the sample applications that we published the code.

- **Extensibility/Future-proofing**: Given the continuous evolution of the visualisation, providing well-documented ways of creating plugins or integrating new functionalities is necessary to enable the community to explore new use cases or paradigms. Our work proposes a modular architecture that allows other developers to adapt the framework to their needs. We describe the proposed architecture in Chapter 3.

- **Reproducibility**: Replicating the research results is a weak point of the visualisation community. Offering an easy way to evaluate the solution and reproduce the reported results is essential when searching for a tool to adopt or contribute. ImmVis tries to address this by providing open-source projects with documentation, tutorials and samples. The framework is available to any developer to download and test with their datasets, generated datasets or datasets available online.

- **Interoperability**: Integrating with existing solutions or enabling users to apply previous knowledge are characteristics that turn a tool more appealing to the visualisation community. Possible solutions to offer these characteristics are using a standard data format, providing an integration layer, or using well-established standards and programming languages. As shown in Chapter 3, we created a prototype of this framework that integrates Python and the Unity game engine.

- **Programming language**: The programming language plays a significant role when adopting a new tool, affecting aspects like interoperability, learning curve, and available functionalities. One of the biggest challenges is to find one programming language that offers functionalities to implement the everyday tasks of an immersive visualisation system. For example, Python is well-established among the data analysis community, but it is not the first choice for creating virtual reality applications. Our approach proposes a way to create data analysis services that can use consumed by multiple programming languages.

The present work is our first step towards creating a unified platform for IA applications where ImmVis enables immersive applications written in different programming languages to offer data analysis capabilities. With that, it is possible to bring existing knowledge from data analysis applications and enable new interactions with the data.

3 ImmVis Framework

This chapter provides an overview of the framework development context, describes its technical requirements, and details some aspects of a practical implementation of the framework. In Section 3.1, we briefly describe the development context of the framework, covering the steps we took to create the framework proposal and some requirements we believe are essential to implement it. Section 3.2 provides technical details about a concrete implementation of the framework, and Section 3.3 presents concluding remarks about the framework.

3.1 Development Context

The current research project's initial intention was the evaluation of Immersive technologies inside the data analysis context. To achieve that, we started to investigate and assess some of the tools and frameworks pointed in Chapter 2. Since only the paid ones offered the possibility of performing data analysis tasks, we decided to create from scratch an open-source immersive data analysis platform to support our experiments and serve as a starting point for other researchers in the future.

To create this platform, we decided to adopt Unity since game engines address some concerns from the data visualisation field like rendering and interaction with the users (REINA et al., 2020). However, we found out during the initial steps that implementing the data analysis features inside this environment would require more effort. The main challenge was that C#, the programming language used by Unity, did not have many data analysis libraries options. Furthermore, the few available were not fully compatible with the Unity development environment. Another important aspect was the lack of adoption from the data analysis community, which is critical to consider to make the platform sustainable over time.

With that in mind, we started considering the integration between the Python programming language and Unity to fill this gap. This approach would allow the usage of well-established tools from the data analysis domain. Also, this integration could contribute to the Python ecosystem, as there are not many tools for developing immersive data analysis using the language.

Our first attempt to implement this integration was to use libraries like *python-net*[1] and *Iron Python*[2], which allow .NET applications to use Python and provide unofficial

[1] <http://pythonnet.github.io/> (Accessed on July 22, 2021)
[2] <https://ironpython.net/> (Accessed on July 22, 2021)

support for the Unity game engine. Unfortunately, we only managed to integrate the "pythonnet" library, but the application kept crashing the Unity development environment [3], making both options unviable. Furthermore, investigating how to integrate them made another issue emerges: embedding any library would work fine on applications running inside high-end computers but could be a problem on portable devices like smartphones or standalone virtual reality headsets.

Still believing in the potential of delegating the data analysis to Python, we decided to tackle the problem by making Unity and Python communicate through the network. This approach seemed more interesting as it decouples the visualisation and data analysis pieces, allowing them to run in different machines with different capabilities. This decoupling could also eliminate the concern regarding less-capable devices and enables integration with other technologies like the Unreal game engine or the R programming language.

Given that implementing this type of solution is a challenge, we narrowed the project's scope to a framework that provides network services to enable immersive applications to perform data analysis tasks. We named this framework as *ImmVis*, an abbreviation of the words "Immersive" and "Visualisation". Figure 3.1 shows a conceptual vision of the framework, which we developed with the following requirements in mind:

- Server must provide data analysis services through the network, not requiring IA applications to have to embed any data analysis libraries.
- The data analysis services should be able to support big volumes of data.
- Clients and servers must exchange data seamlessly using a standard data format.
- Developers must implement the communication between clients and servers with the smallest amount of code possible.
- Solution must have a modular architecture where components can be extended or replaced.
- Developers should be able to integrate the framework into immersive applications written in different programming languages.
- The framework must be open-source and free, enabling anyone to use, extend or contribute to the project.
- The project should provide a reasonable amount of documentation and samples to allow other researchers to reproduce the reported results or integrate the framework with their applications.

[3] Bug reports available on <https://github.com/pythonnet/pythonnet/issues/701> and <https://github.com/pythonnet/pythonnet/issues/945> (Accessed on July 22, 2021)

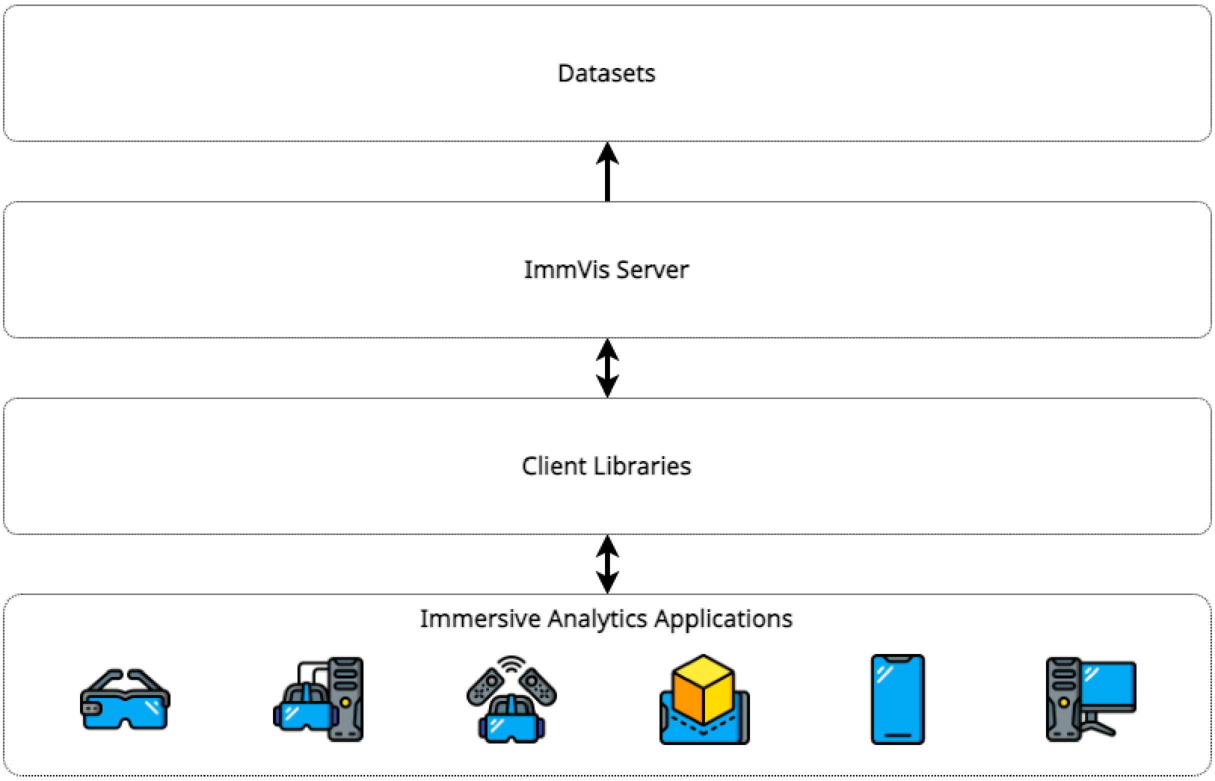


Figure 3.1 – A conceptual vision of the framework where IA applications integrate clients libraries to be able to consume data analysis services from the ImmVis Server. Icons created by Smashicons (2021).

3.2 Implementing ImmVis Framework

After defining the requirements, we decided to create a proof of concept of the framework using the technologies we mentioned before, using Python to implement the server and the Unity game engine to implement a client library. Our first challenge to create this implementation was defining the communication approach and the underlying data format used by that. Our first attempts were to encode the data using JSON and transmit it over the network using technologies like Websockets or REST APIs, which enable the transmission of large amounts of data efficiently. However, they posed an enormous challenge to make Unity and Python understand the data for each new feature, requiring the developers to manually create code to represent and parse the data on each platform. We published the Websockets prototype code on our Github page (PEDROSO, 2021).

Searching alternatives, we found another project using the network to implement the communication between Unity and Python: the "Unity Machine Learning Agents Toolkit" (JULIANI et al., 2018). This toolkit allows the usage of applications created on Unity as environments to train intelligent agents. Although the approach was similar, the authors of this project proposed the usage of the *gRPC* framework to implement the message exchanging between the two environments. After evaluating the framework, we noticed that it solved the communication itself and the data format problem by generating the required code for Unity and Python environments. We considered reusing the code created by Juliani et al. (2018), but its coupling with the Machine Learning domain leads us to use the *gRPC* framework itself on our platform. We provide a brief overview of *gRPC* on the subsection 3.2.1.3.

Figure 3.2 illustrates the architecture of the framework, which is composed of a Python server application and client libraries to be consumed by immersive applications. The server (Section 3.2.1) is responsible by managing datasets (Section 3.2.1.1) and exposing network services to interact with the data (Section 3.2.1.2). These services use the *gRPC* framework (Section 3.2.1.3) to implement the network communication and provide a common data format to facilitate the communication between the server and clients. The server also provides a “Discovery Service”(Section 3.2.1.4) that uses *User Datagram Protocol* (UDP) broadcasts to ease the connection process between client and servers. Finally, another important component from the framework is the client libraries (Section 3.2.2), which IA applications need to integrate or implement to use the framework.

3.2.1 ImmVis Server

The server is the core piece of the framework, concentrating all the data management and data analysis functionalities. We created the server program using the Python programming language and enabled it to run on Windows, macOS, and Linux.

As illustrated in Figure 3.2, the server contains three main components: the data manager, the data services, and the discovery services. This section will detail these components together with a brief introduction of *gRPC* that we used to build the data services layer.

3.2.1.1 Data Managers

Data managers are the components responsible for interfacing with data sources and the data analysis operations. This approach allows the creation of multiple data managers to handle different types of data, like CSV files, images or audio files. Given the diverse nature of data, the framework does not enforce any interface contract for the data managers, allowing them to be simple as using an existing library or creating an infrastructure to support complex data sources.

Data managers only contain code related to their data source, not knowing other aspects of the framework. The data managers functionalities are exposed to clients by the data services, which we describe in Section 3.2.2. It is also important to note that data managers only expose types of data known by the Python environment, leaving the

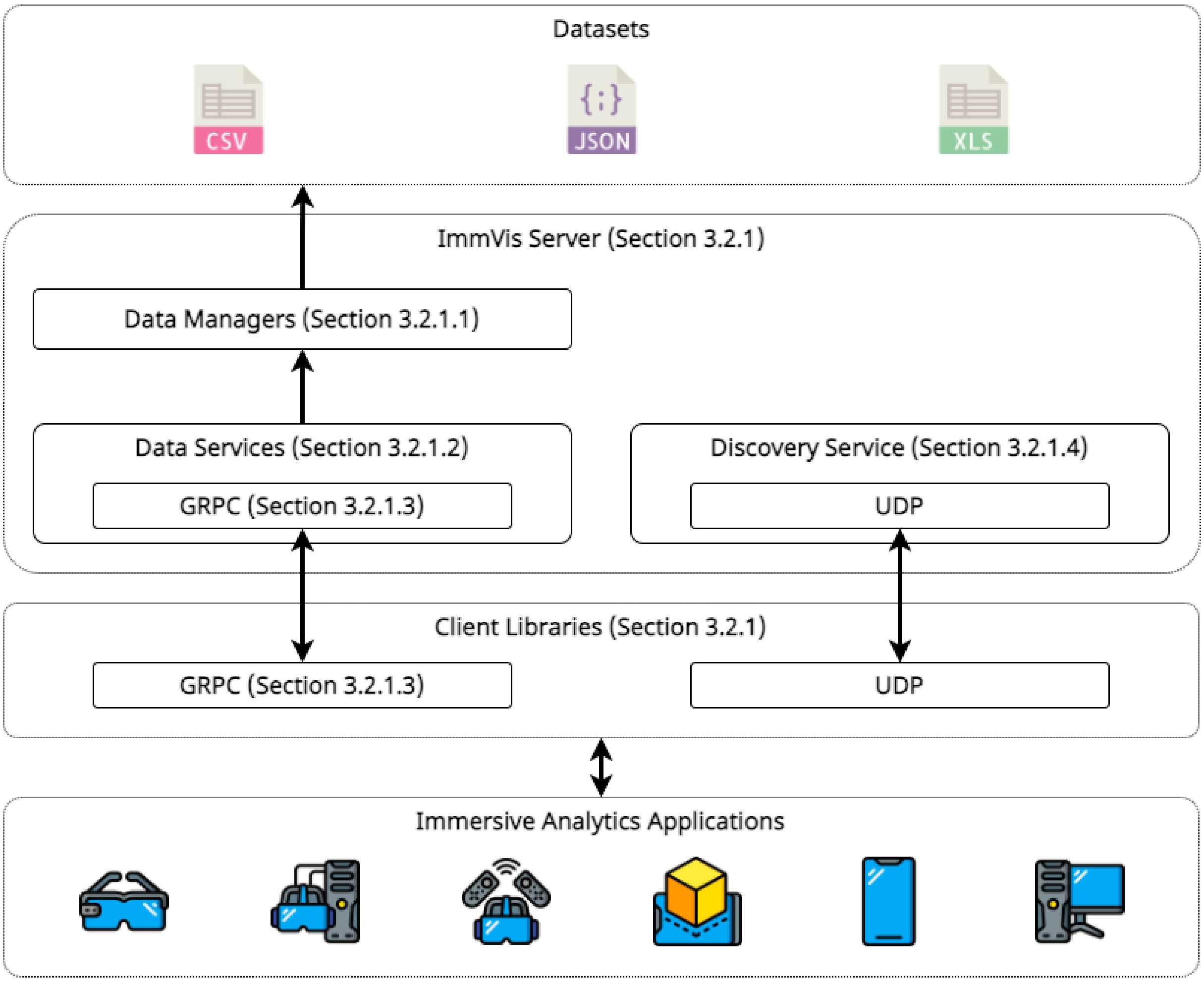


Figure 3.2 – A high-level overview of ImmVis architecture. Icons created by Smashicons (2021).

data services responsible for mapping to the data structures used to communicate with the clients.

The current implementation of the framework includes a single data manager that interacts with the Python libraries *pandas*, *scikit-learn* and *numpy*. Here are the functionalities supported by it:

- Load a local or remote dataset with the following file formats: CSV, JSON and XLS.
- Scan a given directory path from the server and list datasets with the supported formats.
- Infer descriptive statistics about the loaded dataset.
- Normalise the data from the loaded dataset to help clients willing to plot the data.
- Create axis plotting labels to be used by clients.
- Generate a dataset using the *scikit-learn* library.

- Perform a clustering analysis on the loaded dataset using the k-means technique.

3.2.1.2 Data Services

The data services are the components responsible for exposing the functionalities from data managers through the usage of the *gRPC* framework, described in Section 3.2.1.3. Due to the *gRPC* nature and the ImmVis framework architecture, multiple data services can coexist to provide different types of functionalities.

Given that data managers and *gRPC* do not share the same data structure, data services consume the functionalities from data managers and map their response to the format defined using Protocol Buffers. Using the functionality "LoadDataset" from Figure 3.3 as an example, these are the sequence of operations that happen when a client requests to load a dataset:

1. "ImmVisPandas" service receives the request, read the dataset path from the "LoadDatasetRequest"object and request the data manager to load the dataset.
2. The data manager loads the dataset in memory and returns it using the format provided by the *pandas* library.
3. The "ImmVisPandas" service gets the result from the data manager, transforms it into the "DatasetMetadata" and returns it to the client that requested the operation.

This separation between the services and managers aims to enable the reuse of managers code by multiple services and avoid the code from service to grow indefinitely. If the reuse is not needed, developers can create a data service that acts as a data manager.

3.2.1.3 gRPC

Remote Procedure Call (RPC) is a paradigm that has been discussed since 1976 and proposes a mechanism where procedure calls could be invoked over the network the same way of local procedures on a single computer are invoked (BIRRELL; NELSON, 1984). In practice, this approach enables programs to interact with other programs or systems running on different machines or local contexts.

gRPC is an open-source RPC framework released by Google in 2015 to enable the creation of APIs and micro-services in different programming languages and platforms. The framework uses HTTP/2 as part of its foundation, enabling low-latency communication with support to bidirectional streaming. *gRPC* is an recursive acronym which stands to "gRPC Remote Procedure Call" (MARCULESCU, 2015; CHALIN, 2020).

To enable interoperability between platforms and programming languages, *gRPC* relies on Protocol Buffers as its Interface Definition Language (IDL) and message interchange format, requiring developers to create a "proto" file containing the services' definition data structures. With these definitions, the framework can generate the necessary code to allow different programming languages to provide or consume these services.

```
syntax = "proto3";

service ImmVisPandas {
    rpc LoadDataset(LoadDatasetRequest) returns (DatasetMetadata) {};
}

message LoadDatasetRequest {
    string datasetPath = 1;
}

message DatasetMetadata {
    int32 rowsCount = 1;
    int32 columnsCount = 2;
    repeated ColumnInfo columnsInfo = 3;
}

message ColumnInfo {
    Column column = 1;
    repeated DescriptiveStatisticsFeature descriptiveStatisticsFeatures = 2;
}

message Column {
    string columnName = 1;
    string type = 2;
}

message DescriptiveStatisticsFeature {
    string name = 1;
    string value = 2;
    string featureType = 3;
}
```

Figure 3.3 – The Protocol Buffer definition of one of the "load dataset" functionality available on ImmVis

Figure 3.3 illustrates a simple service and related data structures defined using Protocol Buffers. For example, by defining the "LoadDataset" service, a Unity application written in C# can call a procedure with the same name on a server written in Python seamlessly. Subsequently, the server would be able to process the request and answer a "DatasetMetadata" without worrying about converting it to a format recognisable by the client application.

Currently, *gRPC* provides official support for the programming languages: C#, C++, Dart, Go, Java, Kotlin, Node, Objective-C, PHP, Python, and Ruby. In addition, while not listed on the official documentation, the framework Github page also contains projects targeting the .NET Core framework[4] and the programming languages Swift[5] and Javascript[6].

gRPC framework is used on ImmVis Framework to build the data services, allowing the clients to access the framework functionalities without having to implement a Python integrating layer in their solutions. Since the *gRPC* generates code for a diverse range of programming languages, IA practitioners can integrate ImmVis with an extensive range of platforms or solutions. For example, it would be possible to use ImmVis as a data source on IATK, one of the solutions described in Section 2.2.1, as the authors used the Unity game engine to develop it. Defining the functionalities using *Protocol Buffers* also permits the creation of server implementations using other programming languages like R or Julia. We also used *gRPC* to create the alternative architecture for ImmVis described in Section 4.2.

3.2.1.4 Discovery Service

The "Discovery Service" is responsible for helping the clients to find an ImmVis Server running inside a local network. This service aims to enable immersive applications to connect to the server without requesting the users to provide an Internet Protocol (IP) or a previous setup.

We implemented this service by broadcasting messages to the local network using *User Datagram Protocol* (UDP) (POSTEL, 1980). The broadcast starts with the server startup, continuously sending messages containing a string that identifies the server and connection directions that clients should use to connect with the data services. Consequently, the clients need to listen to network broadcasts containing the server identifier to connect to the server automatically. However, using the service is not mandatory, as the framework also enables clients to connect to the server without using it.

This approach does not enable the connection to servers running remotely and do not consider security aspects like authentication, which should be considered in future works if there are specific requirements to support them.

3.2.2 Client Libraries

Client libraries are responsible for providing an integration layer between IA applications and the ImmVis Server. While *gRPC* generates all required to interact with the data services, each platform or programming languages present particularities to use the generated code and connect to the server. The client libraries represent an abstraction layer on top of this code, simplifying the integration with the framework. Having a client

4 <https://github.com/grpc/grpc-dotnet> (Accessed on July 22, 2021)
5 <https://github.com/grpc/grpc-swift> (Accessed on July 22, 2021)
6 <https://github.com/grpc/grpc-web> (Accessed on July 22, 2021)

library for a given platform is not mandatory, as developers can use the *gRPC* tools to do the code generation and write their client libraries, which can be published later as a client library for the missing platform.

The current implementation of ImmVis provides an example of a client library for the Unity game engine. This client library contains a reusable component called *prefab* that developers can add to their projects, configure the network parameters (e.g. use the discovery service or enter the IP address manually) and use the data services using C#. Section 4.1 describes a sample application that used this client library to create a 3D scatterplot.

3.3 Concluding Remarks

Reviewing the design goals listed at the beginning of this chapter, here is a summary of how we achieved each item:

- Modular architecture: all the components described in this chapter were created with a modular approach, allowing their extension or replacement. As mentioned in Section 3.2.1.3, even the server is replaceable by an implementation using other programming languages, keeping compatibility with the whole system.
- Platform and programming language agnostic: since our framework leverages the *gRPC* to implement the data services, developers can integrate it into solutions running in different platforms or written in a variety of programming languages. *gRPC* also satisfies the requirements of having a standard data format, reducing the amount the code required by developers to implement the communication, and avoiding the integration of data analysis libraries inside the IA applications.
- Open-source: the framework is available on the IMDAVI Research Group Github Page under the MIT licence, one of the most popular permissive licenses.
- Documentation: we provide some basic documentation about setting up the environment to use the framework and how to add new features. We also included two code samples with the framework and some utility scripts to ease adding new features. We describe one of the samples in Section 4.1.
- Performance: since *gRPC* leverages HTTP/2, we believe that the framework can benefit from the low latency and bidirectional streaming to enable the transmission of large volumes of data. Preliminary tests showed that, as expected, the amount of time to transmit the data from server to client grows linearly to the amount of data transferred. For example, a dataset containing 50.000 rows and 20 columns

takes an average of 21s to become available to a client running at the same server machine(PEDROSO; COSTA, 2021).

4 Case Studies

This chapter shows two practical implementations using ImmVis, illustrating how to integrate the framework to immersive applications, highlighting relevant aspects and contributing with reference code for future developers and enthusiasts.

Section 4.1 describes the implementation of an immersive application that allows users to plot data on a 3D scatterplot and use the k-means clustering technique to analyse it. This section also provides a brief overview of some challenges beyond the initial scope, including minor contributions of the present work regarding using a game engine as a data visualisation tool. Finally, Section 4.2 covers an experimental version of the framework that we create to improve the experience when using ImmVis inside Jupyter Notebooks.

The code for all the implementations discussed in this chapter is available on the IMDAVI Github webpage (PEDROSO; COSTA, 2021a).

4.1 Data Analysis on Unity applications

Good documentation is a fundamental piece when creating a framework for developers, as it facilitates its adoption and provides more insights into its capabilities. One of the most typical ways of documenting a framework is creating code samples that showcase the functionalities available in a practical scenario. The Unity client library ships two code samples to help developers get up to speed about using it inside the Unity game engine: a demonstration of essential framework's APIs and an immersive scatterplot sample that illustrates a more advanced use case. This section focuses on the advanced sample and some of the challenges faced during its development process.

The sample consists of a Unity 3D VR-ready application that allows users to plot data into a 3D scatterplot using a menu inside the virtual environment, exercising some of the framework's capabilities. Figure 4.1 shows the sequence of menus that the user needs to go through to plot a scatterplot or execute a k-means analysis inside the virtual environment, where we will detail each step ahead. A video of this sample is available on our Youtube channel to illustrate better its usage (PEDROSO; COSTA, 2021b).

As described in Chapter 3, ImmVis client applications need to connect to a Python server through the network, which can pose some usability issues as typing an IP address inside the immersive environment can be a little bit challenging. Our framework addresses that by providing a way to detect if a server is available on a local network and establishing the connection automatically. The sample uses this characteristic, allowing the

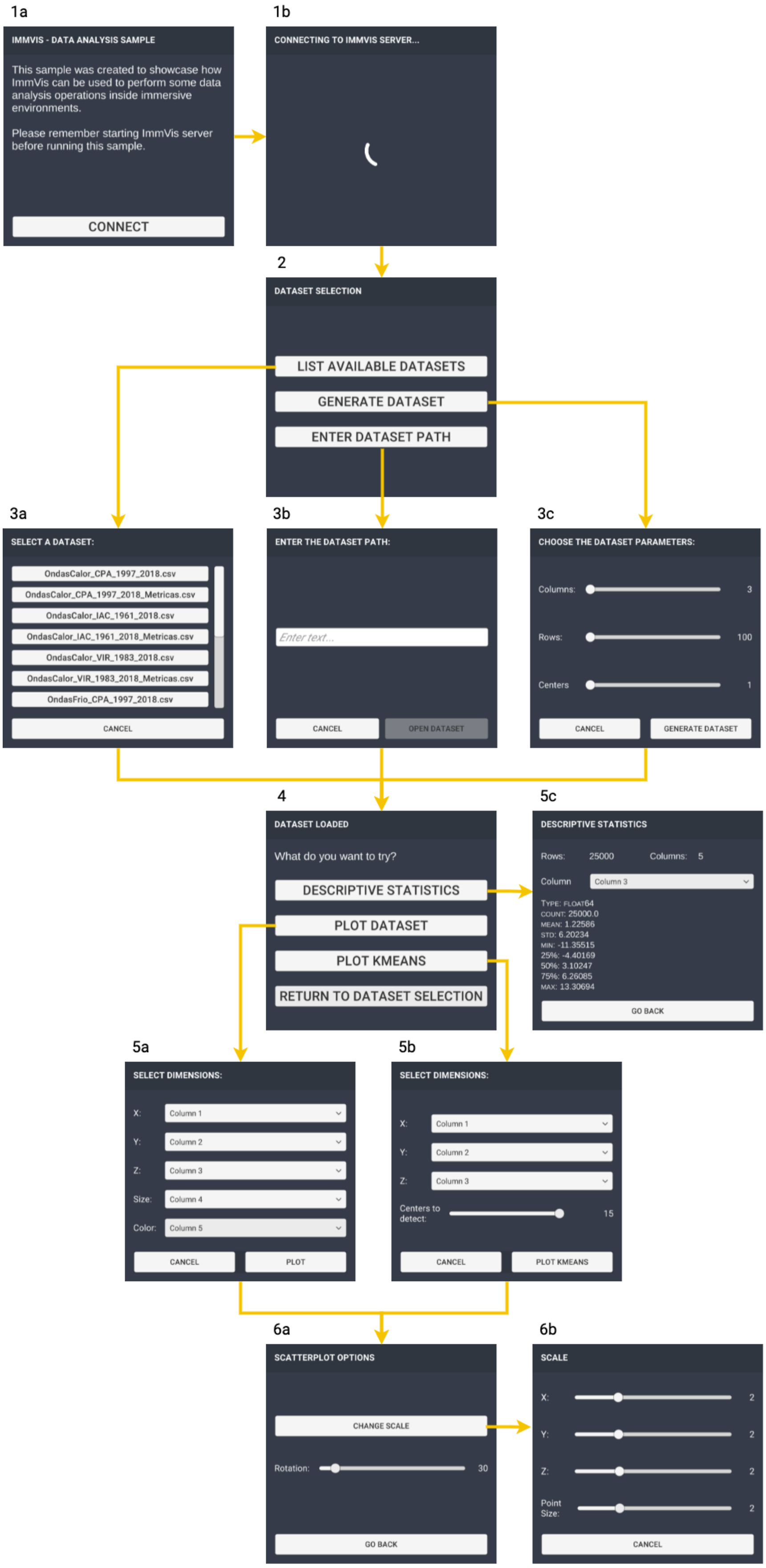


Figure 4.1 – A diagram representing the menu flow from the advanced sample that is bundled with ImmVis Unity client library.

user to click on a button and wait for the connection with the server (Figure 4.1, menus "1a" and "1b').

After connecting to the server, the user will choose how the app will load a dataset among three options (Figure 4.1, menu "2"). The first one (Figure 4.1, menu "3a") provides a list of datasets available on a pre-specified folder known by the server, where the users can copy their datasets previously. The second one (Figure 4.1, menu "3c") requests the server to generate a sample dataset using the user's parameters, using the Python's *sklearn* library. Finally, the last option (Figure 4.1, menu "3b") allows the user to type a path or an URL to the dataset, enabling users to upload their datasets to the internet or using existing ones. All the options will use Python's *pandas* library to load the datasets and allow users to perform the same set of tasks described ahead.

The next step happens after the user selects a dataset, when the application will present three options to the user (Figure 4.1, menu "4"):

- **Descriptive Statistics**: presents some characteristics of the dataset and its dimensions, as illustrated on Figure 4.1, menu "5c".
- **Plot Dataset**: requests the user to select from three to five dimensions to be plotted (Figure 4.1, menu "5a"). After the user clicks on "Plot", the server will return a subset of the data containing the normalised values from the selected columns, and the client will plot it as shown in Figure 4.2. Each point of the dataset will be encoded using the following characteristics: position on the X-axis, position on the Y-axis, position on the Z-axis, point size and point colour.
- **Plot KMeans**: this option lets the user select some parametrisation to run a k-means analysis on the server. After the user clicks on the plot, the server performs the analysis and returns a normalised version of the dataset where the point colour will encode which cluster the point pertains. The analysis result also contains a smaller dataset containing the centroids of each cluster. Figure 4.2 illustrates a scatterplot representing the result of a k-means analysis.

After plotting the data, the application shows a menu to allow the user to change the rotation and scale from the plotted data (Figure 4.1, menus "6a" and "6b").

To implement this sample, we needed to overcome some challenges that, despite not using the framework directly, can be reused by other visualisation projects created with Unity:

- **Menu Navigation System**: The menu system shown in Figure 4.1 needed to be created from scratch as we could not find an approach that was free and offered a

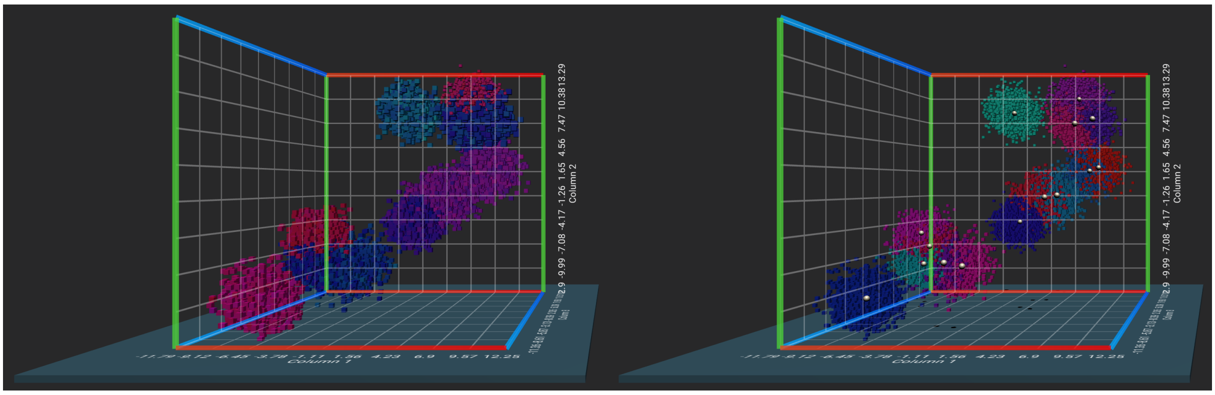


Figure 4.2 – Two screenshots of the data analysis sample. The left shows a dataset where we encoded five dimensions to the spacial position (X, Y, and Z), point size and colour. The screenshot on the right shows the result of a k-means analysis of the same dataset. Since the generated dataset has clusters, the figure on the right gives a hint of the analysis performed on the second screenshot.

simple API to use. Our menu system was implemented using a stack data structure and enable developers to create simple menus with rudimentary support to back navigation. This system could be extracted to a separate library and be used in different contexts to solve similar problems.

- **Network transmission optimisation**: one of our goals during the sample implementation was to demonstrate that the framework was able to handle large amounts of data, which can pose some challenges for the communication between the ImmVis server and its clients. We tried different paths to optimise this communication during the implementation phase, from creating *gRPC* streams to consider different approaches to create the Protobuf objects. Our basic *gRPC* knowledge points out that we have a good solution, but we believe there is space for investigation and improvements. For example, preliminary tests using an Oculus Quest device showed that a dataset with 50.000 lines takes about 14.5 seconds to be transmitted using the framework (PEDROSO; COSTA, 2021).

- **Implementing a Scatterplot on Unity**: We also intended to provide a solution to render considerable amounts of data on the Unity game engine during the initial stages of the current work since it does not provide tooling for this specific domain. Along the way, we shifted the focus to the current ImmVis state but wanted to apply some of the acquired knowledge. The advanced sample uses one of the techniques investigated, "GPU instancing", to handle an immense amount of points. We extended an existing open-source sample[1] that uses this technique to implement it, connect it to ImmVis data format, and add custom functionalities. The main things added were the usage of point size to encode a dataset dimension, the transformation matrix

[1] <https://github.com/noisecrime/Unity-InstancedIndirectExamples> (Accessed on July 22, 2021)

manipulation to rotate and scale the graph, and the possibility of having multiple graphs with different transparency levels to plot the k-means analysis result. We also implemented visual indicators like the axis labels and the plane with the axis's dynamic grids.

This sample is bundled together with the ImmVis Unity client library, and its source code is also available on Github.

4.2 Jupyter Notebooks Integration

Jupyter Notebooks are a well-established medium among data science practitioners and the scientific community. It allows sharing discoveries and knowledge by creating interactive documents with code, visualisations, and equations. Given that Python is one of the main languages supported by Jupyter Notebooks, ImmVis can run inside them from the beginning. However, this approach presented some network problems during our tests because ImmVis and Jupyter are both "server" applications and can create conflicts like using the same network port.

As described in Chapter 3, the architecture of the ImmVis framework proposes the creation of client applications being the point of control of the system, sending commands to a server that processes the data and returns the results to be displayed. This type of approach is directly against the interactive and imperative nature of Jupyter, allowing users to input commands on a web application and see the results right after that. Another challenge would be creating the proper user experience to enable the users to go back and forth between Jupyter and the immersive environment.

After experimenting with Jupyter and ImmVis, we created an experimental version of the framework that allows the connection of a Jupyter Notebook to an ImmVis "display server", an approach inspired by Python visualisation tools like *Matplotlib* and *seaborn*. In practice, this approach makes the interaction with ImmVis more familiar to Jupyter users as they can continue using the conventional tools to explore and wrangle the data and then connect to ImmVis to plot their visualisations inside an immersive environment. Figure 4.3 illustrates a sample code running inside the Jupyter Notebook environment. A video with the demonstration is also available on our Youtube channel (PEDROSO; COSTA, 2021b).

While this version of ImmVis is not the focus of the present work, we believe that the approach is applicable in scenarios where the user's goal is to explore the data without creating a whole immersive application. We also envision other types of interactions for the approach, like displaying the notebook inside the immersive environment or enabling two users to collaborate during the data analysis process, one "piloting" the Jupyter notebook

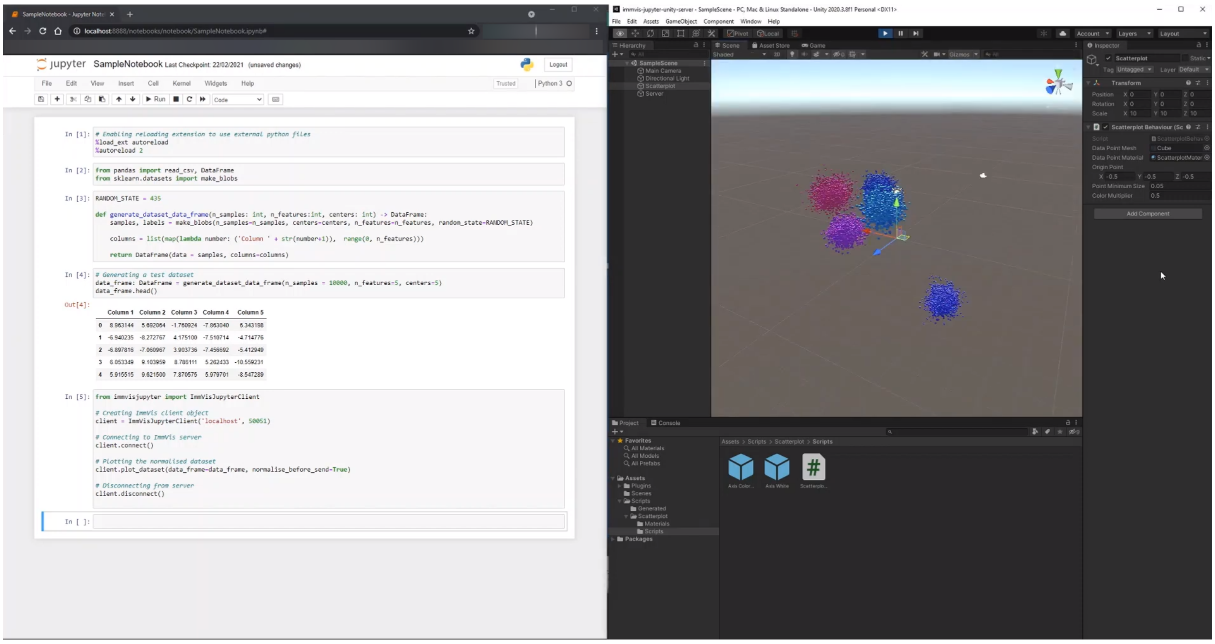


Figure 4.3 – A screenshot from a Jupyter Notebook (left) plotting data into an ImmVis Display Server (right) running inside a Unity application.

and another inside the immersive environment.

The implementation of ImmVis for Jupyter includes a sample notebook and a display server created for the Unity game engine, reusing the scatterplot implementation described in section 4.1. It is important to note that it is possible to create display servers in other platforms and programming languages, as the communication between clients and server also uses *gRPC*.

ImmVis for Jupyter is also open source, and its code is available on Github as well (PEDROSO; COSTA, 2020b; PEDROSO; COSTA, 2020a).

4.3 Concluding Remarks

This chapter describes the outcomes of an empirical exploration process that we did during ImmVis development. Our initial intention was to create these tools and then perform experiments with human subjects to evaluate data visualisation efficiency inside immersive environments and discover pain points during the development process. Unfortunately, by the time we finished our prototype, the COVID-19 pandemic had hit the world, making us change the current work's focus to something that would not require experimenting with people. Nevertheless, we believe that the evaluations could bring more value to the framework and, in the future, benefit from the framework's existence and the case studies presented here.

5 Conclusion and Future Work

This work presented an open-source framework to enable IA applications to use data analytics functionalities from the Python ecosystem. The framework uses *gRPC* to establish a standard data format and allow network communication between a data analysis server and client applications written in different programming languages.

The framework comes to complement the existing IA tools and frameworks, including the ones described in Chapter 2. We intended to enable them to integrate data analysis functionalities and explore new ways to interact with data inside the immersive environment.

The proposal of this framework came after exploring some alternatives to integrate data analysis functionalities into immersive applications. Our first attempt was to embed Python inside the applications, which did not succeed due to the technologies' limitations. Our second approach was to create data analysis services that enabled applications to consume these functionalities through the network, offloading the responsibility to more capable hardware and the proper technologies. To implement these services, we explored the usage of technologies like REST and Websockets, which had an impressive performance but required more development efforts from final users. Finally, we decided to adopt the *gRPC* framework, which enabled us to implement the data services and make them available for different platforms and programming languages. After all the exploring and tests, we established the framework's requirements to enable other researchers to understand our choices and explore other technologies for the same purpose.

With the framework requirements in hand, we created a prototype and a sample application that serves both as documentation and a proof-of-concept application of the ImmVis Framework, showing its capabilities and enabling the reproduction of the results reported in this work. We published the source code of everything that we developed during the development of the present work.

The framework is a direct byproduct of the initial research goal, which was to investigate the effectiveness of IA applications to explore data. During the initial stages of this research, we identified that the field lacks the methodology to do this type of evaluation, which made us dedicate a good portion of our time to find answers to this challenge. During this phase, we planned to execute some experiments with people using applications created using the framework. To support these experiments, we researched how to generate synthetic data and render vast amounts of data, contributing to some aspects of the framework's prototype. However, after having the Ethics Committee Approval (CAAE 6679619.7.0000.5404) to perform the experiments, we needed to change plans due to the COVID-19 pandemics.

During the final stages of our project, we started exploring using our framework prototype inside Jupyter Notebooks, a well-known tool amongst data analysis practitioners. While our prototype was always able to run inside this environment, we noticed that it was not suitable for the interactions provided by Jupyter Notebooks. Therefore, as a side project, we created an alternative version of the framework that turns an IA application into an "ImmVis Display Server", enabling users to plot data from a Jupyter Notebook into the immersive environment. The source code of this project is also available as open-source so future researchers can explore this alternative.

We consider that our framework proposal is a first step towards creating a standard platform for IA practitioners to build their applications and evaluate emergent technologies and innovative interactions with data analysis inside immersive environments.

5.1 Future Work

As future work, we envision the following relevant and immediate opportunities:

- **Development of client libraries for other technologies**: as described in Section 3.2.2, the framework currently only offer the Unity client library. While this should be enough to cover different devices and platforms, it would be interesting to cover other game engines or even platforms. For example, creating a client library for web pages to integrate with WebXR[1] could support researchers to experiment with IA applications inside web browsers.
- **Framework expansion to other domains**: the current version of the framework only includes a data manager capable of handling tabular datasets. The framework's architecture allows its extension to support other types of data like images and audio. To achieve that, developers only need to create data managers that can handle this type of data and expose the functionalities through a data service. Another possibility to explore is creating an interface contract for different data types to enable developers to explore the integration with libraries that can handle those alternative data types.
- **Formal analysis of network performance**: as mentioned in Section 3.3, we did preliminary performance tests and obtained satisfactory results. Still, we believe that it would be interesting to perform a formal analysis and investigate if it would be possible to optimise the network connection. Additionally, it would be interesting also to evaluate Apache Arrow Flight, another framework built on top of *gRPC*

[1] <https://immersive-web.github.io/webxr/> (Accessed on July 22, 2021)

to transport large data sets through high-performance data services (MCKINNEY, 2019; FOUNDATION, 2019).

- **Perform a user study with an IA application that used ImmVis Framework**: One of our goals before the COVID-19 pandemics was to create an IA application and run experiments with users to evaluate the effectiveness of this type of application. Due to the current world scenario and the involved risks, we needed to shift the focus of the present work to the current state. Nevertheless, we believe this kind of study is essential and should be one area to be explored by other researchers.

- **Evolution of the Jupyter Notebooks implementation**: in Section 4.2 we described an experimental version of ImmVis framework that enables Jupyter Notebooks users to plot data similarly to other Python visualisation languages. We believe that this implementation could be evaluated and extended to allow researchers to explore different scenarios and use cases.

Bibliography

BIRRELL, A. D.; NELSON, B. J. Implementing Remote Procedure Calls. *ACM Transactions on Computer Systems (TOCS)*, ACM New York, NY, USA, v. 2, n. 1, p. 39–59, 1984. Cited in page 28.

BUTCHER, P. W. S.; JOHN, N. W.; RITSOS, P. D. VRIA: A Web-based Framework for Creating Immersive Analytics Experiences. *IEEE Transactions on Visualization and Computer Graphics*, IEEE, 2020. Cited 2 times in pages 11 and 20.

CHALIN, P. *Introduction to gRPC*. 2020. <https://grpc.io/docs/what-is-grpc/introduction/>. Accessed on July 22, 2021. Cited in page 28.

CHANDLER, T. et al. Immersive Analytics. In: *2015 Big Data Visual Analytics (BDVA)*. [S.l.: s.n.], 2015. p. 1–8. Cited 2 times in pages 14 and 16.

CHANDLER, T.; MORGAN, T.; KUHLEN, T. W. Exploring Immersive Analytics for Built Environments. In: *Immersive Analytics*. [S.l.]: Springer, 2018. p. 331–357. Cited in page 18.

CHEN, S.; LIN, L.; YUAN, X. Social Media Visual Analytics. *Computer Graphics Forum*, v. 36, n. 3, p. 563–587, 2017. Cited in page 16.

CORDEIL, M. *(IATK: An Immersive Analytics Toolkit - YouTube*. 2019. <https://www.youtube.com/watch?v=qiWIPePTsOA>. Accessed on July 22, 2021. Cited in page 20.

CORDEIL, M. et al. IATK: An Immersive Analytics Toolkit. In: IEEE. *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. [S.l.], 2019. p. 200–209. Cited 3 times in pages 11, 18, and 19.

CORDEIL, M. et al. ImAxes: Immersive Axes as Embodied Affordances for Interactive Multivariate Data Visualisation. In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. [S.l.: s.n.], 2017. p. 71–83. Cited 2 times in pages 11 and 18.

CUI, W. Visual Analytics: A Comprehensive Overview. *IEEE Access*, IEEE, v. 7, p. 81555–81573, 2019. Cited in page 16.

CZAUDERNA, T. et al. Immersive Analytics Applications in Life and Health Sciences. In: *Immersive Analytics*. [S.l.]: Springer, 2018. p. 289–330. Cited in page 18.

DATAR. *DatAR: An Immersive Literature Exploration Environment for Neuroscientists / AIVR conference 2020 - YouTube*. 2020. <https://www.youtube.com/watch?v=PnOPECRNc_w>. Accessed on July 22, 2021. Cited in page 17.

DONALEK, C. et al. Immersive and Collaborative Data Visualization Using Virtual Reality Platforms. In: IEEE. *Big Data (Big Data), 2014 IEEE International Conference on*. [S.l.], 2014. p. 609–614. Cited 2 times in pages 11 and 18.

DWYER, T. et al. Immersive Analytics: An Introduction. In: *Immersive Analytics*. [S.l.]: Springer, 2018. p. 1–23. Cited 2 times in pages 11 and 16.

FEDERICO, P. et al. The Role of Explicit Knowledge: A Conceptual Model of Knowledge-Assisted Visual Analytics. In: IEEE. *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*. [S.l.], 2017. p. 92–103. Cited in page 11.

FILONIK, D. et al. Glance: Generalized geometric primitives and transformations for information visualization in AR/VR environments. In: *Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry-Volume 1*. [S.l.: s.n.], 2016. p. 461–468. Cited in page 18.

FOUNDATION, A. S. *Arrow Flight RPC — Apache Arrow v4.0.0*. 2019. <https://arrow.apache.org/docs/format/Flight.html>. Accessed on July 22, 2021. Cited in page 41.

FULMER, W. et al. ImWeb: Cross-Platform Immersive Web Browsing for Online 3D Neuron Database Exploration. In: *Proceedings of the 24th International Conference on Intelligent User Interfaces*. [S.l.: s.n.], 2019. p. 367–378. Cited in page 18.

GIESELER, W. *The Creation of LookVR*. 2017. <https://looker.com/blog/creation-of-lookvr>. Accessed on July 22, 2021. Cited in page 18.

GORODOV, E. Y.; GUBAREV, V. V. Analytical Review of Data Visualization Methods in Application to Big Data. *JECE*, Hindawi Publishing Corp., New York, NY, United States, v. 2013, p. 22:2–22:2, Jan. 2013. ISSN 2090-0147. Cited in page 11.

HANULA, P. et al. DarkSky Halos: Use-Based Exploration of Dark Matter Formation Data in a Hybrid Immersive Virtual Environment. *Frontiers in Robotics and AI*, Frontiers, v. 6, p. 11, 2019. Cited in page 17.

JULIANI, A. et al. Unity: A General Platform for Intelligent Agents. *arXiv preprint arXiv:1809.02627*, 2018. Cited 2 times in pages 25 and 26.

KARPATNE, A. et al. Theory-guided Data Science: A New Paradigm for Scientific Discovery from Data. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 29, n. 10, p. 2318–2331, 2017. Cited in page 11.

KEIM, D.; KOHLHAMMER, J.; ELLIS, G. *Mastering The Information Age – Solving Problems with Visual Analytics*. 1st. ed. [S.l.]: Eurographics Association, 2010. Cited 4 times in pages 8, 11, 14, and 15.

KITWARE. *VTK Flavors*. 2016. <https://vtk.org/flavors>. Accessed on July 22, 2021. Cited in page 19.

KOEBEL, K.; AGOTAI, D.; ÇÖLTEKIN, A. Exploring Cultural Heritage Collections in Immersive Analytics: Challenges, Benefits, and a Case Study Using Virtual Reality. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Copernicus GmbH, v. 43, p. 599–606, 2020. Cited in page 18.

KWON, B. C. et al. RetainVis: Visual Analytics with Interpretable and Interactive Recurrent Neural Networks on Electronic Medical Records. *IEEE Transactions on Visualization and Computer Graphics*, v. 25, n. 1, p. 299–309, 2019. Cited in page 16.

LAWTON, B. et al. SIERA: The Seismic Information Extended Reality Analytics Tool. In: *Companion Proceedings of the 2020 Conference on Interactive Surfaces and Spaces*. [S.l.: s.n.], 2020. p. 73–77. Cited in page 18.

LIN, H. et al. RCLens: Interactive Rare Category Exploration and Identification. *IEEE Transactions on Visualization and Computer Graphics*, v. 24, n. 7, p. 2223–2237, 2018. Cited in page 16.

LIN, T. et al. SportsXR–Immersive Analytics in Sports. *arXiv preprint arXiv:2004.08010*, 2020. Cited in page 17.

LOOKER. *VR Data Analytics with LookVR*. 2017. <https://www.youtube.com/watch?v=u76ww3NJFgE&t=126s>. Accessed on July 22, 2021. Cited 2 times in pages 8 and 19.

MARCULESCU, M. *Introducing gRPC, a new open source HTTP/2 RPC Framework*. 2015. <https://developers.googleblog.com/2015/02/introducing-grpc-new-open-source-http2.html>. Accessed on July 22, 2021. Cited in page 28.

MARR, B. *Big Data: 20 Mind-Boggling Facts Everyone Must Read*. [S.l.]: Forbes Magazine, 2015. <https://www.forbes.com/sites/bernardmarr/2015/09/30/big-data-20-mind-boggling-facts-everyone-must-read>. Accessed on July 22, 2021. Cited in page 11.

MARRIOTT, K. et al. Immersive Analytics: Time to Reconsider the Value of 3D for Information Visualisation. In: *Immersive Analytics*. [S.l.]: Springer, 2018. p. 25–55. Cited in page 16.

MARTORELL-MARUGÁN, J. et al. DatAC: A visual analytics platform to explore climate and air quality indicators associated with the COVID-19 pandemic in Spain. *Science of The Total Environment*, Elsevier, v. 750, p. 141424, 2021. Cited 3 times in pages 8, 14, and 15.

MCCORMACK, J. et al. Multisensory Immersive Analytics. In: *Immersive Analytics*. [S.l.]: Springer, 2018. p. 57–94. Cited in page 16.

MCKINNEY, W. *Introducing Apache Arrow Flight: A Framework for Fast Data Transport*. 2019. Https://arrow.apache.org/blog/2019/10/13/introducing-arrow-flight. Accessed on July 22, 2021. Cited in page 41.

MERINO, L. et al. Toward Agile Situated Visualization: An Exploratory User Study. In: *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. [S.l.: s.n.], 2020. p. 1–7. Cited in page 21.

MUNZNER, T. *Visualization Analysis and Design*. [S.l.]: AK Peters/CRC Press, 2014. Cited in page 16.

NEBELING, M. et al. MRAT: The Mixed Reality Analytics Toolkit. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. [S.l.: s.n.], 2020. p. 1–12. Cited in page 21.

O'LEARY, P. et al. Enhancements to VTK enabling scientific visualization in immersive environments. In: IEEE. *2017 IEEE Virtual Reality (VR)*. [S.l.], 2017. p. 186–194. Cited 3 times in pages 8, 19, and 20.

PEDROSO, F. *ImmVis Websockets Server*. 2021. <https://github.com/imdavi/ImmVisWebsocketsServerPython>. Accessed on July 22, 2021. Cited in page 25.

PEDROSO, F.; COSTA, P. ImmVis: Bridging Data Analytics and Immersive Visualisation. In: INSTICC. *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 1: IVAPP,*. [S.l.]: SciTePress, 2021. p. 181–187. ISBN 978-989-758-488-6. Cited 2 times in pages 32 and 36.

PEDROSO, F.; COSTA, P. D. P. *ImmVis Display Client Jupyter*. 2020. <https://github.com/imdavi/ImmVisDisplayClientJupyter>. Accessed on July 22, 2021. Cited in page 38.

PEDROSO, F.; COSTA, P. D. P. *ImmVis Display Server Unity*. 2020. <https://github.com/imdavi/ImmVisDisplayServerUnity>. Accessed on July 22, 2021. Cited in page 38.

PEDROSO, F.; COSTA, P. D. P. *IMDAVI (IMmersive DAta VIsualization)*. 2021. <https://github.com/imdavi>. Accessed on July 22, 2021. Cited in page 33.

PEDROSO, F.; COSTA, P. D. P. *ImmVis Framework Youtube Playlist*. 2021. <https://www.youtube.com/playlist?list=PLZ0OELoN0Q7ohPmQIqAWeh58nHOL65rX1>. Accessed on July 22, 2021. Cited 2 times in pages 33 and 37.

POSTEL, J. *RFC0768: User Datagram Protocol*. [S.l.]: RFC Editor, 1980. Cited in page 30.

PREIM, B.; LAWONN, K. A survey of visual analytics for public health. In: WILEY ONLINE LIBRARY. *Computer Graphics Forum*. [S.l.], 2020. v. 39, n. 1, p. 543–580. Cited in page 14.

QUAMMEN, C. Scientific Data Analysis and Visualization with Python, VTK, and ParaView. In: *Proceedings of the 14th Python in Science Conference (SciPy 2015)*. [S.l.: s.n.], 2015. p. 32–39. Cited 2 times in pages 19 and 20.

REINA, G. et al. The Moving Target of Visualization Software for an Increasingly Complex World. *Computers & Graphics*, Elsevier, v. 87, p. 12–29, 2020. Cited 3 times in pages 19, 21, and 23.

SAIFEE, M. et al. VR-Viz: Visualization System for Data Visualization in VR. 2018. Cited in page 21.

SATYANARAYAN, A. et al. Vega-Lite: A Grammar of Interactive Graphics. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2017. Accessed on July 22, 2021. Available at: <http://idl.cs.washington.edu/papers/vega-lite>. Cited in page 19.

SHARKEY, K.; BENNETT, J.; PEBAY, P. *The VTK Parallel Statistics Module - Kitware Blog*. 2013. <https://blog.kitware.com/the-vtk-parallel-statistics-module/>. Accessed on July 22, 2021. Cited 2 times in pages 19 and 20.

SICAT, R. et al. DXR: A Toolkit for Building Immersive Data Visualizations. *IEEE Transactions on Visualization and Computer Graphics*, IEEE, v. 25, n. 1, p. 715–725, 2018. Cited 4 times in pages 11, 18, 19, and 20.

SMASHICONS. *Free icons created by Smashicons | Flaticon*. 2021. <https://www.flaticon.com/br/autores/smashicons>. Accessed on July 22, 2021. Cited 3 times in pages 8, 25, and 27.

TADEJA, S.; SESHADRI, P.; KRISTENSSON, P. O. AeroVR: An Immersive Visualization System for Aerospace Design and Digital Twinning in Virtual Reality. *The Aeronautical Journal*, Cambridge University Press, v. 124, n. 1280, p. 1615–1635, 2020. Cited 2 times in pages 16 and 17.

THOMAS, J. J.; COOK, K. A. A Visual Analytics Agenda. *IEEE Comput. Graph. Appl.*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 26, n. 1, p. 10–13, Jan. 2006. ISSN 0272-1716. Cited in page 14.

TROOST, I. et al. DatAR: An Immersive Literature Exploration Environment for Neuroscientists. In: IEEE. *2020 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. [S.l.], 2020. p. 55–56. Cited in page 17.

VIRTUALITICS, I. *Virtualitics Homepage*. 2015. Https://www.virtualitics.com. Accessed on July 22, 2021. Cited in page 18.

WANG, J. et al. DQNViz: A Visual Analytics Approach to Understand Deep Q-Networks. *IEEE Transactions on Visualization and Computer Graphics*, v. 25, n. 1, p. 288–298, 2019. Cited in page 16.

YE, S. et al. ShuttleSpace: Exploring and Analyzing Movement Trajectory in Immersive Visualization. *IEEE Transactions on Visualization and Computer Graphics*, IEEE, 2020. Cited in page 17.

YIM, D. et al. NiwViw: Immersive Analytics Authoring Tool. In: *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*. [S.l.: s.n.], 2018. p. 425–428. Cited in page 21.

ZHAO, X. et al. iForest: Interpreting Random Forests via Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, v. 25, n. 1, p. 407–416, 2019. Cited in page 16.