DIGITAL HERITAGE (2025)
S. Campana, D. Ferdani, H. Graf, G. Guidi, Z. Hegarty, S. Pescarin, and F. Remondino (Editors)

Procezo: Data Processing Services for 3D Analytics

G. Gosti† [1], S. Menconero[1], C.F. Amadei[1], B. Fanini [1]

[1] National Research Council (CNR), Institute of Heritage Science, Italy

Abstract

The recent spread of metaverse technologies has propelled the digitalization of cultural heritage, particularly the production of 3D models of historical artifacts and sites. This context presents new challenges and opportunities, among these arises the possibility of investigating the design of public engagement and cultural dissemination initiatives through the analytical study of user behavior. With this objective, we developed Procezo, a modular data analysis suite that facilitates the processing and aggregation of user experience data via an easy-to-use web interface, specifically engineered for cultural heritage applications. Indeed, with immersive XR devices, motion-tracking tools, sensors, and online web applications, we can easily record users' experiences in virtual or real environments. From these recorded experiences, with Procezo's specifically developed web-based analytics, we may obtain crucial insights into user interaction patterns. Procezo is part of a larger pilot developed under the H2IOSC project named "Interlumo". The pilot is divided into three stages: data capture (Kapto), data processing (Procezo), and data inspection (Merkhet). These stages are based on a strong modular design, both at the logical and software levels. The logical separation allows the implementation of these stages together or separately, and the software separation allows us to run the stages on separate dedicated servers. This modularity allows for greater reuse and scalability. We demonstrate the application of Procezo in data cleaning and preprocessing protocols, as well as its implementation for machine learning (ML) algorithms for pattern discovery, specifically through kernel density estimation (KDE), a reliable non-parametric density estimation methodology. Our implementation is based on a graphical web interface that allows analysts to share and compare different machine learning (ML) pipelines. The presented suite improves the quality and efficiency of the analysis process and enables collaboration between domain and analytics experts. Under the H2IOSC project, we assess Procezo on visitors' experiences exploring a virtual reproduction of Cerveteri Etruscan Tomb, which were captured during remote public exhibits and dissemination events. This approach can be easily applied to several case studies, ranging from interactive installations, to online applications, with the objective of accelerating the detection of interaction patterns.

CCS Concepts

• ***Mathematics of computing*** → ***Exploratory data analysis;*** • ***Information systems*** → ***Asynchronous editors;*** *Data analytics; Data mining;* • ***Human-centered computing*** → *Systems and tools for interaction design;*

1. Introduction

The latest improvements in head-mounted display (HMD) technologies and the large investments by major tech companies have boosted the spread of "metaverse" applications. Even though the initial enthusiasm for the possibility of a broad range of applications is diminishing, some of the more focused applications are gaining the attention of researchers from different fields [DHB*22, FFD*21]. Specifically, in the digital heritage context, this process has propelled cultural heritage's digitization, particularly the development of 3D models of historical artifacts and sites [BMD*24]. 3D models and archaeological surveys enable a range of innovative applications including virtual learning experiences, archaeological conservation, and the systematic documentation of heritage asset deterioration [BMD*24].

This process, together with the emergence of the new deep neural network approaches that have revolutionized the scope of AI applications, opens up new challenges and opportunities [FG24]. One of these challenges is the opportunity to improve and develop digital heritage user experiences through a data-driven process. Head-mounted displays (HMDs), eye tracking devices, brain–computer interface (BCI) headsets, and motion tracking equipment, as well as wearable sensors or devices used in digital heritage interactive experiences, allow us to capture massive volumes of data. Unfortunately, the detection of relevant information on visitor interaction patterns and troubleshooting requires complex explorative analysis [KAA*21]. Nevertheless, if carried out systematically, this explorative analytics may allow us to determine the visitor experience quality, determining among the other possibilities, engagement,

† giorgio.gosti@cnr.it

DOI: 10.2312/dh.20253099

satisfaction, and retention [CCC*24, PCR*25]. This approach will allow us to develop new design best practices tailored to improve the visitor experience, otherwise known as user experience (UX) analytics [KC20]. This process may also impact cultural heritage accessibility, taking into account the visitors' diversity and their varying needs and perspectives [KK23, SPK20].

H2IOSC (https://www.h2iosc.cnr.it/) is a federated cluster of four research infrastructures with operating nodes across Italy. Procezo's suite is the second stage of the "Interlumo" immersive analytic modular pipeline, which is itself exposed as a cross-domain pilot in H2IOSC. Interlumo is developed in task 7.7 of the H2IOSC project, under the E-RIHS infrastructure. The Procezo step processes and aggregates time series data, which we call records, through ML models and tools. The other two components are Kapto and Merkhet. Kapto is the first component, its purpose is to capture user session records from different sources: XR devices, sensors, web applications, and other live sessions. Merkhet is the last component, it is designed for the visualization and integration of the aggregated data in the source 3D model, where the data was captured. As this pipeline is highly modular, each component can be used separately or reconfigured in pipelines with other applications. Here we focus only on Procezo's suite.

The objective of the Procezo suite is to offer an User Interface (UI) and a series of methods that will allow the cooperation of UI/UX specialists, machine learning experts, curators, exhibit designers, neuroscientists, and project designers to develop and implement data-driven cultural heritage user experiences. Procezo's modular data processing suite offers easy-to-use interfaces for user experience data inspection and preprocessing, together with methods for the extraction of user patterns with machine learning models, such as Kernel Density Estimation (KDE) and hierarchical clustering. Furthermore, the suite is meant to be easily extensible, fostering collaboration within multidisciplinary team. Procezo'a suite allows data scientists to deploy machine learning models and analytics dashboards, which give cultural heritage experts immediate access to data inspection and curation.

2. Related Work

The fast-growing collections of cultural heritage digital assets [TLPW24] brings about the challenge for the development of analytics tools that allow the evaluation of the accessibility and effectiveness of these digital cultural assets [NBCG24, LX24]. While tracking-based analysis of user experience in museums has a long tradition, see [YB09, McM22] for comprehensive reviews, there is a lack of specifically designed cloud-based cultural heritage analytics tools tailored for UX analysis. Frameworks that offer behavioral tracking functionalities usually provide integrated hardware and software solutions, and manage most project aspects, from sensor instrumentation to analytics dashboards: *e.g.*, Noldus Observer XT (https://noldus.com/observer-xt-human), iComfort (https://www.icomfort.it), and inVRsion Shelfzone (https://www.invrsion.com/shelfzone) [CCC*24, PCR*25]. These solutions often require extensive resources compared to cloud-based frameworks that offer mouse-tracking, or eye-tracking for website UX analytics: *e.g.*, iMotions (https://imotions.com), which allows the use of multiple camera feeds or other external hardware, or Gazerecorder (https://gazerecorder.com), which allows the use of common webcams. In practice, many museum and neuroscience teams develop custom frameworks using personalized combinations of software and hardware [CCCO21, CCCP20, JRMJ18, CKK19]. Consequently, most solutions are implemented ad-hoc, and require the collaboration of specifically trained data engineers and data scientists with cultural heritage domain experts.

Previous research [BCBM18, GJS23] shows that team collaboration is beneficial for data inspection and analysis task, especially for cross-disciplinary teams. Research has also found that workspace awareness and objective sharing was enanced by the possibility to share interactive interfaces, which enable multiple analysts to carry out visual analytics tasks on multivariate datasets [LHC*20]. Nevertheless, collaborative IA faces additional challenges, as highlighted in [EBC*21].

A series of collaborative tools has greatly improved collaboration among analysts. Version Control System (VCS) (*e.g.*, Git [CS14] and SVN [Zan10]) enables data scientists to immediately share machine learning (ML) models and datasets. Furthermore, the introduction of Jupyter Notebook [KRKP*16], a web-based interactive computing platform, especially when used together with versioning control systems, has greatly facilitated the collaborative development of new machine learning methods. Meanwhile, cloud-based platforms, such as Google Colab [Bis19], often developed on Jupyter framework, allow asynchronous collaboration, and allow immediate access to virtual machines with powerful hardware (e.g., GPUs). Finaly, third-party open-source machine learning libraries simplify the implementation of complex models and data analysis pipelines, among the most popular, there are Python Scikit-learn [PVG*11], TensorFlow [AAB*15] or PyTorch [PGC*17]. All together, these tools foster and broaden the community of data engineers allowing users to implement complex machine learning models very quickly, giving the data scientist more time to test different alternatives. Nevertheless, synchronous collaboration in shared environments is still unsupported; at each deployment version, the code must be committed to common repositories for other users to upload changes. For this reason, data versioning tools such as Pachyderm are emerging [Kar22]. Unfortunately, these tools still have a steep learning curve, and do not offer simple interfaces that allow the direct inspection of the data pipeline for domain experts who are not necessarily data engineers. Furthermore, simultaneous data inspection is not allowed, and the potential benefits of virtual co-localization in shared representations are restricted.

In this paper, we focus on two machine learning methods Kernel density estimation (KDE) and hierarchical clustering. KDE is a non-parametric method that estimates the probability density function (pdf) of random variables [Par62, Ros56]. KDE is particularly useful when the random variables can not be approximated with simple unimodal distributions such as normal or exponential. Visually, KDE provides a statistically reasonable and immediate way to show where the data points accumulate, by using a color map of the density distribution as a heat map indicating which points are more likely. Hierarchical clustering algorithms are a particular kind of clustering methods, which use the similarity among the data entries to form a hierarchical structure of clusters [Nie16]. Clustering methods are unsupervised algorithms that allow the recognition of common patterns in subgroups without requiring labeled datasets.

3. Procezo Design

As mentioned earlier, Procezo aims to process records, which are derived from user sessions. User session records are obtained from different kinds of experiences, either virtual or real. For this reason, we assume a simple nested organization of the records. The records are organized into groups, and the groups are organized into projects. Consequently, each record would have its *recordId*, together with a *groupId* and a *projectId*. As mentioned, Procezo is part of the Interlumo immersive analytic modular pipeline. If Procezo is used in the pipeline, it receives records from Kapto, which is Interlumos's capture stage. Kapto is a cloud service that allows to produce records from user sessions. Procezo gets the *recordId*, *groupId* and *projectId* variables from Kapto, then it retrieves the records and processes them to extract visitors' interaction patterns. This is called the process stage in Interlumo. Nevertheless, thanks to Interlumo's modular structure, Procezo can also get records generated with other applications as long as the records are in tabular form.

We assume a simple nested organization of the records because both UX design settings, neuroscience experiments, and other similar initiatives reveal a similar organizational pattern. In the case of UX design, a developer's project is usually concerned with a website or exhibit, and in neuroscience, the scientist's project would usually consider an experiment. Furthermore, each website or exhibit is generally divided into different scenes that depend on the designer focus, and the experiments are traditionally divided into separate trials or treatments as specified by the experimental protocol. Procezo is implemented using widely adopted libraries, and is deployed with a Docker container, which allows it to run efficiently in diverse environments and scale according to user demand. Procezo is developed with long-term sustainability as a core design principle. It is part of the Interlumo open-source pipeline, and is integrated into E-RIHS (`https://www.e-rihs.it/`), a major European Heritage Science. This integration ensures ongoing access to services and databases that support its long-term viability. Procezo is fully open-source and available on GitHub. It uses standard, widely interoperable file formats such as csv and json.

To determine Procezo's workflow, we examined the different styles of workflows that emerged in data mining and machine learning teams, both in industry and academia [FG24]. The literature describes several different Analytics methodologies [MPCOF*21]. Among these, CRISP-DM is a fundamental reference point [Cha09], which was developed by a consortium of universities and companies to improve collaboration and cross-fertilization [MPCOF*21]. Even though analytics has shifted to a more descriptive character if compared to the more goal-oriented data mining workflow, CRISP-DM still describes the typical steps of a data project, the tasks involved in each step, and the relationships among these steps. Even if it is recognized that CRISP-DM does not consider all possibilities, it remains an industry-proven methodology, and its main steps are still valid [MPCOF*21]. The main steps in a data project according to CRISP-DM are: problem definition, data acquisition, data preparation, data analysis/modeling, evaluation, and deployment. Procezo's workflow is concerned with the data preparation step, the data analysis/modeling step, and the deployment step.

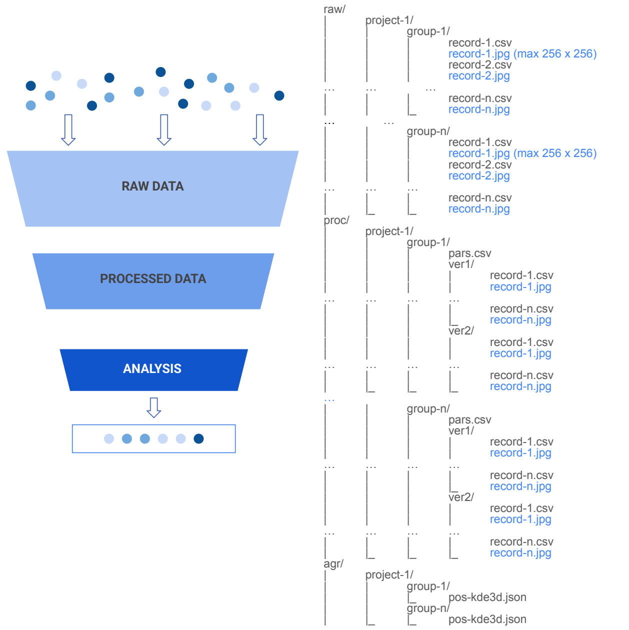

Figure 1: *A) Data flow schema. Raw data is transformed into processed data, and processed data is transformed into aggregate data. B) The file system is organized into folders that reflect the data flow, and it allows for tailoring separate components' writing permissions.*

Procezo is aimed at interdisciplinary teams composed of domain experts, *e.g.*, cultural heritage curators, UX designers, together with ML experts and analytics experts. For this reason, it is implemented with a Web-based interface that allows rapid interaction with the data for local and remote collaborations. The web-based interface is designed to be user-friendly, and it allows all team members to directly inspect the data, and to contribute to the data processing pipeline without the requirement of knowing specific programming languages.

The input to Procezo consists of raw user session records. These records are time series, *i.e.*, are composed of sequences of data points indexed by time. In our case, the records are saved as csv tables with each row indexed with a different timestamp, but json files can equivalently be used. These records could be generated by the Kapto module or from a different application. From here after, we refer to them as raw records or raw data. Once the records are loaded, the Procezo workflow is divided into two main classes of methods, which can be thought of as the pipes that form the pipeline: process methods and aggregate methods. The process methods allow the processing of single records or the filtering/gating of all the records in a group. The process methods carry out the CRISP-DM preparation step. The processing of the single record may consider the selection of a certain region of interest (ROI), or other normalizations/denoising methods. The filtering/gating of the records involves the definition of gating thresholds on the records in a group according to global properties that are computed on the single record (*e.g.*, record duration, variables' average, variables' variability). Finally, the aggregate methods are where the data is aggregated through the use of statistics or machine learning models to produce a mathematical synthetic representation of a subset of records. The aggregate methods carry out the CRISP-DM analysis/modeling step.

To protect the integrity of the data and mitigate unforeseen errors, we divided the records and the different versions that are obtained at each processing/aggregation method in a nested system of directories that reflect the gradual transformation through the workflow (Figure 1). The first directory is named 'raw/', where the raw record data is stored. This folder should be writable from the application producing the records, but not by Procezo's methods. Meanwhile, Procezo should be able to read the records from 'raw/' and save the new version of the record data in the different subfolders in 'proc/', which reflect the different versions produced by the process methods. Finally, Procezo should be able to read the different versions of the data in 'proc/', and write the aggregated data generated by the machine learning methods in the 'agr/' folder.

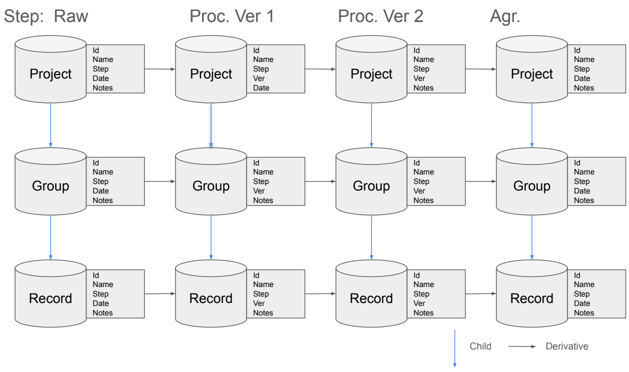

Figure 2: *Schematic representation of the relational object data structure containing the information regarding the nested organization project-group-record (child relation), and the relation between derivative data obtained with Procezo's process and aggregate methods (derivative relation).*

To implement the web application, we implemented a relational data structure composed of three different classes that define the respective objects: a project class, a group class, and a record class. The objects are connected by two different relational links: a child link and a derivative link. Figure 2 shows a schematic representation of this relational data structure. The child-link connects the project to its child groups and the groups to their child records. The derivative-link connects each object to the derivative object obtained through one of Procezo's methods. The project class defines an object that contains specific attributes such as the name, date, and the project stage ('raw', 'proc', and 'agr'). The project class also allows the storage of the relational attributes that connect the project object to the child groups and the derivative processed project. Similarly, the group class defines an object that contains, among other things, the name, state, and version. The group class also contains the relation attributes that connect the group to its parent project and its child records, together with the derivative groups that are the result of a processing method. Finally, the record class defines the object that contains the record name, state, and version. It also contains the relational connection to the parent group and the derived records. This data structure allows the implementation of a RESTful web APIs [LCZL16, EACM22], that enables the connectivity between Procezo's, its apps, and the other steps in Interlumo's modular pipeline, though the typical requests: GET, POST, PUT, PATCH, and DELETE.

Procezo interfaces are developed as web applications. The web applications consist of a backend developed on Python using modules such as Numpy [HMvdW*20], Pandas [pdt20, WM10], Scipy [VGO*20], Scikit-learn [PVG*11], Pytorch [PGC*17]. We used FastAPI as middleware to have a RESTful web API with the relevant OpenAPI documentation. Finally, for the frontend, we used HTML with JavaScript integration and developed the dashboards with Plotly and Dash [Inc15]. This stack design allows us to implement cloud-based processing tools that can be easily connected to other services in a cloud environment.

4. Data Process and Aggregate Methods

Procezo's workflow is divided into two main method classes: process methods and aggregate methods. The process method allows the processing of the single raw records, and the filtering/gating of the records in a single group. These methods can be implemented through Python scripts, Jupyter Notebooks, or interactive apps; in any case, when executed, each method produces a new folder named with a *versionId* in the 'proc/' folder, and stores in this folder the data derived from that method. The aggregate methods are usually implemented through a Jupyter Notebook. They read a specific version of the record data in 'proc/' and generate an aggregate version (with a specific *versionId*) of the data in the 'agr/' folder.

In the case study, two process methods are used: the record edit method for ROI selection and filtering, and the group edit method for filtering/gating the records. The record edit method is used to select the records that are used for the analysis and determine the ROI on which the analysis is performed. It is important to remember that the protocol for dropping out records and the selection of the ROI should be defined *a priori* on factors that do not bias the analysis outcome. Figure 3 shows the record edit method interface. The record captures an experience in an immersive virtual environment, and reports the visitor's HMD position $P = (p_x, p_y, p_z)$ and view/head direction $V = (v_x, v_y, v_z)$ samples at regular time steps.

The group edit method for filtering/gating the records in a group allows users to plot certain measurable scalar quantities, such as session time and total variance, and define upper and lower thresholds to gate the records on these quantities. In this case, the session time would be the difference between the final and initial time of the record ROI, and the total variance would be the sum of the variances from the variables determined by the record columns. This process is not necessarily required, but allows the users to drop out outlier records that present gross deviations from the bulk dis-

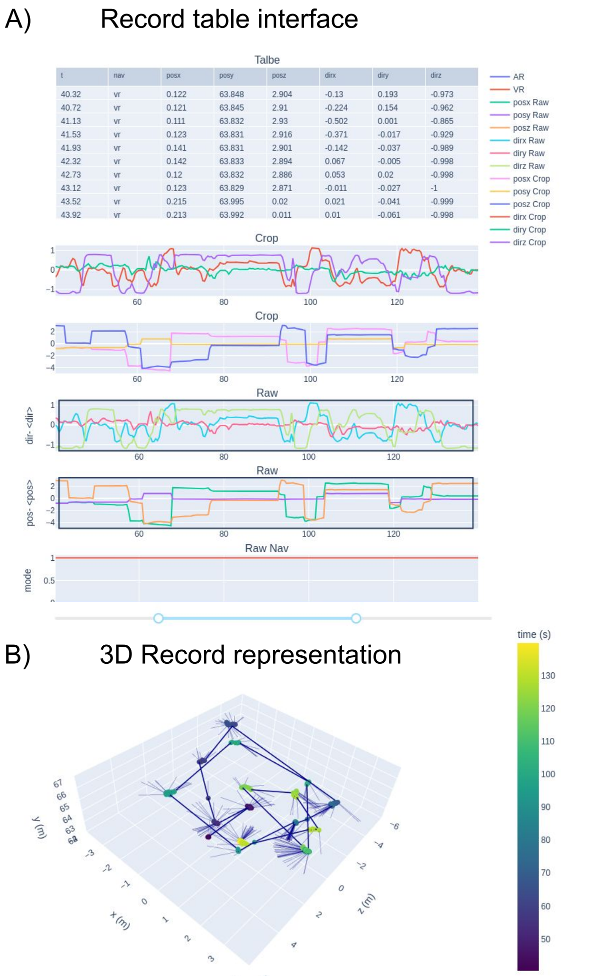

Figure 3: *A) The record edit method interface uses a range slider, and a checkbox to select the records that are kept for analysis. On top, it presents a table, and below, 2D plots showing the records columns on the y-axis and time on the x-axis. In this case, we show a record with the visitor's HMD position P (3D vector), and view/head direction D (3D normalized vector). The first two plots show the ROI cropped with the slider, the second two plots show the full record with rectangles representing the ROI, and the last plot shows the navigation mode (*i.e., *VR or AR). B) 3D trajectory of the visitor's HMD positions, with color representing time, and arrows representing the HMD direction for each recorded position.*

tribution. It is well-known that certain statistical aggregates are biased by gross outliers. Figure 4 shows an example of group-records gating. In this case, records were gated according to session time length and variation. The reason is that too short session times may reflect accidents where the session was started and immediately finished before the user could start her/his experience. Similarly, if the variance is very low, it may be because the user did not move, as she or he may have experienced a technical problem.

The main aggregate methods discussed in this paper and used in the case study are Kernel Density Estimation (KDE) and Hierarchical clustering [FG24]. Nevertheless, different machine learning methods can be implemented into new aggregate methods using Jupyter Notebooks or Python scripts.

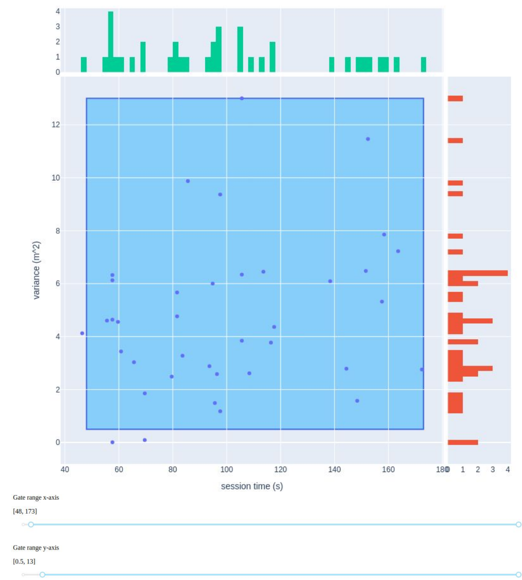

Figure 4: *The record edit method interface uses two range sliders to define the upper and lower thresholds for the filtering/gating strategy. In this case, we considered the option to gate records for session time length and total variance.*

For the KDE aggregation method, we used the Gaussian KDE algorithm (Scipy implementation [VGO*20]). The Gaussian KDE algorithm estimates the probability density function (pdf). Given a sample of points $(X_0, X_1, \ldots, X_n)$, the pdf f estimation is obtained with the expression

$$\hat{f}_{h(X)} = \frac{1}{nh} \sum_{i=0}^{n} K\left(\frac{X - X_i}{h}\right) \tag{1}$$

where $K(X)$ is a normal kernel, and h is the bandwidth.

For the hierarchical clustering implementation, we selected agglomerative clustering [BJGJ01, LW67]. The agglomerative clustering algorithm works through an iterative loop, constructing a binary tree, bottom-up, from the leaves to the root. This binary tree, or dendrogram, represents the clusters' hierarchical structure, composed of smaller clusters joining together in larger clusters as branches merge. The binary tree is constructed in the following way. At the beginning, the initial input elements start in singleton clusters, which form the leaves of the binary tree. Then, at each step, the clusters corresponding to different branches are merged, and a level in the hierarchy is added. The couple of clusters to be merged are iteratively selected with a greedy algorithm on the basis of a linkage clustering criterion. The linkage clustering criterion is determined by a distance measure defined on the current cluster. Thus, after merging the clusters, we update the cluster list, and continue in the iterative loop. At the end, the last two remaining clusters are merged into the root node. From the hierarchical clustering process, we obtain the hierarchical binary tree, or dendrogram.

A common distance measure among entries is the Cartesian distance. Otherwise, it can be derived from Pearson's correlation. Given a distance measure, we can define a clustering linkage criterion, or within-cluster distance (WCD), which may be, for example, the *complete* criterion

$$d(U,V) = \max_{u \in U, v \in V} d(u,v) \tag{2}$$

where U and V are two clusters and u and v are two data entries from each cluster, respectively [Nie16]. In our case study, we used the Python Scikit-learn implementation of agglomerative clustering [PVG*11].

4.1. Case Study and Applications

To test Procezo's implementation, we developed a WebXR application and used Kapto to generate the user experience records. The WebXR application allows access to different virtual environments hosted in ATON, which is an open-source framework designed to present and enable interaction with 3D models through a standard web browser across multiple devices [FFD*21]. For our case study, we considered a 3D survey of the Etruscan Tomb of Reliefs located in the Banditaccia Necropolis, Cerveteri (Italy). The web application tracked spatial data, which was obtained from anonymous general public users who participated in three different exhibits in Italy and a dissemination event at our lab. The three different exhibits were ArcheoVirtual 2023, TourismA 2024, and TourismA 2025. ArcheoVirtual 2023 was held in Paestum (3 days), TourismA 2024 and 2025 were held in Florence (2 days). These events allowed us to test different aspects of Procezo and the Interlumo pipeline. First, in the case of the exhibits, the two separate servers running the Interlumo pipeline and ATON were located in the CNR research area in Rome, and users performed their sessions using remote devices. Thus, data transmission requires an internet connection. Second, the exhibit users were visitors attending events focused on heritage and AI. These conditions allowed us to test the opportunity to study and investigate how people respond to and interact with both cultural heritage and generative virtual content, using HMDs (immersive VR).

4.2. Equipment and Setup

The setups used in the events replicate the setup introduced in [FG24]. It is composed of two server nodes with public access. The servers were accessible over the internet through REST API and web interfaces.

1. Interlumo immersive analytic modular pipeline: this dedicated server hosts Kapto and Procezo, under the H2IOSC project.
2. ATON server: this dedicated server hosts the main instance of the ATON framework [FFD*21], providing web application and 3D content. In this case, the WebXR application, the virtual cultural heritage and generative 3D content, and the "Merkhet" WebXR for visual inspection.

Both servers run Linux OS (Ubuntu Server 20.04.6 LTS) with Node.js and PM2 setups for the cluster deployment of microservices, and are located in Rome. The exhibit equipment for the onsite physical installations was composed of the following:

- A single workstation and one HMD (HP Reverb G2 Headset) were used to experience immersive VR scenes for the ArcheoVirtual event;
- A standalone HMD (Meta Quest PRO) was used to experience immersive VR scenes for the exhibits and the in-lab dissemination event.

An internet connection was required for both setups to connect with ATON and the Interlumo immersive analytics modular pipeline. This case study enabled us to benchmark Procezo, which ran without particular slowdowns, interfaces loaded in a few seconds, and the augmentation modules completed their workload in the order of minutes.

4.3. Web app

We developed a WebXR application [FG24] that allows visitors to access and explore the virtual environments online using a standard browser. To track a well-defined set of attributes in 3D spaces, we instrumented the WebXR app with Kapto for acquiring the record data [FG24]. Then we used a fixed sample time (interval) of $0.2s$. Such a sample time demonstrated a good trade-off between the file size of the session records and the trajectory resolution. Attributes that were tracked included the user's location P (3D vector), orientation O (Quaternion), view/head direction V (3D normalized vector), navigation mode (string), and field-of-view (float). Eye tracking was not implemented in all the HMDs, thus, we implemented [UE17]'s method, which estimates an approximated field-of-view with ray-casting given V. Users were informed before the experience by onsite personnel about the ongoing anonymous tracking of such spatial attributes. During the exhibits, the WebXR application generated more than 500 raw variable duration records (immersive sessions) with the Interlumo immersive analytic modular pipeline. In all cases, the onsite experience was supervised by dedicated personnel: the initial explanation of the experience and the start/end of session recording.

4.4. Cerveteri Etruscan Tomb

For the Cerveteri Etruscan Tomb virtual environment, we used a previously published 3D scene. This virtual environment presented semantic annotations validated by professionals. The annotation text was first automatically translated (Italian-to-English), and then it was converted into narrating voices with AI text-to-speech services. Aural content labels were used to augment the virtual environment with semantic annotations. Image 5A shows semantic labels in overlay.

Exhibit and in-lab visitors were free to move around and inspect the 3D model using an HMD. For virtual environment navigation, the visitors were allowed to use natural movement and teleport. Exhibit visitors tended to teleport frequently to overcome the limited free movement space given by the kiosk setup (Figure 5B). Contrarily, in-lab visitors could move more-freely because they were intentionally situated in a large room that did not inhibit free movement (Figure 5C). All exhibit visitors produced zig-zag trajectory

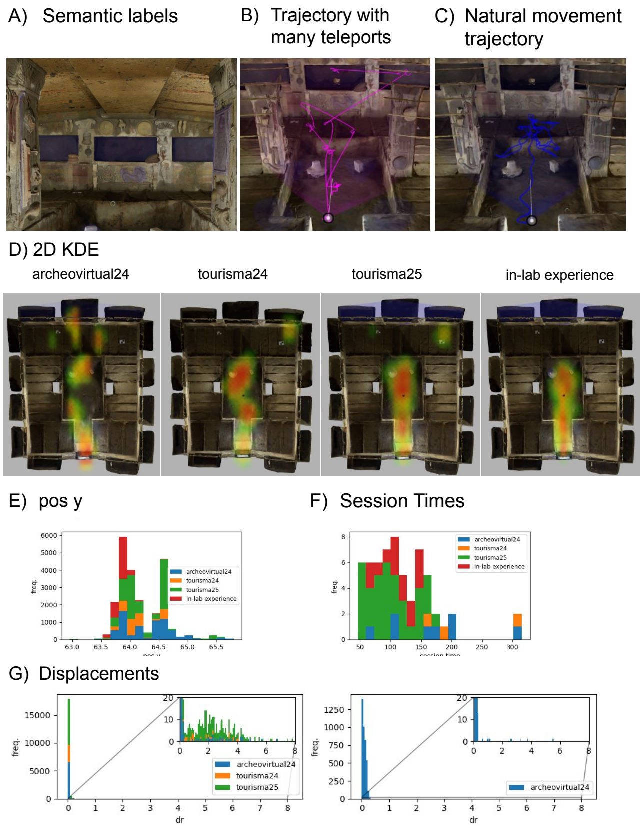

Figure 5: *A) Semantic labels. B) Trajectory with several teleports. C) Natural movement Trajectory. D) position KDE (P_x, P_y, P_z) computed on the three exhibit events (ArcheoVirtual23, TourismA24, TourismA25) and in the in-lab experience. E) HMD height position P_y histogram, in y-up coordinates. F) Session time length histogram. G) Displacement distance between consecutive capture events.*

patterns, clearly discernible at a visual inspection, typical of teleport navigation. Meanwhile, all the in-lab visitors' trajectories presented strikingly different wavy patterns.

Panel 5D shows the KDE computed on the visitors' positions. Given that, in this case, we considered each recorded visitor's trajectory u to be a sequence of positions, $\mathbf{P}_u = (P_1, P_2, \ldots, P_{n_u})$, and $\mathbf{P}$ the set containing all the sampled positions from all the visitors together. We computed the positions' KDEs, $\hat{f}_\sigma(P|\mathbf{P})$, using Equation (1) and Scott's rule [Sco15] for bandwidth selection. Panel 5D shows KDEs from the three exhibits and the in-lab experience. From a visual inspection, we find that the KDEs from the exhibits (*i.e.*, ArcheoVirtual 2023, TourismA 2024, and TourismA 2025) are more consistent among themselves compared to the in-lab dissemination event KDE (in-lab experience). The in-lab 2D KDE shows how the free-moving visitors mostly moved on the tomb floor and did not explore the parts of the tomb that required stepping on the raised tumb "carved benches", while the exhibit visitors KDE showed that exhibit visitors tended to position themselves on these raised structures. An interpretation is that a behavior they would have had in real life was carried over in the in-lab setting, where the visitors were free to move more naturally, but was not carried over in the situation in which the use of teleport seemed to "inhibit" a natural exploration. Alternatively, since the teleport uses an illuminated disk as an interface to show reachable positions, this may encourage the users to explore more unusual portions of the scene. Panel 5F shows that the P_y position was mostly constant. Panel 5G shows that session duration was consistent between exhibit and in-lab visits, with some outliers in the exhibit sessions. Panel in 5G shows that given the sampling rate 0.2 s, most of the displacement in position between successive positions is very small. As opposed to teleport movements that are clearly marked by large instantaneous displacements. Indeed, we observe that in the in-lab experience case, visitors' teleport use is negligible. See the zoom insert in the plots in 5G.

To study how the users' trajectories $\mathbf{P}_u$ clustered in different groups, we implemented the agglomerative clustering model discussed in section 4. We computed the position's KDE $\hat{f}_\sigma(P|\mathbf{P}_u)$ for each user's trajectory, u. To do this, we used Equation 1, and Scott's rule [Sco15] for bandwidth selection. Figure 6 shows the hierarchical clustering output we obtained. To obtain this output, we selected the Pearson correlation to compute the cost function for the agglomerative clustering algorithm. The Pearson correlation between the view trajectory KDE couples is defined as

$$\begin{aligned}\rho(u,v) =& \rho(\hat{f}_\sigma(P|\mathbf{P}_u), \hat{f}_\sigma(P|\mathbf{P}_v)) = \\ =& \frac{\sum_{x,z}(\hat{f}_\sigma(P|\mathbf{P}_u) - M_u)(\hat{f}_\sigma(P|\mathbf{P}_v) - M_v)}{\sqrt{\sum_{x,z}(\hat{f}_\sigma(P|\mathbf{P}_u) - M_u)^2 \sum_{x,z}(\hat{f}_\sigma(P|\mathbf{P}_v) - M_v)^2}}\end{aligned} \tag{3}$$

where $M_a = \sum_{x,z} \hat{f}_\sigma(P|\mathbf{P}_a)$. Panel 6A shows the trajectories' pairs with the highest and lowest Pearson correlations. Given the Pearson correlation definition, we obtained the Pearson correlation matrix shown in Panel 6B on the right. Thus, we computed the agglomerative clustering algorithm, with $d(u,v) = 1 - \rho(u,v)$ and the complete linkage criterion, and obtained the dendrogram in Panel 6B on the left. We stopped the iterative agglomeration process when we reached three clusters (see Panel 6B). Figure 6C shows, with Merkhet [FFD*21], where the visitors in the three clusters C0, C1, and C2, respectively, stopped the most, elucidating how the three clusters were differentiated by their exploration patterns. Visitors in clusters C0 and C1 spent most of the time exploring the central area delimited by the floor of the Tomb. Meanwhile, visitors in C2 spent more time in the raised sections of the Tomb and got closer to the niches. Trajectories in C0 spent more time on the raised sections of the Tomb compared to visitors in C1, and spent more time close to the right column. In comparison, the visitor in C2 spent more time close to the left column. Visitors in C2 and C0 explored the Tomb more systematically.

5. Discussion and Future Work

Procezo's is meant to foster collaboration among teams of experts with different backgrounds. For this reason, Procezo's methods can be implemented either as intuitive web interfaces, as Python scripts, or as Jupyter Notebooks. Allowing teams composed of data scientists and domain experts to collaborate by having data scientists

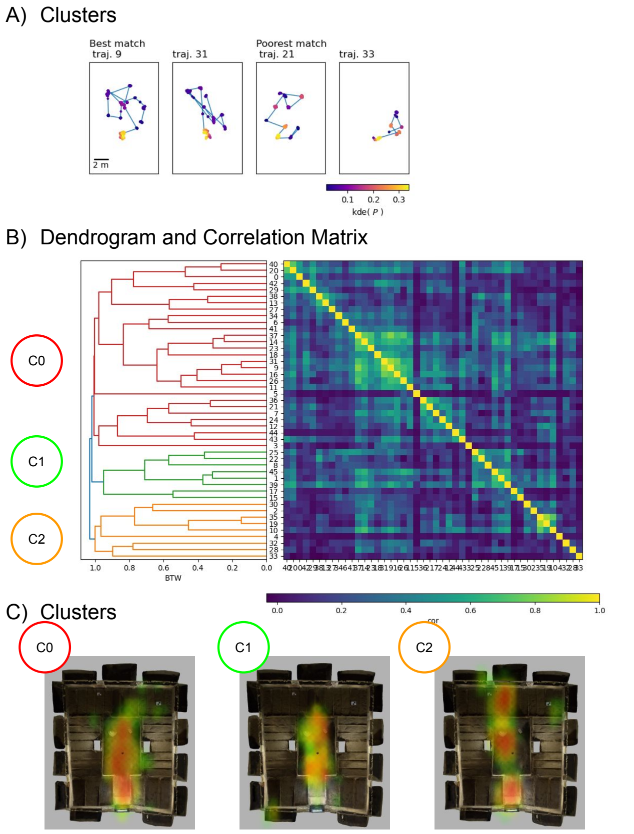

Figure 6: *Agglomerative clustering of Cerveteri's Tumb users' position trajectories. A) Shows the two visitors' trajectories with the highest correlation (best match), and the two trajectories with the lowest correlation (poorest match). B) Agglomerative clustering dendrogram, obtained using $d(u,v) = 1 - \rho(u,v)$ as a metric (where $corr(x,y)$ is the Pearson correlation between the KDE of two trajectories), and the complete linking condition. The x-axis is the WCD, the linking condition value at which the clusters merge. We obtained three clusters. To the right of the dendrogram, we present the Pearson correlation matrix between all view trajectory KDE couples. C) Shows the three trajectories' clusters, and the relative aggregated cluster KDE.*

prepare apps that allow domain experts to interact directly with the data. Furthermore, the nested data system and the relative data structure are conceived to protect the data, and to track the data version evolution obtained with process methods and aggregate methods. This structure allows the team to test different pipelines and different methods parameterizations simultaneously. Furthermore, this structure allows easy changes in data analysis strategies that can be especially useful in exploratory analysis. At the end of the project, we will have domain expert evaluations through the H2IOSC transnational application (TNA) calls, which we will use to validate Procezo's usability and derive quantitative performance metrics.

Procezo's main challenge is to demonstrate how analytics and machine learning can guide cultural heritage curators to improve outreach and engagement of virtual and real exhibits visitors. Here, we showed how Procezo was able to recognize interaction patterns in AI-generated panoramic landscapes and cultural heritage applications. Showing the ability to distinguish different patterns in different visitor control groups.

In the future, we intend to test how well Procezo can be used to test different experimental conditions, or to study the effects of small design changes in cultural heritage applications. For example, we could test the efficacy of different audio content in maintaining visitor engagement, or we could test the effects of different museum layouts. These experiments would allow cultural heritage creators to apply data-driven user experience design. Furthermore, we aim to extend Procezo, for real-world tracking, but this would require a new module able to segment data that is not already in tabular form. This would be of great interest for the application of Procezo to real-life exhibits. Finally, we will also test the opportunity to use different kinds of sensors, such as EEG neuroscience sensors, and smart environmental control sensors. EEG sensors will allow us to apply Procezo in experiments that combine cultural heritage and neuroscience. Environmental sensors could be interesting to monitor cultural heritage sites and to predict the effect of environmental conditions on degradation.

6. Conclusion

We developed a modular application that allows the collaboration of teams composed of data scientists and cultural heritage experts. Procezo allows data scientists to develop customizable pipelines and cloud-based web interfaces using Python and Jupyter. Consequently, with these interactive web interfaces, cultural heritage experts can interact directly with the data and formulate different testable hypotheses that allow data-driven design. The implementation of data-driven design in the context of cultural heritage assets will enable the testing and development of new outreach and dissemination paradigms, grounded in the observation and analysis of visitor interaction patterns. This may have an important impact on how the general public relates to cultural heritage, fostering a positive cycle of enrichment of the community and recognition of the cultural heritage. It may also allow the design of cultural heritage assets tailored for visitors with varying needs and perspectives. On the other side, data-driven UX may lead to excessive gamification as observed in social media, and this may shallow the understanding of the cultural complexity and connections in cultural heritage assets that form the intricate texture of human heritage. In conclusion, Procezo and more generally the Interlumo immersive analytic modular pipeline may have an important impact on the curation of virtual and real-life cultural heritage experiences. In future research, we hope to be able to explore these deep-rooted intricacies.

Data Availability

Procezo' suite can be found on GitHub's repository https://github.com/ggosti/procezo-dush.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial intelligence
BCI	Brain–computer interface
HMD	Head-mounted display
H2IOSC	Humanities and Cultural Heritage Italian Open Science
IA	Immersive analytics
KDE	Kernel density estimation
ML	Machine learning
RI	Research Infrastructure
UI	User Interface
UX	User Experience
VR	Virtual Reality
WCD	Within-cluster distance

Acknowledgments

This research is funded by the H2IOSC Project—Humanities and Cultural Heritage Italian Open Science Cloud (https://www.h2iosc.cnr.it/), funded by the European Union NextGenerationEU—National Recovery and Resilience Plan (NRRP)—Mission 4 "Education and Research" Component 2 "From research to business" Investment 3.1, "Fund for the realization of an integrated system of research and innovation infrastructures", Action 3.1.1 "Creation of new research infrastructures strengthening of existing ones and their networking for Scientific Excellence under Horizon Europe"—Project code IR0000029-CUP-B63C22000730005. Implementing Entity CNR.

References

[AAB*15] Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G. S., Davis A., Dean J., Devin M., Ghemawat S., Goodfellow I., Harp A., Irving G., Isard M., Jia Y., Jozefowicz R., Kaiser L., Kudlur M., Levenberg J., Mané D., Monga R., Moore S., Murray D., Olah C., Schuster M., Shlens J., Steiner B., Sutskever I., Talwar K., Tucker P., Vanhoucke V., Vasudevan V., Viégas F., Vinyals O., Warden P., Wattenberg M., Wicke M., Yu Y., Zheng X.: TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. URL: https://www.tensorflow.org/. 2

[BCBM18] Billinghurst M., Cordeil M., Bezerianos A., Margolis T.: Collaborative immersive analytics. *Immersive Analytics* (2018), 221–257. 2

[Bis19] Bisong E.: *Google Colaboratory*. 2019. doi:10.1007/978-1-4842-4470-8_7. 2

[BJGJ01] Bar-Joseph Z., Gifford D. K., Jaakkola T. S.: Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics 17* (2001). doi:10.1093/bioinformatics/17.suppl_1.S22. 5

[BMD*24] Buragohain D., Meng Y., Deng C., Li Q., Chaudhary S.: Digitalizing cultural heritage through metaverse applications: challenges, opportunities, and strategies. *Heritage Science 12* (8 2024), 295. doi:10.1186/s40494-024-01403-1. 1

[CCC*24] Ceccarelli S., Cesta A., Cortellessa G., Benedictis R. D., Fracasso F., Leopardi L., Ligios L., Lombardi E., Malatesta S. G., Oddi A., Pagano A., Palombini A., Romagna G., Sanzari M., Schaerf M.: Evaluating visitors' experience in museum: Comparing artificial intelligence and multi-partitioned analysis. *Digital Applications in Archaeology and Cultural Heritage 33* (6 2024), e00340. doi:10.1016/J.DAACH.2024.E00340. 2

[CCCO21] Centorrino P., Corbetta A., Cristiani E., Onofri E.: Managing crowded museums: Visitors flow measurement, analysis, modeling, and optimization. *Journal of Computational Science 53* (7 2021), 101357. doi:10.1016/J.JOCS.2021.101357. 2

[CCCP20] Casolla G., Cuomo S., Cola V. S. D., Piccialli F.: Exploring unsupervised learning techniques for the internet of things. *IEEE Transactions on Industrial Informatics 16* (4 2020), 2621–2628. doi:10.1109/TII.2019.2941142. 2

[Cha09] Chapman P.: *CRISP-DM 1.0: Step-by-step data mining guide*. Tech. rep., The CRISP-DMconsortium, 2009. 3

[CKK19] Clay V., König P., König S.: Eye tracking in virtual reality. *Journal of Eye Movement Research 2019, Vol. 12, Pages 1-18 12* (4 2019), 1–18. doi:10.16910/JEMR.12.1.3. 2

[CS14] Chacon S., Straub B.: *Pro git*. Apress, 2014. 2

[DHB*22] Dwivedi Y. K., Hughes L., Baabdullah A. M., Ribeiro-Navarrete S., Giannakis M., Al-Debei M. M., Dennehy D., Metri B., Buhalis D., Cheung C. M., Conboy K., Doyle R., Dubey R., Dutot V., Felix R., Goyal D. P., Gustafsson A., Hinsch C., Jebabli I., Janssen M., Kim Y. G., Kim J., Koos S., Kreps D., Kshetri N., Kumar V., Ooi K. B., Papagiannidis S., Pappas I. O., Polyviou A., Park S. M., Pandey N., Queiroz M. M., Raman R., Rauschnabel P. A., Shirish A., Sigala M., Spanaki K., Tan G. W.-H., Tiwari M. K., Viglia G., Wamba S. F.: Metaverse beyond the hype: Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. *International Journal of Information Management 66* (10 2022), 102542. doi:10.1016/J.IJINFOMGT.2022.102542. 1

[EACM22] Ehsan A., Abuhaliqa M. A. M., Catal C., Mishra D.: Restful api testing methodologies: Rationale, challenges, and solution directions. *Applied Sciences 12*, 9 (2022), 4369. 4

[EBC*21] Ens B., Bach B., Cordeil M., Engelke U., Serrano M., Willett W., Prouzeau A., Anthes C., Büschel W., Dunne C., et al.: Grand challenges in immersive analytics. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (2021), pp. 1–17. 2

[FFD*21] Fanini B., Ferdani D., Demetrescu E., Berto S., d'Annibale E.: Aton: An open-source framework for creating immersive, collaborative and liquid web-apps for cultural heritage. *Applied Sciences 11*, 22 (2021), 11062. 1, 6, 7

[FG24] Fanini B., Gosti G.: A new generation of collaborative immersive analytics on the web: Open-source services to capture, process and inspect users' sessions in 3d environments. *Future Internet 16*, 5 (2024), 147. 1, 3, 5, 6

[GJS23] Garrido D., Jacob J., Silva D. C.: Performance impact of immersion and collaboration in visual data analysis. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (2023), IEEE, pp. 780–789. 2

[HMvdW*20] Harris C. R., Millman K. J., van der Walt S. J., Gommers R., Virtanen P., Cournapeau D., Wieser E., Taylor J., Berg S., Smith N. J., Kern R., Picus M., Hoyer S., van Kerkwijk M. H., Brett M., Haldane A., del Río J. F., Wiebe M., Peterson P., Gérard-Marchant P., Sheppard K., Reddy T., Weckesser W., Abbasi H., Gohlke C., Oliphant T. E.: Array programming with NumPy. *Nature 585*, 7825 (Sept. 2020), 357–362. doi:10.1038/s41586-020-2649-2. 4

[Inc15] Inc. P. T.: Collaborative data science, 2015. URL: https://plot.ly. 4

[JRMJ18] John B., Raiturkar P., Meur O. L., Jain E.: A benchmark of four methods for generating 360° saliency maps from eye tracking data. *Proceedings - 2018 IEEE International Conference on Artificial Intelligence and Virtual Reality, AIVR 2018* (7 2018), 136–139. doi:10.1109/AIVR.2018.00028. 2

[KAA*21] Komarudin O., Arrumaisha H., Azurat A., De R., Scheidt F., Vilain P., Dantas M. A. R., Pimenov D., Solovyov A., Askarbekuly N., Mazzara M.: Data-driven approaches to user interface design: A case study. *Journal of Physics: Conference Series 2134* (12 2021), 012020. doi:10.1088/1742-6596/2134/1/012020. 1

[Kar22] KARSLIOGLU S.: *Reproducible Data Science with Pachyderm: Learn how to build version-controlled, end-to-end data pipelines using Pachyderm 2.0*. Packt Publishing Ltd, 2022. 2

[KC20] KONSTANTAKIS M., CARIDAKIS G.: Adding culture to ux: Ux research methodologies and applications in cultural heritage. *Journal on Computing and Cultural Heritage 13* (2 2020). doi:10.1145/3354002. 2

[KK23] KOUSTRIAVA E., KOUTSMANI M.: Spatial and information accessibility of museums and places of historical interest: A comparison between london and thessaloniki. *Sustainability 2023, Vol. 15, Page 16611 15* (12 2023), 16611. doi:10.3390/SU152416611. 2

[KRKP*16] KLUYVER T., RAGAN-KELLEY B., PÉREZ F., GRANGER B., BUSSONNIER M., FREDERIC J., KELLEY K., HAMRICK J., GROUT J., CORLAY S., IVANOV P., AVILA D., ABDALLA S., WILLING C.: Jupyter notebooks – a publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (2016), Loizides F., Schmidt B., (Eds.), IOS Press, pp. 87 – 90. 2

[LCZL16] LI L., CHOU W., ZHOU W., LUO M.: Design patterns and extensibility of rest api for networking applications. *IEEE Transactions on Network and Service Management 13*, 1 (2016), 154–167. 4

[LHC*20] LEE B., HU X., CORDEIL M., PROUZEAU A., JENNY B., DWYER T.: Shared surfaces and spaces: Collaborative data visualisation in a co-located immersive environment. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2020), 1171–1181. 2

[LW67] LANCE G. N., WILLIAMS W. T.: A general theory of classificatory sorting strategies: 1. hierarchical systems. *The Computer Journal 9* (3 1967), 373–380. doi:10.1093/comjnl/9.4.373. 5

[LX24] LIAN Y., XIE J.: The evolution of digital cultural heritage research: Identifying key trends, hotspots, and challenges through bibliometric analysis. *Sustainability 2024, Vol. 16, Page 7125 16* (8 2024), 7125. doi:10.3390/SU16167125. 2

[McM22] MCMURTRIE R. J.: Observing, recording, visualising and interpreting visitors' movement patterns in art museums: A mixed method approach. *Multimodality & Society 2* (6 2022), 93–113. doi:10.1177/26349795221100132. 2

[MPCOF*21] MARTINEZ-PLUMED F., CONTRERAS-OCHANDO L., FERRI C., HERNANDEZ-ORALLO J., KULL M., LACHICHE N., RAMIREZ-QUINTANA M. J., FLACH P.: Crisp-dm twenty years later: From data mining processes to data science trajectories. *IEEE Transactions on Knowledge and Data Engineering 33* (8 2021), 3048–3061. doi:10.1109/TKDE.2019.2962680. 3

[NBCG24] NAPPI M. L., BUONO M., CHIVĂRAN C., GIUSTO R. M.: Models and tools for the digital organisation of knowledge: accessible and adaptive narratives for cultural heritage. *Heritage Science 12* (12 2024), 1–22. doi:10.1186/S40494-024-01219-Z;KWRD=MATERIALS+SCIENCE. 2

[Nie16] NIELSEN F.: Introduction to hpc with mpi for data science. *Springer* (2016). 2, 6

[Par62] PARZEN E.: On estimation of a probability density function and mode. *The Annals of Mathematical Statistics 33* (9 1962), 1065–1076. doi:10.1214/aoms/1177704472. 2

[PCR*25] PALOMBINI A., CECCARELLI S., ROMAGNA G., SANZARI M., SCHAERF M., ODDI A.: Artificial intelligence algorithms for the characterisation of visitors trajectories in a cultural context. 117–135. doi:10.1007/978-981-96-4550-3_8/FIGURES/13. 2

[pdt20] PANDAS DEVELOPMENT TEAM T.: pandas-dev/pandas: Pandas, Feb. 2020. doi:10.5281/zenodo.3509134. 4

[PGC*17] PASZKE A., GROSS S., CHINTALA S., CHANAN G., YANG E., DEVITO Z., LIN Z., DESMAISON A., ANTIGA L., LERER A.: Automatic differentiation in pytorch. In *NIPS-W* (2017). 2, 4

[PVG*11] PEDREGOSA F., VAROQUAUX G., GRAMFORT A., MICHEL V., THIRION B., GRISEL O., BLONDEL M., PRETTENHOFER P., WEISS R., DUBOURG V., VANDERPLAS J., PASSOS A., COURNAPEAU D., BRUCHER M., PERROT M., DUCHESNAY E.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830. 2, 4, 6

[Ros56] ROSENBLATT M.: Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics 27* (9 1956), 832–837. doi:10.1214/aoms/1177728190. 2

[Sco15] SCOTT D. W.: *Multivariate Density Estimation*. Wiley, 3 2015. doi:10.1002/9781118575574. 7

[SPK20] SONG H., PARK C., KIM M.: Tourism destination management strategy for young children: Willingness to pay for child-friendly tourism facilities and services at a heritage site. *International Journal of Environmental Research and Public Health 17* (10 2020), 7100. doi:10.3390/IJERPH17197100. 2

[TLPW24] TANG Y., LIU L., PAN T., WU Z.: A bibliometric analysis of cultural heritage visualisation based on web of science from 1998 to 2023: a literature overview. *Humanities and Social Sciences Communications 11* (12 2024), 1–11. doi:10.1057/S41599-024-03567-4. 2

[UE17] UPENIK E., EBRAHIMI T.: A simple method to obtain visual attention data in head mounted virtual reality. In *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)* (2017), IEEE, pp. 73–78. 6

[VGO*20] VIRTANEN P., GOMMERS R., OLIPHANT T. E., HABERLAND M., REDDY T., COURNAPEAU D., BUROVSKI E., PETERSON P., WECKESSER W., BRIGHT J., VAN DER WALT S. J., BRETT M., WILSON J., MILLMAN K. J., MAYOROV N., NELSON A. R. J., JONES E., KERN R., LARSON E., CAREY C. J., POLAT İ., FENG Y., MOORE E. W., VANDERPLAS J., LAXALDE D., PERKTOLD J., CIMRMAN R., HENRIKSEN I., QUINTERO E. A., HARRIS C. R., ARCHIBALD A. M., RIBEIRO A. H., PEDREGOSA F., VAN MULBREGT P., SCIPY 1.0 CONTRIBUTORS: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods 17* (2020), 261–272. doi:10.1038/s41592-019-0686-2. 4, 5

[WM10] WES MCKINNEY: Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference* (2010), Stéfan van der Walt, Jarrod Millman, (Eds.), pp. 56 – 61. doi:10.25080/Majora-92bf1922-00a. 4

[YB09] YALOWITZ S. S., BRONNENKANT K.: Timing and tracking: Unlocking visitor behavior. *Visitor Studies 12* (1 2009), 47–64. doi:10.1080/10645570902769134;WGROUP:STRING:PUBLICATION. 2

[Zan10] ZANDSTRA M.: *Version Control with Subversion*. 2010. doi:10.1007/978-1-4302-2926-1_17. 2