Computers & Graphics (2022)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Audio-visual Annotation Graphs for Guiding Lens-based Scene Exploration

Moonisa Ahsan[a,*], Fabio Marton[a,*], Ruggero Pintus[a,*], Enrico Gobbetti[a,*]

[a]*CRS4, Via Ampère 2, Cagliari (CA), I-09134, Italy*

ARTICLE INFO

Article history:
Received 11 february 2022
Accepted 2 May 2022
Available online 11 May 2022

Keywords:
interactive visualization lenses, annotations, user interfaces, interactive exploration, guidance, guided tour

ABSTRACT

We introduce a novel approach for guiding users in the exploration of annotated 2D models using interactive visualization lenses. Information on the interesting areas of the model is encoded in an annotation graph generated at authoring time. Each graph node contains an annotation, in the form of a visual and audio markup of the area of interest, as well as the optimal lens parameters that should be used to explore the annotated area and a scalar representing the annotation importance. Directed graph edges are used, instead, to represent preferred ordering relations in the presentation of annotations, by having each node point to the set of nodes that should be seen before presenting its associated annotation. A scalar associated to each edge determines the strength of this constraint. At run-time, users explore the scene with the lens, and the graph is exploited to select the annotations that have to be presented at a given time. The selection is based on the current view and lens parameters, the graph content and structure, and the navigation history. The best annotation under the lens is presented by playing the associated audio clip and showing the visual markup in overlay. When the user releases control, requests guidance, opts for automatic touring, or when no available annotations are under the lens, the system guides the user towards the next best annotation using glyphs, and potentially moves the lens towards it if the user remains inactive. This approach supports the seamless blending of an automatic tour of the data with interactive lens-based exploration. The approach is tested and discussed in the context of the exploration of multi-layer relightable models.

1. Introduction

The virtual inspection of digital scenes, including the simulation results or digital replicas of physical objects, is of fundamental importance for many use cases in disparate application fields. A typical example occurs in Cultural Tourism and Cultural Heritage (CH) domains, where the inspection of models is recognized as a precious means to support the three main stages related to the enjoyment of the artworks, i.e., the pre-visit (documentation and planning), visit (immersion and enhancement) and post-visit (emotional possession and linking) phases [1, 2]. In these contexts, the conventional approaches that restrict exploration to a 2D plane, e.g. by offering a pan and zoom interface, are among the most widespread solutions, mainly because they reduce the learning curve associated to full 3D control for both expert and casual users [3].

Creating an informative and engaging experience requires, however, to go beyond the pure visual presentation of the digital twins. In particular, annotations linked to the digital model are often used to provide better insights to the user [4]. Traditionally, such annotations let authors identify specific regions,

*Corresponding authors
e-mail: gobbetti@crs4.it (Enrico Gobbetti), ruggero@crs4.it (Ruggero Pintus), marton@crs4.it (Fabio Marton), moonisa@crs4.it (Moonisa Ahsan)

Fig. 1: **Overview.** Left: The user explores the scene using an interactive lens, and the best annotation under the lens is presented by playing the associated audio clip and showing the visual markup in overlay. Middle: when the user releases control, requests guidance, opts for automatic touring, or when no available annotations are under the lens, the system indicates the next best annotation using glyphs. Right: if the user remains inactive, the lens is moved towards the selected target. This approach can be used to generate intuitive tours through the data that dynamically respond to user actions, seamlessly transitioning from full user control to automatic navigation.

visually mark them with overlay text or drawing, and link them to metadata or other information that characterizes the significance of those regions [5]. However, finding relevant annotations, and presenting them in a comprehensible way without cluttering the display, and in a coherent order while conveying a context- and user-dependent narrative, is very challenging [6, 7].

Recently, visualization lenses, i.e., movable tools that provide alternative visual representations for selected regions of interest of a display [8], have shown to offer promising solutions for the exploration of annotated models. In particular, by displaying a single annotation at a time under the movable lens, selecting it using a recommendation system that takes into account the current camera position, current interactive lens parameters, and navigation history, a context-dependent clutter-free display can be achieved [7]. This approach, however, does not consider the relation among annotations themselves, and has thus limitations in the ability to prescribe presentation orders to define meaningful tours through the data [9, 10].

In this work, we introduce a novel approach for guiding users in the exploration of annotated 2D models by exploiting an annotation graph generated at authoring time (Fig. 1). Each graph node contains an annotation, in the form of a visual and audio markup of the area of interest, as well as the optimal lens parameters that should be used to explore the annotated area and a scalar representing the annotation importance. Directed graph edges are used, instead, to represent preferred ordering relations in the presentation of annotations. A scalar associated to each edge determines the strength of this constraint (Sec. 3). Such edges let us introduce storytelling features by letting each node point to the set of nodes that should be seen before presenting its associated annotation. At run-time, a user explores the scene with the lens, and the graph is exploited to select the annotation that has to be presented at a given time (Sec. 4). We call it the *best annotation*, to reflect it is the particular one which optimizes a set of selection criteria, that considers the current view and lens parameters, the graph content and structure, and the navigation history, through a novel technique that also takes into account topological distance among subsequently presented nodes in the annotation graphs (Sec. 5). The best annotation under the lens is presented by playing the associated audio clip and showing the visual markup in overlay. The use of audio clip to audibly present the additional information lets users focus on the visual content lens, without further clutter. When the user releases control, requests guidance, opts for automatic touring, or when no more annotations are available under the lens, the system points towards the next best annotation using glyphs, and potentially moves the lens towards it if the user remains inactive (Sec. 6). This approach can be used to automatically generate intuitive tours through the data that dynamically responds to user actions in real-time.

This article is an invited extended version of our contribution to the 8^{th} *Smart Tools and Applications in Graphics conference (STAG 2021)* [11]. Here, we not only provide a much more thorough exposition, but also significant new material. Our novel contributions include: a new representation of graph dependencies, that makes it possible to express hierarchical grouping and levels of abstraction; an improved scoring system that best preserves the ordering relations by exploiting topological distance in the annotation graph; an improved state machine for intuitive transition between interactive control and auto-touring features; and the seamless handling of audio markups. We also include a user-evaluation on challenging datasets.

While our methods are generally applicable to any 2D visualization, our motivating application is particularly in the cultural heritage domain, where it is essential to deliver informative and engaging real-time experiences to the general public in walk-up-and-use settings or in other similar easy-to-use web-based tools. In this work, we present an objective and subjective evaluation of our method for the exploration of stratigraphic relightable models (Sec. 7).

2. Related Work

Exploration of annotated models and interactive lenses are broadly studied topics within the visualization community. Here we focus only on the approaches most related to ours. For a wider coverage, we refer the reader to the established surveys on annotations [5], visualization lenses [8], and spatial interfaces [12].

Selecting and presenting relevant annotations in a comprehensible manner without clutter is one of the major challenges in effective visualization displays [6]. Displaying all of them at the same time is infeasible as it generates cluttering and cognitive overload [8]. Several attempts have been made to address

the challenges of overcrowded displays and improving intuitive interaction. Some authors proposed to guide users towards interesting areas by controlling the camera, creating animation paths, or defining fixed video tours [13, 14]. Others suggested enabling and disabling specific categories [15], as well as modifying the appearance (e.g., filtering data or using variable opacity), or distorting and zooming the images [16].

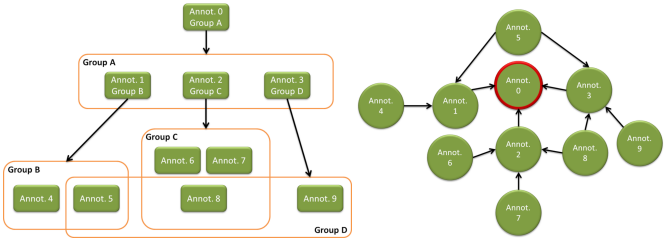
Serial temporal presentation to enhance the view [16] is one of the approaches that has been used to deal with overcrowded display, and, in conjunction with authoring or automatic determination of temporal precedence, it provides a way to deliver a narrative meaningful tours through the data [17]. Manually writing or defining fixed key-frames and forcing a single path is one of the most adopted solutions [13], which has also been used by touring through annotations [14]. This approach, however, leads to the generation of static videos rather than interactive experiences.

Interactive lenses, used widely in scientific and information visualization [8] offer a very flexible solution to deal with complex displays, as they support overview+details, focus+context and cue-based techniques [18]. Clutter is often reduced in combination with lenses by sub-sampling [19] or by selecting an annotation at a time [20]. Bettio et al. [7] recently proposed a novel approach for assisting users in navigating with the lenses, meanwhile also exploiting the data annotations. Their approach introduces a controller that guarantees maintenance of focus-and-context constraint by jointly adapting view- and lens-parameters, as well as a scheme to determine the next best annotation in the database based on the current view and lens parameters and the navigation history.

We build on these prior approaches, significantly extending the annotation representation, moving from a simple flat list of annotations to an annotation graph, in which the edges express semantic relationships among nodes, exploiting these relations for automatic data touring and generating guided suggestions. Annotation selection is based on a score that extends to annotated lens graphs, with the Degree-of-Interest (DOI) concept introduced by Furnas [21] for trees and extended by Van Ham and Perer [22] to graphs. Similarly to Gladisch et al. [23], DOI computation also takes into account the past behavior of the system. The camera-control work of Balsa et al. [17] is the most similar to ours, as it selects only a single item at a time from a viewpoint graph. Our annotation graph and scoring system is, however, targeted to support lens-based navigation of an annotated model, and has a different structure. In particular, we expand the approach of Bettio et al. [7] by introducing a dependency score to support hierarchical grouping, and a topology score to drive the system towards an orderly visit of the graph by penalizing changes in levels of abstraction of topic switches. Moreover, we introduce a new state machine design to seamlessly combine automatic touring with self-guided visits.

3. The audio-visual annotation graph

Traditionally, annotations are used to identify specific regions, linking them to metadata or other characteristic information [5]. In this paper, we want to exploit annotations for



(a) Hierarchical grouping (b) Graph representation

Fig. 2: **The annotation graph for hierarchical grouping.** Edges in the graph point to enabling nodes.

guidance and data presentation. Similarly to Bettio et al. [7], we associate to each annotation a *visual overlay* and an *external annotation description*, together with the parameters that should be used for an effective lens-based exploration of the annotated area. In this article, the visual overlay is simply an image or text that can be drawn over the model, while the external annotation description is a link to a hypertext with additional information. The exploration parameters are used for navigation control, and consists an *annotation importance* (i.e., a user-defined weight to associate a higher (or lower) likelihood that an annotation will be displayed before (or after) the others), a *lens and context area description* (i.e., the position and size of the best lens and camera-angle for viewing the annotated area), and a set of *rendering parameters*. For this work, focusing on stratigraphic relightable models, the rendering parameters include the layers that are displayed inside and outside the lens and the light configuration in terms of both direction and type (e.g., collinear or spot light); for the spot light we also specify the light beam aperture. Other rendering parameters can be defined (e.g., brightness, gamma value), also related to different rendering strategies (e.g., shape/color enhancement operators).

In addition, we also include with each annotation an *audio description*, which is an explanatory text that describes the annotated area, and is intended to be played when the annotation is visited. The audio clip can be generated by synthesizing the textual description, or be a pre-taped recording. In both cases, the audio clip duration defines the minimum time that the system considers should be spent for considering an annotation seen (Sec. 5). Using audio to describe the annotated area is particularly appropriate for our use case and interface design. In particular, using the audio clip rather than a displayed text to convey non-visual information allows us to let users concentrate on the model, and to produce a lean visual overlay when exploring the scene with the lens.

Using audio for enhancing a museum visit is very common, and it is employed in a range of solutions, from conventional audio guides presenting short burst of audio information at each stop [24] to virtual audio-visual visits [14]. Supplementing visual information with audio has also been shown to improve memorability [25]. However, usage of audio may not always be appropriate. For instance, handling multiple co-located users performing separate visits requires special care. The typical solutions in museum settings are the set-up of isolated display area, with space management complications and limitations in the number of active interactive displays, or the usage of head-

sets by individual visitors, locking them into isolated experiential bubbles with the risk of reducing inter-personal interaction [24]. For handling those cases, we can provide a purely visual experience, in which the annotation description is displayed on screen. Note that, since we use a lens, we cannot simply display the text in a separated area, as in classic annotated model presenters [15], since this solution would force users to lose their focus. Our current solution is to display the annotation in a small area under the lens. We plan to improve this approach by considering an optimal shape, placement, and scaling of the text box attached to the lens border, so as to reduce the masking of the annotated area, as done in external labeling techniques [26].

In order to specify a preferred presentation order, we introduce dependency links, transforming the annotation database into an *annotation graph*. In this representation, each node is an annotation, and directed edges point to a set of enabling nodes (one or multiple parents) that should be seen before visiting it. The presence of edges allow authors to define a preferred global order, that can be used to create a story like structure between annotations, or, e.g., to go from coarse to finer details as prescribed by the visualization mantra [27]. A weight associated with the edge, ranging from zero to one, defines the strength of the dependency between the nodes (see Sec. 5).

By using this graph representation, for instance, it is possible to structure an annotation database into hierarchical groups of nodes, to represent information at various levels of abstraction, as done for complex graph exploration [23]. Semantically, each node in the graph can be seen as a coarser representation of its children, and this translates into the fact that the annotation associated with a leaf node is best presented to the user only after its parents. This particular view of the dependencies helps guide authors in the definition of links, as they can proceed to structure annotations coarse to fine, or inversely grouping them from fine to course during their editing process.

Fig. 2 shows an example of hierarchical organization of a set of ten annotations. In particular, Fig. 2a depicts the idea of *hierarchical grouping*, where each annotation can represent a set (or group) of other annotations, which we can consider its children. As previously explained, visualization order depends on high level annotations that, once viewed, enable the visualization of the groups they represent. On the right (Fig. 2b), we present how the grouping and the corresponding visualization order is implemented through dependency edges; note that while all the authored hierarchical groupings can be expressed/transformed in a graph, not all the graphs can be transformed in a hierarchy of groups, e.g., some cyclic graphs. If an arrow points from a node B to a node A, it means that the visualization of B depends (with a certain level of dependency) on the fact that the node A has already been visualized. The annotation at the highest level is the first displayed annotation (we can consider it the root of the navigation), and it is depicted with a bold red contour in Fig. 2b.

The annotation attributes and their organization into a graph is exploited by our system to present information in a context-dependent and graph-dependent order during navigation (Sec. 4).

Note that authoring details are orthogonal to the subject of this paper. For the sake of completeness, we mention here that we annotate the models during the lens navigation as described by Bettio et al. [7]. The system allows users to move the lens to interesting areas, draw annotations with a simple image editor, and store them in an annotation database containing the lens and context area description, as well as the rendering parameters. The node table is then edited off-line by adding dependencies to nodes, and enriching the description of each annotation with an audio recording.

4. Interactive and guided lens-based exploration

At run-time, the user explores the annotated scene using a visualization lens that interacts with the scene by moving and scaling the focus area and activating relevant annotations. Since only a single context-dependent annotation is selected at a time, clutter is reduced. The sequence of selected annotations must be relevant to the current spatial context and maintain a flow, so that, more general information is presented before dependent details. We do that by running a state machine that exploits the annotation graph and responds to user actions. Our goal is to support the seamless transitioning between two behaviors. On one extreme, we would like the system to be capable of producing automatic tours of the data, by presenting annotations in a sequence, as for a video tour. On the other extreme, users should be allowed to explore the scene at their own pace, with relevant annotations appearing in sequence as the user moves to the annotated areas. In the common intermediate situation, we would like to support users that start with automatic touring, then explore the scene for a while, then restart auto-touring, possibly in other areas depending on their interest.

The state machine is made basically of two intertwined parts, one devoted to react to interaction, and one devoted to perform automatic tours. The user can interact with the state machine in three ways: by doing nothing (the user accepts what is proposed by the state machine), moving the lens (the state machine accepts what the user proposes), or sending a `Done` signal to communicate that the exploration of the current annotation is completed.

The machine is composed of 5 states: *Start, Anticipation, Goto, Show, Interact* (Fig. 3). The loop *Anticipation, Goto, Show,* constitutes the auto-tour part. From this loop the user can exit only by starting to interact with the lens.

The state machine employs a function *Find Next* that, given the current situation, identifies the next best annotation. From the *Start* state, the first annotation is selected and the lens is moved over it through the *Goto* state. During *Goto*, the lens position and the rendering parameters are interpolated from the currently displayed situation toward the target one, encoded in the node database together with the annotation. Note that, for the particular case of the relightable stratigraphic models used in this paper, adjusting the rendering parameters includes the selection of inner and outer layers and the update of the illumination settings in terms of light intensity and direction. This means that, when interpolation is complete, the model is displayed with the full visualization settings that annotation-author has stored in the database.

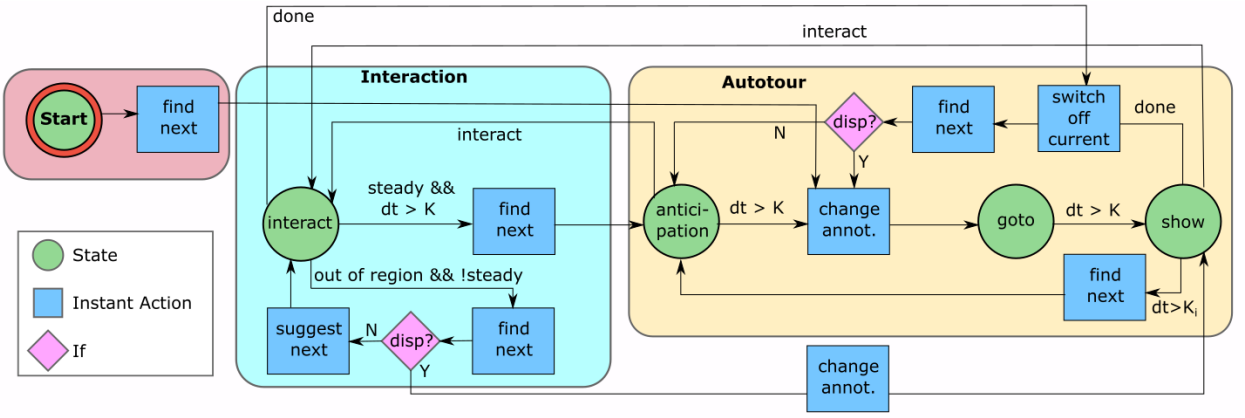


Fig. 3: **Annotation Navigation State Machine**. Two main navigation modalities have been implemented, i.e., the manual interaction (cyan box) and the auto-tour (yellow box). In the first mode the users freely move the lens, while in the latter they are guided through annotations that are automatically selected. To enter the auto-tour mode the users just stop the interaction with the lens interface; re-touching the interface will bring to the manual mode.

When interpolation finishes, the state changes to *Show*. The *Show* state displays the annotation for a content-dependent time K_i. In particular, if not specified by the user in the annotation database, this time is equal to a small setup time of half a second plus the duration of the audio clip associated with the annotation.

It is possible to exit from this state in three ways: by interacting with the lens, changing the state to *Interact*; when the allocated time elapses, entering the *Anticipation* state; or finally when the user signals that he has `Done` with the current annotation. This last operation is used to speed-up exploration in case the user is not interested in the current content anymore. To decide what to do next, the system evaluates if the next selected annotation is directly *displayable*.

An annotation is considered displayable if, by drawing it as an overlay, it can be reasonably well perceived by the viewer, without the need to resort to directing or leading cues to indicate where the annotation is located. The displayable condition, thus, requires checking whether the view is approximately at the same scale of the annotation (in this paper, within a factor of two larger or smaller in zoom factor), and at least some portion of the annotation is within the focus area of the lens.

If the next best annotation is displayable, the machine changes the state to *Goto* and the lens is moved to properly center the new annotation while remaining in the auto-tour loop. If the next annotation is not displayable, the state is changed to *Anticipation*.

The *Anticipation* state has a twofold purpose: it alerts the user with visual cues that the lens is going to move towards the next annotation and provides the users with information on where the next annotation is placed using visual hints (see Sec. 6 for details on visual signals and direction hints). From the *Anticipation* state, there are two possible exiting transitions: if a certain amount of time with no user action elapses, the system considers the next annotation accepted, and the auto-tour continues by changing to the *Goto* state; otherwise, upon user interaction, the system enters the *Interact* state.

During the *Interact* state, the user keeps visiting the current displayed annotation, possibly moving the lens. From this state it is possible to exit in three ways: being steady after the whole annotation time has elapsed, going outside the annotation area, or through the `Done` signal. Steadiness is considered as having completed the inspection, thus the system alerts the user by entering the *Anticipation* state. Instead, when going outside the current annotation area, an indication of the next best annotation is presented. Then, two events can produce the state change: if the user keeps moving but passes over an annotation considered displayable, the state changes to *Show* and the annotation is made immediately visible, or if the user stops interacting, after a small amount of time the state goes to *Anticipation* to present the new annotation. Finally, sending `Done`, produces a situation similar to auto-touring: if the next annotation is displayable the state changes to *Goto*, otherwise to *Anticipation*.

Note that, with this approach, we cannot distinguish if a user remains inactive after having inspected an annotation because the inspection has been completed or because the user is closely inspecting/pondering on the current view. In the latter case, repeatedly receiving a suggestion, through entering the *Anticipation* phase, might be considered annoying. To reduce this effect, we increase the time for receiving a suggestion by doubling the inactive timeout each time the user does not accept the suggestion by interacting during the *Anticipation* phase, up to a maximum timeout. By contrast, the timeout is halved each time a suggestion is accepted or a suggestion is requested, until we reach the default timeout. In this paper, the minimum and default timeout is 5s, and the maximum is 40s. An alternative solution would have been to remove the timeout, and ask explicitly the user to signal the completion of interaction viewing, which is now optional. This is still possible by configuring the timeout to (much) larger values. However, we consider in this work the small-timeout version, to test the typical setting of cultural heritage visits, which take into account the limited span of attention of visitors and the need to streamline visits in order to increase the visitor throughput while delivering enjoyable experiences.

During the visualization experience, thus, the system continuously performs two main tasks. The first is the selection (when required) of the next best annotation to display (Sec. 5). The

second is the management of the user activity through several device mapped interactions (Sec. 6). Those two elements drive the annotated model visualization and allow the user to seamlessly switch between interactive and automatic navigation.

5. Best annotation selection

During the navigation the system selects the next best annotation for the automatic tour using a scoring system. Following Bettio et al. [7], we assign to each recorded annotation node i a score $N_i = \gamma_i \sigma_i H_i$, where γ_i is the author-defined annotation importance, σ_i is the similarity score depending on spatial and semantic distance, and H_i is the history score depending upon the activity log of the active user, that equals to 1 when the node has not been visited in a recent time, and 0 when it has just been visited. H_i is thus initialized to 1 for all the not visited nodes. When a node is visited it goes immediately to 0, in order to avoid presenting again the same node, then smoothly gets back to 1 over a certain time, meaning that after a certain elapsed time the user tends to forget the content of a node, and could be presented again; this time can be set proportional to the amount of the duration of the visual-audio annotations, in order to avoid disturbing repetition of annotations before having seen the vast majority (or all) of them. More details of each of these individual scores are presented in the original publication [7].

In order to consider dependencies, we extend this formulation by multiplying the node score N_i by a dependency score δ_i, which takes into account node precedence relations and their weights, and by a topology score τ_i, which depends from level of abstraction distances, to obtain a final annotation score

$$S_i = \delta_i \tau_i N_i = \delta_i \tau_i \gamma_i \sigma_i H_i \tag{1}$$

5.1. The dependency score

The dependency score δ_i takes into account node precedence relations and the corresponding weights. It expresses the fact that the author would prefer a given node to be presented after its enabling nodes, with a weight depending on the edge strength. This is achieved by taking the fuzzy logic AND (i.e., min operator) of a per-edge quantity that expresses if the node has already been presented, and which strength should have this information. The dependency score of node i is thus given by

$$\delta_i = \min_j \left(1 - e_{ij} H_j\right) \tag{2}$$

where j loops over all enabling nodes, e_{ij} is the author-selected edge weight linking node i to node j, and H_j is the history weight of the node j. For a strong dependency with weight $e_{ij} = 1$, when the parent node is not visited ($H_j = 1$), the dependency weight δ_i is 0, thus blocking the presentation of the node. When, instead, the parent is visited ($H_j = 0$), the node has $\delta_i = 1$, so the node is completely enabled, and is thus included in the potential candidates for selection. When dependencies are, instead, weak, i.e. $e_{ij} < 1$, the dependency score will not reach 0, permitting, with low probability, the visit of a node even if all the enabling nodes have not yet been visited.

5.2. The topology score

The aim of the topology score is to provide a configurable orderly visit of the annotation database, without large semantic jumps among proposed content. Since the graph structure encodes relations among annotations, e.g., by grouping and definition of levels of abstraction, we define a weight that favors proximity relations in the graph. For instance, it is preferred to visit children and siblings of the last active node, as the content of their annotation is likely more strictly related to what just presented than the content of other annotations in the graph.

For that purpose, we can define the semantic distance among two annotations as the topological distance between the two nodes containing them, which is the length of the shortest path among the two nodes. In order to give the same proximity value to siblings and children, we virtually insert edges among siblings. Note that this can be done directly in the distance computation method, without any structure modification, by setting the distance among siblings to be one rather than two, as it would be required if we were forced to go up to the parent and then down again.

Given the current candidate node i and the last visited node j, the topology score τ_i is then defined as:

$$\tau_i = 1 - \beta \frac{\min\left(d_{ij}, d_{MAX}\right)}{d_{MAX}} \tag{3}$$

where d_{ij} is the shortest path in the graph between node i and node j, while d_{MAX} is a normalization factor used to define the maximum allowed topology distance, which is independent of global graph size (i.e., adding or removing distant nodes), and is defined at authoring time (Fig. 4), and β is a scalar that is equal to zero if the user is interactively moving the lens and one otherwise.

In order to speed-up run-time evaluation of the topology score, we precompute in advance all the mutual distances between graph nodes with the Floyd Warshall Algorithm [28], which provides the shortest distances between every pair of vertices in a graph.

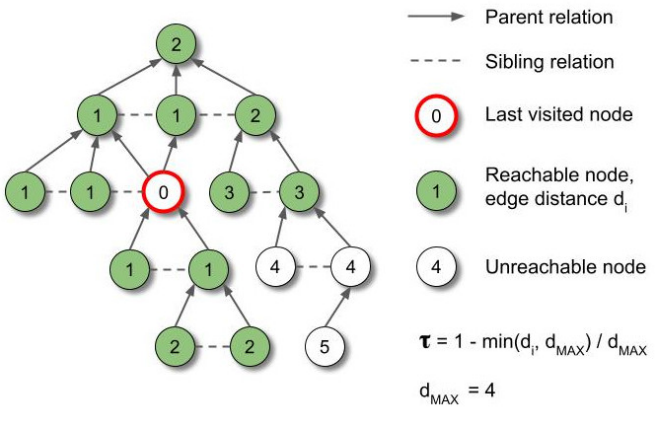


Fig. 4: **Topology distance.** Annotation graph with parent and sibling relations. Topology distances computed with respect to the red node. Topology score is derived by the depicted formula with $d_{MAX} = 4$.

The scalar β allows us to tune the behavior of the system depending on the current situation. Taking into account the topo-

logical distance in the annotation graph is of primary importance during automatic touring or when the user explicitly asks for suggestions, as it is very reasonable to strongly favor semantic proximity over spatial proximity. On the other hand, when the user is freely moving, especially far from the last displayed annotation, switching subjects in order to show locally relevant information is often the expected behavior, as users typically move for new knowledge discovery, also signaling with their motion that the current story flow can be modified.

5.3. Choosing the best annotation

The next best annotation is then selected by taking into account the scores S_i, which determine the suitability of each particular annotation for a given context. When scores are widely different, the annotation with the highest score must definitely be preferred, as lower-scored annotations would seem out-of-context. However, when scores are very similar, several different annotations might be considered suitable. This is not an unlikely situation, especially when no annotation is overlapping with the current lens. To take into account this situation, rather than just selecting the annotation with the highest score, we perform a stochastic selection among a small set of nodes that have a similar high score. In particular, we select a cutoff score S_c equal to a fraction C of the maximum achieved score, and extract the subset of K nodes which have a score higher than this threshold. We then assign to each node in this subset a picking probability $p_k = \frac{S_k}{\sum_j S_j}$, and select the next best annotation according to this probability. In such a way, the exploration is open to a wider range of possible paths, while maintaining the author dependency requirements.

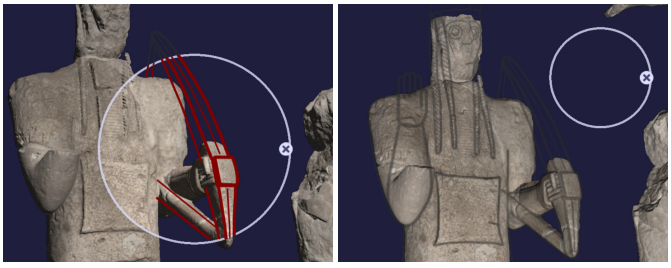
When the cutoff C is set to 100%, there is no stochastic selection, and the system, let alone, always repeats the same tour. In this paper, the cutoff C is, instead, tuned depending on the current situation. In particular, it is equal to 95% when searching for annotations while the user moves the lens, and to 60% otherwise. This makes it possible to choose among a large number of likely paths when performing automatic tours or the user is requesting suggestions, increasing the variability of the exploration experience, while avoiding the selection of incoherent solutions. This variability is important for casual visitors, as it makes the visit more engaging and less repetitive (see Sec. 7.2 for an evaluation of the effect).

6. User interface and device mapping

Our user interface for lens-based exploration requires minimal user input, and can be mapped to input devices in a variety of ways. For lens and camera movement, we employ the recently-introduced approach of Bettio et al. [7], that couples lens and camera motion to always ensure a good focus-and-context placement of the lens within the view. Using this approach, the user manipulates only the lens, changing its position and radius, and the system automatically computes the camera translation and scale updates in order to always maintain a good focus-and-context situation. In our current implementation, we realized both a multi-touch solution and a mouse-controlled version.

At the start of the navigation, the lens is moved to the best position for the first annotation selected (a selected root node in the annotation graph). Then, the user can pan the lens by a one-finger pan gesture (or by using the left mouse button), and use pinch-to-zoom (or the wheel or up/down movement holding the right mouse button) to modify lens scale. In both cases, the controller adjusts camera position and zoom to maintain the focus-and-context condition. The state machine, running in background, reacts to lens and camera motions to change interaction modes and update the display, as detailed in Sec. 4.

The user interface also includes additional features that implement all the characteristics of the controller. In particular, during the *Show* and *Interact* state, the lens has always a small button with a cross that, when clicked (with a touch or a left mouse click), triggers the `Done` signal (Fig. 6). That signal communicates to the system the fact that the user has finished inspecting the current annotation, and asks to visualize the next best annotation. The `Done` button is not available during the *Anticipation* and the transition state *Goto*.



(a) (b)

Fig. 5: **Annotation Rendering.** Rendering within the lens shows the original annotation colors, instead for content outside of the lens the colors are transformed into grayscale.

Visual signals also enrich the interface during transitions. The *Anticipation* state, in particular alerts the user that the current annotation is going to be replaced by the next one. To convey this message, we progressively change the color of the lens boundary from white to red, to alert the user that the system is about to go to the next annotation if the user does not restart to interact with the lens.

The other important visual signals concern annotation display. The representation used for the annotation depends on whether it is the currently active annotation or the next proposed one, as well as on whether the annotation is within its display range or outside it.

The current annotation, when considered displayable, is rendered with full color within the lens and dimmed outside the lens (Fig. 5a). In the *Anticipation* state, as well as in the *Interact* states when the lens moves out of the current annotation, both the current annotation and the next one are displayed. When the next annotation becomes current, the previous one disappears.

One of the main problems to tackle is the display of annotations that are not currently visible (e.g., far from the lens, outside of the view frustum, or outside of the zoom range). This occurs, in particular, when the user must suggest the next annotation and provide directions towards it.

The problem of displaying out-of-view objects is subject

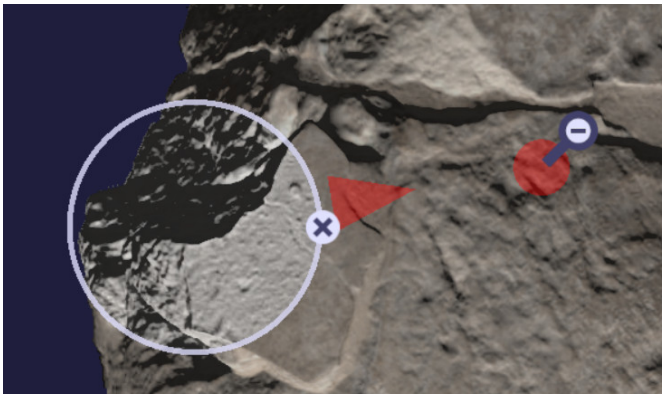


Fig. 6: **Interface glyph**. Glyphs rendered during interaction with the lens outside the current annotation area. A cross button, placed over the lens border, can be used to communicate `Done` signal. Red arrow and spot indicate the direction and position of the next annotation center. Hand lens with minus sign, indicates necessity of zoom out (plus sign would be used for zoom in).

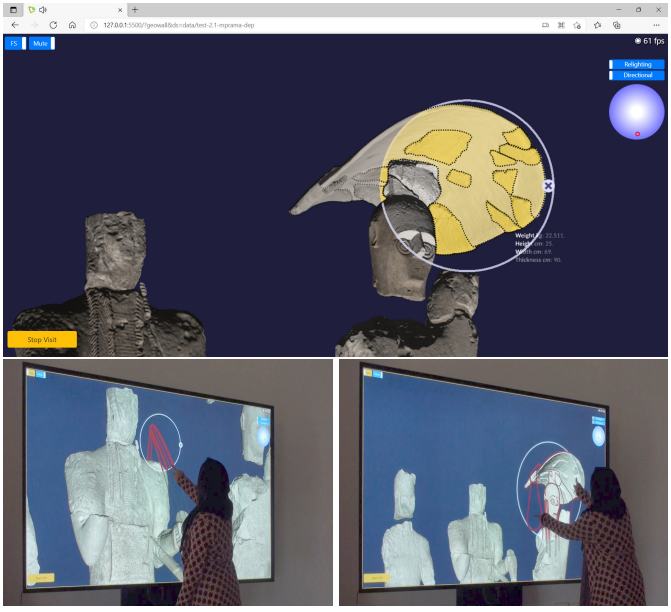


Fig. 7: **Multiplatform application.** The same web-based implementation is used for multiple use cases. The top image shows the application running inside a web browser on a desktop platform. The bottom images show two frames from the recorded video of an interactive session on a large touch screen for a walk-up-and-use museum installation.

of much research, and the main techniques are distinguished among the use of *leading cues* attached to the target, meaning that some part of the cue is always spatially connected to the target, and *directing cues*, mostly fixed in the user's view and giving the user a general direction to the target instead of providing a direct path to follow [29]. In this work, we use a combination of both.

In particular, to indicate to the user an annotation that cannot be displayed, we use a combination of three indication glyphs: arrow, spot and zoom (Fig. 6). A dynamically oriented arrow placed around the lens, similar to a compass needle, points in the direction toward the next annotation center. It acts as a directing cue, but is not fixed in the view but attached to the lens, since it is, at the same time, the object that controls navigation and the area where the user is focusing. Moreover, if the annotation cannot be displayed but is within the viewport (e.g., because out of zoom range), we use as leading cue a red filled dot placed at the center of the new annotation. A zoom indicator, a small hand lens with plus or minus sign, shows if a change of zoom is required to properly see the annotation.

In addition to annotation overlays and leading and directing cues, we also employ audio to convey semantic information without overloading the visual channel. This is particularly important for our lens-based interface, since fixed text areas would require users to move their focus out of the lens, while moveable text areas attached to the lens would provoke considerable masking in the lens context.

7. Implementation and results

We implemented the proposed approach on a web-based platform. The inspected model, the annotations, and all corresponding metadata are made available by a standard web server to a web client running in a browser on top of WebGL2, a JavaScript API that closely conforms to OpenGL ES 3.0 and can be used in HTML5 <canvas> elements without requiring plugins. The client can run in regular web browsers (we tested, in particular, Firefox, Chromium, and Chrome on both Windows and Linux platforms, and Edge on Windows), supporting both mouse or multi-touch input using the TouchEvent API. Fig. 7 shows how we can adapt to multiple use cases, including full-screen display on large multitouch installations, and desktop or tablet visualization for web distribution.

While our methods are generally applicable to 2D exploration use cases, the particular implementation discussed here refers to multi-layered relightable image models. These models consist of a series of registered multiresolution image-based layers of shape and material information. These layers can come out of a variety of pipelines which produce parametric information, including both RTI representations and spatially-varying normal and BRDF fields, possibly obtained by fusing multi-spectral data.

The preparation of the relightable images and their layers, all the annotations and the authored annotation grouping (node and edge attributes) in the relation-dependent, hierarchical graph, and all the associated audio clips are done off-line. They are stored in a repository that contains the set of image layers, the audio clips, a configuration file that manages the arrangement of those layers, and a file that includes both the text annotations with all the graph structure. At run-time, the viewer loads a scene description that includes the annotation database, and starts navigation by placing a lens at the root position.

We have tested our system on a variety of models. In this paper, we provide an objective and subjective evaluation centered around the exploration of a cultural heritage scene, with the aim of analyzing and assessing the suitability of our navigation system for casual users, as typical on museum web sites or walk-up-and-use installations. The accompanying videos provide an illustration of the behavior of our method, as well as sample footage from our user tests.

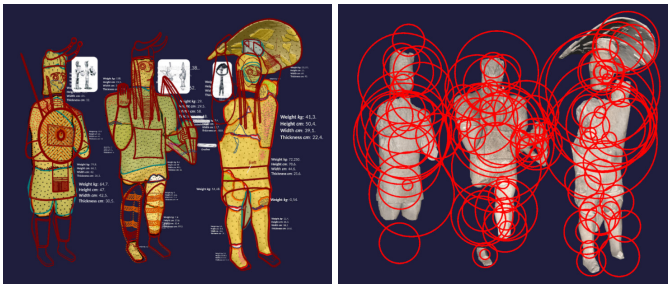


Fig. 8: **Mont'e Prama Dataset**. Three statues from the Mont'e Prama collection of prehistoric stone sculptures (from left to right): Warrior n.3, Archer n.5, and Boxer n.15. The left image shows the content of all the annotations of the database, while the right image shows the corresponding lenses.

7.1. Dataset preparation

The test dataset is a relightable multi-layered rendered image of three representative models from the Mont'e Prama collection of prehistoric stone sculptures [30, 31]: Archer n.5, Boxer n.15, and Warrior n.3 (Fig. 8). The annotation database concerns reconstruction hypotheses, artistic details and part descriptions. It contains 108 annotations at multiple scales that form 25 annotation groups; in total we have 107 edges that express groups and nodes dependencies. An illustration of all annotations in one single frame and the density of all lenses is shown in Fig. 8. All the annotations were given the same authored importance.

In creating our annotation database, our goal was to enrich the plain visual representation with pieces of interesting information taken from the historical and semantic knowledge in the Mont'e Prama related literature. The intention was to ensure that the annotations, in terms of visual and audio content, are easy to interpret for a common user, without any prior knowledge of paleo history.

Starting from the source information contained in a dedicated archaeological books series [32], we created a variety of annotation, that can be concisely classified into (A) graphical extensions of missing parts, limbs and accessories of the statues *(total n. 22)* (Fig. 9a), (B) prominent regions of peculiar patterns and designs *(total n. 11)* (Fig. 9b), (C) highlighted areas with particular conservation states (e.g., showing well preserved parts for virtual reconstruction or their dimensions) *(total n. 54)* (Fig. 9c), (D) visual pointers of biological phenomena unseen to the naked eyes *(total n. 13)* (Fig. 9d), (E) frames marked with exclusive segments for additional historic and sculpting details (e.g. highlighting fine carving details over the surface together with information about sculpture techniques and tools) supported by pictorial references or images (e.g., comparison with small bronze statues) *(total n. 08)* (Fig. 9e). Each annotation contains both a visual markup, intended lens position and rendering parameters, and an explanatory audio clip.

7.2. Scoring system analysis

The proposed framework allows one to mix purely automatic navigation with interaction, since the user may take control of the lens during any path, and auto navigation restarts from the new user-updated lens and view configuration. We show this behaviour in Fig. 11. The transitions marked with red arrows depict lens movements/positions decided by the user (not by the automatic generated path); the next annotation selected by the automatic algorithm (transitions marked with green arrows) takes into account the dependency graph and the history, while being consistent with the lens positioning provided by the user. In Fig. 11, after the first three automatic frames, the user interrupts the automatic navigation three times, in order to move and inspect all the three statues. The accompanying video shows additional examples of this behavior, in which we seamlessly move from automatic touring to interactive exploration, and, each time, the tour restarts taking into account the possibly largely modified local context.

In order to evaluate the behavior of our scoring system, we tested the automatic navigation without the free movements introduced by the manual exploration (see Sec. 7.3 for a detailed analysis of users' interaction and their subjective validation).

In this setup, since the graph was authored with a single root node (the statue overview) required for all further inspection (dependency weight=1), we expect that the navigation always starts from the root and, from there, a relevance-based order would be followed by navigation, taking into account graph hierarchy and node/edge priorities. In addition, we expect our navigation to enable a good level of variety, due to the stochastic aspects of our annotation selection (Sec. 5.3). We performed 20 automatic tours, each of them visiting 20 nodes, always starting from an initial position at the center of the screen and viewing all the scene in the viewport. Despite the same initial conditions, the 20 tours visited a total of 69 nodes with respect to the 108 contained in the graph. This fact shows how a stochastic component in path selection avoids full repetitiveness, providing different exploration experiences to the users, also in fully automated mode.

Fig. 10 (bottom three rows) shows three runs of the auto-navigation, with time going from left to right. It is clear how the first view is always the same, i.e., the graph root presenting the annotation related to the whole set of statues. Even if the lens is centered, providing a higher node priority to the center statue, there are several situations in which one of the other statues is selected first, due to our selection strategy with picking probability proportional to weight and our loose β value in this mode of operation. From there, the navigation continues with a spatial and semantic consistency, e.g., if a high-level node of a statue has been visited, the navigation continues with higher probability in the leaf nodes of that same statue. The semantic choice cooperates with the spatial one; if the visualization of a statue's annotation enables the visiting of the detail nodes of that statue, the visiting of nearby details of another statue still remain blocked until their enabling nodes have been enabled. Thus, the authored hierarchical grouping of the annotations enables the introduction of semantic constraints.

Conversely, when edges are not present, as in previous work on lens navigation [7], such constraints are not possible, and the next best annotation in a navigation path may be selected from a nearby statue based on pure proximity consideration. We repeated the 20 runs with 20 annotations each, using the same database, but with edge dependencies disabled. In such a

(a) Graphical Extensions (b) Patterns and Designs (c) Cons. (d) Biological Phenomena (e) Historic and Sculpting details

Fig. 9: **Annotation Classes**. We create a variety of annotation classes, i.e., (a) graphical extensions of missing parts, (b) regions of peculiar patterns and designs, (c) highlighted areas with particular conservation states, (d) visual pointers of biological phenomena, and (e) historic and sculpting details.

Fig. 10: **Automatic navigation**. Top row (yellow outline):an example of automatic navigation without using the dependency graph. The path proceeds by going from an annotation to the most similar one, without taking into account semantic aspects (e.g., same statue, from more general to specific annotation). Other rows (blue outline): several examples of automatic navigation with the dependency graph. All exploration paths start from the same annotation, and all tours share a similar flow, dictated by authored graph dependencies. Nonetheless, they introduce variations due to our stochastic next-best annotation selection process. The dependencies introduce semantic aspects, in this example favoring the presentation of a statue's detail after presenting its overview.

situation, we explored a total 72 nodes. Since more degrees of freedom are available due to the lack of edges linking to enabling nodes, the number of visited nodes is slightly larger. However, the paths are less structured, as they jump more frequently, for instance, from one statue to another. The first row of Fig. 10 shows an example of that kind of navigation, where edges are removed, and navigation proceeds purely by selecting the most similar annotations. Without taking into account a hierarchy of nodes, the storytelling aspect of the automatic navigation might get lost, as also demonstrated by our dedicated user study (Sec. 7.3.1).

In order to better understand how the next annotation is selected, and to have a more clear idea about the contributions of the single weights to the final annotation ranking score, we also launched several autotours, and we collected a series of data to compute the Pearson correlation coefficient $\rho_{S,x} = {cov(S,x)}/{\sigma_S \sigma_x}$ between the final score S and the individual components x of the scoring system of Equation 1. In particular, we consider the author-defined annotation importance γ, the history score H, the dependency δ, the topology weight τ, and the three weights that produce the similarity value σ, i.e., the lens overlap σ_{lens}, the context overlap σ_{cont}, and the location similarity σ_{loc}. The results are reported in Fig. 12. As we can see, there is a good balance in the different terms, but the three most important factors are the *Overlap*, *Topology*, and *Location* weights. It is important to note that overlap and location are closely related to selection by spatial proximity, while topology relates to semantic continuity.

7.3. User Study

The proposed navigation framework has been obtained by combining two main elements. From the narrative side, we have the audio-visual annotations and their structured organization, while from the purely visual data we have the multi-layered 2D model. From the user point of view, it is extremely challenging to assess and validate the combination of a narrative element and the interface that drives the communication between that and the user. This is mainly due to the lack of reliable

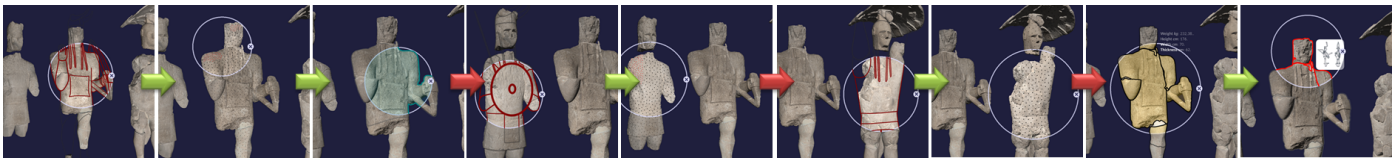


Fig. 11: **Mixing automatic and free exploration**. Our framework enables both automatic and free navigation. As soon as the user moves the lens (transitions marked with red arrow), the automatic navigation stops. When it restarts (transitions marked in green), the next frame is selected by taking into account both the dependency graph, the navigation history, and the user-updated lens and view configuration.

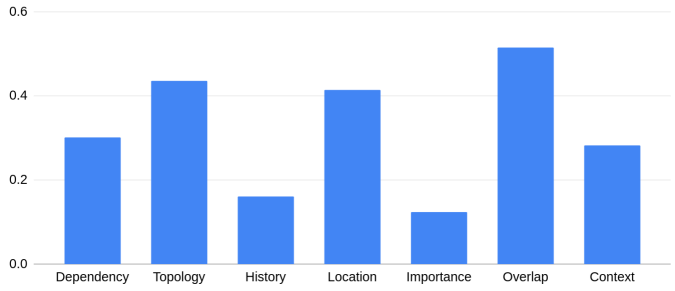


Fig. 12: **Score vs Weights Correlation**. We show the Pearson correlation coefficient between the final annotation score and each factor that contributes to that score. We can see that the three most important factors are the *Overlap*, *Topology*, and *Location* weights.

and standard metrics or practices that have enough consensus when assessing interface usability together with content user understanding. Considering and evaluating each part separately might be a good way to quantify their contribution to the user experience [33]. However, this approach does not take into account the effects that only arise because of the combination of several elements. If those components increase in number, the combinatorial nature of the problem makes the evaluation even more complex, unreliable, and non-practical.

For this reason, we concentrate here on answering two main research questions that are connected to the main differences between this work and previous ones.

The first question is whether the introduction of an annotation graph, with edges connecting annotations to their enabling nodes, leads to explorations that are perceived as an important improvement over presentations using methods that only considered an individual list of annotations (e.g., [7]). This question is explored through our *Autotour* test (Sec. 7.3.1).

The second question is, instead, related to the overall user experience. In particular, we want to investigate whether casual users remain active or passive in presence of system that provide both automatic touring and interactive exploration, and if they prefer a system that actively follows their actions or prefer to limit their interaction to local investigations inside an inflexible authored story. This question is explored through our *Interaction* test (Sec. 7.3.2).

7.3.1. Autotour Test

Goal. The purpose of this test is to understand, from the user perspective, if the introduction of the structured annotation graph increases the user experience during the *Autotour* mode compared to an automatic navigation that is free and ignores the semantic relationships between annotations. In this setup, in order to have a more controlled experiment, user interaction is not considered.

S1-G	GRAPH Visit is more engaging than FREE Visit.
S2-G	GRAPH Visit held my attention more than FREE Visit.
S3-F	GRAPH Visit is more boring than FREE Visit.
S4-G	Information presentation is more clear in GRAPH Visit than in FREE Visit.
S5-G	GRAPH Visit presents the information more organically than FREE Visit.
S6-G	GRAPH Visit provided me with more intellectual stimulation than FREE Visit.
S7-G	GRAPH Visit, more than FREE Visit, motivated me to learn more about the Mont'e Prama collection
S8-F	GRAPH Visit, more than FREE Visit, presents the cultural heritage content in a more scattered way.
S9-F	GRAPH Visit is more distracting than FREE Visit.
S10-G	The story told by GRAPH Visit is better structured than that told by FREE Visit.
S11-G	In GRAPH Visit I gained more knowledge than in FREE Visit.
S12-G	I enjoyed GRAPH Visit more than FREE Visit.

Table 1: **Autotour Test - Statements.** List of statements in the *Autotour* evaluation Likert-scale questionnaire. In order to avoid the agreement bias, half of the participants were presented the questions in their reverse form, i.e., swapping GRAPH and FREE as the preferred method.

Configurations. We produce two types of videos of automatic exploration of the digital model. One is obtained by launching the *Autotour* mode that uses the structured annotation graph, while the other produces an automatic exploration by completely ignoring annotation semantic relationships encoded in edges, therefore approximating the method of Bettio et al. [7] that works on a flat annotation database. The content presented in the two videos starts exactly from the same database in terms of visual representations (shape, color, illumination, etc.), texts, drawings, and audio clips. The main difference between the two videos is the way the information has been selected and organized for presentation. Each visit has an equal length of two minutes. We have produced many different *Autotour* navigations with the two modalities, in order not to have biases produced by a particular exploration run. We call the two modalities *GRAPH* visit and *FREE* visit.

Tasks. We ask participants to take a look at the two virtual visits of a set of three statues from the Mont'e Prama collection,

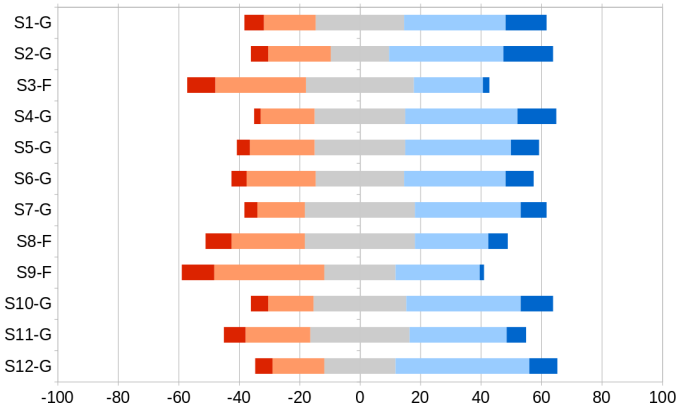


Fig. 13: **Autotour Test - Evaluation**. Histograms of responses for the statements in Tab. 1. Responses are color mapped from left (dark red, *Strongly Disagree*) to right (dark blue, *Strongly Agree*).

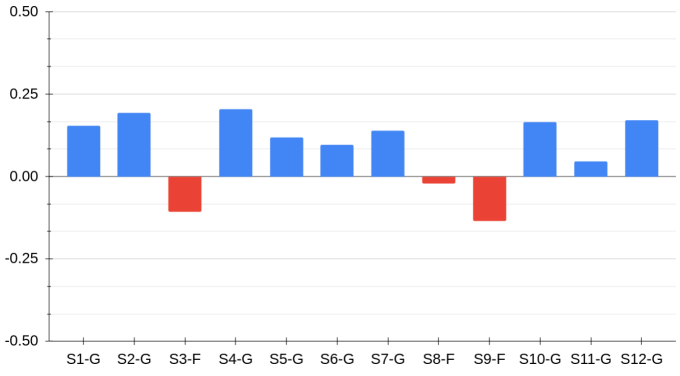


Fig. 14: **Autotour Test - Statements Score**. Scores obtained by each statement in Tab. 1. Positive scores mean agreement, while negative scores mean disagreement. In blue are statements that favor the *GRAPH* visit, while in red are those that favorite the *FREE* visit. In all statements, users agree that *GRAPH* visit is better than *FREE* one.

and to build an opinion on which of them they prefer. We will ask them several questions to understand that opinion.

Design. The test is subdivided in three phases. We first ask general questions to the users, in order to understand the type and distribution of the participants. In the second part, we blindly show participants two videos of a navigation of a digital model. The participants don't know which video is the *GRAPH* or *FREE* mode; the users don't even know the details of the two modalities, they only know that the two videos present different model explorations. From the produced videos in the two modalities (with or without structured annotation graph), for each user we randomly pick one example from each modality, and we randomized the video presentation order. Finally, we ask several questions to understand which videos/exploration they have preferred. We design the questionnaire as a Likert Scale [34] form with a series of twelve statements (Tab. 1), with five possible choices, i.e., *Strongly Disagree*, *Disagree*, *Neutral*, *Agree*, *Strongly Agree*. The statements are marked as $SX-G$ or $SX-F$ depending on the fact that a positive feedback is respectively given to the proposed solution (*GRAPH*) or the reference navigation strategy (*FREE*). The questions are inspired by Othman's work [35] about measuring visitors' experiences and engagement in museum visits. To avoid the agreement bias, i.e., the tendency of a respondent to agree with a statement when in doubt, half the respondents were presented with the questions reversed, i.e., swapping *GRAPH* and *FREE* references in their formulation. To simplify the presentation of the results, we have transformed back the order for the half cases in which we inverted the G/F order. In addition, some questions are slightly similar or opposite to each other to create a redundancy that is useful to test if the user has given consistent responses. We take this into account in the computation of the questionnaire consistency score.

Participants. The group of participants consists in 140 users (65% female and 35% male). The 3.6% are high school graduates, 5% with an associate's degree, 29.3% with a Bachelor's degree, 41.4% with a Graduate or professional degree, and 20% have a PhD. About 84% have a STEM background, while 10% of them come from the Humanities field. They were recruited using a mailing lists across various leading institutions involved in both Computer Science (specifically Computer Graphics and Visualization), CH studies, and applications. Through direct mailing, we have also tried to include participants representative of the general public, with a more heterogeneous background. They are researchers (15%), students (22.1%), teachers/professors (22.1%), IT professionals (2.1%), developers (5.0%), house wives (4.3%), and others (29.3%), which include freelancers, technologists, managers, and unemployed people. The age is ranging from 18-25 (41.4%), to 26-35 (29.3%), 36-50 (18.6%), 51-64 (10%), and over 65 (0.7%). We also have a heterogeneous set of people in terms of familiarity with museums/exhibitions and virtual museum presentations. About 60% of them have visited a museum last year, but 45% of them have no familiarity with virtual museum presentations; for the 60% of them, this is the first time they try an interactive setup similar to that proposed in this paper. Finally, half of them did not have any knowledge of the Mont'e Prama collection presented in the test.

User evaluation. We evaluate the *Autotour* test from three points of view, i.e., graphically, by a scoring system, and by computing the Cronbach's alpha reliability of the questionnaire. We choose the Cronbach's alpha since it is the most commonly used metrics to assess the internal consistency of a questionnaire made up of multiple Likert-type scales [36, 37]. First, we plot the histogram of responses for each statement (Fig. 13). The responses are color mapped from left (dark red is *Strongly Disagree*) to right (dark blue is *Strongly Agree*. It is clear how the $SX-G$ statements are more towards the *Agree* and *Strongly Agree* part, while the $SX-F$ statements contain more disagreement form the user. These results confirm that the user prefers more the *GRAPH* than the *FREE* navigation; this can be seen in the last statement, which explicitly ask the user the preference between the two exploration strategy. Here, 53.6% of the participants prefer the *GRAPH* Visit, 23.6% have a neutral opinion, while only 22.8% prefer the *FREE* Visit. In order to assign a numerical score to each single statement and a global score to the entire test, we linearly map each of the five responses to

scoring value, as typical for Likert scales [34]. In our case, respectively, from *Strongly Disagree* to *Strongly Agree*, we assign $-1, -1/2, 0, 1/2, 1$ values. The statement score is the average of the responses received by participants. As illustrated in Fig. 14, the statements marked as $SX-G$ obtain a positive score, while the statements that judge positively the *FREE* Visit (marked as $SX-F$), received a negative score. This, again, confirms that in each statement the users prefer the *GRAPH* Visit. In order to compute the final global score, we take the average of statement scores, after negating those marked as $SX-F$, obtaining a value between -1 and 1. A positive global score would mean that the users prefer our proposed automatic exploration system, a negative score that they prefer the other one, while a close to zero score would mean no preference. The final global score is 0.55, showing a very marked preference for the *GRAPH* version. We found that the reliability of the questionnaire is very high, with a Cronbach's alpha equal to 0.91. Since some questions are by design redundant, and since this can cause a bias in the Cronbach's alpha computation, we have also estimated the reliability by removing statements $2, 5, 10$, and 12; the Cronbach's alpha becomes 0.81, which is still very high. The user test thus confirms that the more coherent order induced by the graph, as evaluated in Sec. 7.2, leads to a perceivably improved experience.

S1-A	ADAPTIVE exploration is more engaging than FIXED exploration.
S2-F	FIXED exploration held my attention more than ADAPTIVE exploration.
S3-A	FIXED exploration is more boring than ADAPTIVE exploration.
S4-A	Information presentation is more clear with ADAPTIVE exploration than with FIXED exploration.
S5-F	ADAPTIVE exploration, more than FIXED exploration, guides you toward new annotations far from the region you want to explore.
S6-A	ADAPTIVE exploration provided me with more intellectual stimulation than FIXED exploration.
S7-F	FIXED exploration, more than ADAPTIVE exploration, motivated me to learn more about the Mont'e Prama collection.
S8-F	FIXED exploration, more than ADAPTIVE exploration, follows better your exploration intention.
S9-A	FIXED exploration is more distracting than ADAPTIVE exploration.
S10-F	They story told by FIXED exploration satisfies you more than that told by ADAPTIVE exploration.
S11-A	With ADAPTIVE exploration I gained more knowledge than with FIXED exploration.
S12-F	I enjoyed more FIXED exploration than ADAPTIVE exploration.

Table 2: **Interaction Test - Statements.** List of statements in the *Interaction* evaluation Likert-scale questionnaire. In order to avoid the agreement bias, half of the participants were presented the questions in their reverse form, i.e., swapping FIXED and ADAPTIVE as the preferred method.

7.3.2. *Interactive navigation test*

Goal. We aim to compare a classical exploration based on fixed authored tours with the new proposed solution where the tour

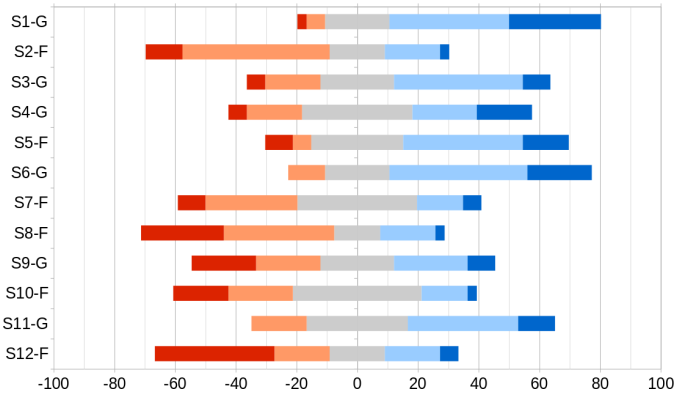


Fig. 15: **Interaction Test - Evaluation**. Histograms of responses for the statements in Tab. 2. Responses are color mapped from left (dark red, *Strongly Disagree*) to right (dark blue, *Strongly Agree*).

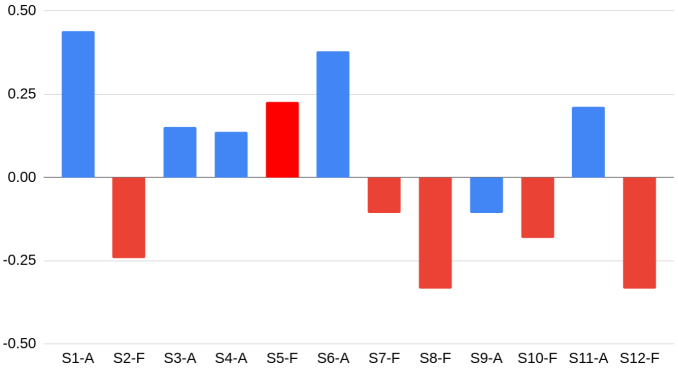


Fig. 16: **Interaction Test - Statements Score**. Scores obtained by each statement in Tab. 2. Positive scores mean agreement, while negative scores mean disagreement. In blue are statements that favorite the *ADAPTIVE* exploration, while in red are those that favorite the *FIXED* navigation. Apart from statement 5 and 9, users agree that *ADAPTIVE* visit is better than *FIXED* one.

is adaptively adjusted in response to user actions. For more generality, rather than restricting our comparison to the fully static video presentations proposed by systems such as CHER-Ob [14], we chose as a term of comparison the slightly more flexible interruptible video navigation method popularized by ArtMyn [38], which allows users to pause the video presentation to perform local exploration. In addition to analyzing user preferences, we also want to investigate whether casual users remain active or passive in front of these presentation systems.

Configurations. We configure two types of interaction experiences. In the first one, called *FIXED*, we show participants an image of an artwork, and we guide them through a pre-established and pre-recorded (but interruptible) navigation of a sequence of annotations attached to it. At any moment, the user can interrupt the navigation, and interact with the virtual environment to inspect the database, e.g., to look in more detail at some areas discussed in the pre-recorded story. After interaction, the automatic navigation continues from the point where it was stopped (as in the navigation method popularized by ArtMyn [38]), following the fixed pre-recorded annotation path. In the second interaction test, called *ADAPTIVE*, we adaptively select and present annotations with the methods presented in this paper, which allow users to freely mix interaction with guided

touring.

Tasks. The experiments consisted in letting users to freely explore the annotated sculptures, after a minimal training and without external direction. Users were told that their goal was simply to enjoy the experience and acquire information at their own pace in a prescribed short limited amount of time, exploiting the audio-visual annotations provided by the system, and using the interaction capabilities of the lens-based interface. This reflects well the scenario of a walk-up-and-use experience in a museum setup, as well as the situation encountered in museum web sites.

Design. The test is subdivided in two phases, focused on the interface usability and the presentation rationale/order. In the first phase, each participant actively tests the two exploration modalities, i.e. *FIXED* and *ADAPTIVE*. The modalities are presented to the participant in a random order. First, we make users familiar with the interface and the navigation task; so, before the actual test, users receive a one-page instruction describing the overall test, the interface, and the user-interface mapping; they are also allowed to test the tool without performing the task. After that, the exploration task is performed with the two configurations, and a series of variables are recorded to measure the user experience, i.e., number of annotation visited (manually or during an autotour), autotour or interaction time, etc. Each test has a fixed duration of 3 minutes. At the end of both the two interaction experiments, the participants were asked to fill a Likert scale questionnaire with five options for each question, i.e., *Strongly Disagree*, *Disagree*, *Neutral*, *Agree*, *Strongly Agree*. The statements are marked as $SX-A$ or $SX-F$ depending on the fact that a positive feedback is respectively given to the proposed solution (*ADAPTIVE*) or the reference navigation strategy (*FIXED*). The twelve statements of the questionnaire are shown in Tab. 2. The type and order of the statements are designed and presented to the user with the same rationale of the previous test (see Sec. 7.3.1), including the strategy to avoid the agreement bias. To simplify result presentation, questions are presented here in their canonical form.

Participants. The group of participants consists in 33 users (21.2% female and 78.8% male) recruited among students, families and friends of researchers working at our center. All subjects had normal or corrected to normal vision and, as now extremely common, had basic computer or smartphone literacy. The 6.1% are high school graduates, 24.2% with a Bachelor's degree, 33.3% with a Graduate or professional degree, 33.3% have a PhD, and 3.0% prefer not to answer. About 94% have a STEM background, while 3% of them come from the Humanities field. They are researchers (54.5%), students (18.2%), teachers/professors (6.1%), developers (12.1%), and others (9.1%), which include home workers, designers, and administrators. The age are ranging from less than 18 (3.0%), 18-25 (9.1%), 26-35 (24.2%), 36-50 (48.5%), and 51-64 (15.2%). We also have an heterogeneous set of people in terms of familiarity with museums/exhibitions and virtual museum presentations. About 80% of them have visited a museum last year, and 9.1% of them have no familiarity with virtual museum presentations; for 15.2% of them, this is the first time they try an interactive setup. Finally, only 6.1% of them did not have any knowledge of the Mont'e Prama collection presented in the test.

User evaluation. As for the previous test, we evaluate the *Interaction* test from three points of view, i.e., graphically, by a scoring system, and by computing the Cronbach's alpha reliability of the questionnaire. First, we plot the histogram of responses for each statement (Fig. 15). The responses are color mapped from left (dark red, *Strongly Disagree*) to right (dark blue, *Strongly Agree*). It is clear how the majority of $SX-A$ statements are more towards the *Agree* and *Strongly Agree* part, while most of the $SX-F$ statements express a disagreement form the user. While responses are generally consistent, we discovered that some questions are not clear to some of the users. $S5-F$ asks the users if the *ADAPTIVE* exploration leads farther away from the region they want to explore than the *FIXED* path navigation; *ADAPTIVE* exploration typically remains close to the position of the user, while *FIXED* mode will continue to the next pre-defined annotation, completely ignoring user will. In fact, when the statement expresses the same concept, but in a different way, such as the statement $S8-F$, some users recognize that the *FIXED* exploration does not follow the participants exploration intention better than the *ADAPTIVE* exploration. Although users gained more knowledge from the *ADAPTIVE* exploration ($S11-A$), and found the proposed solution more satisfactory ($S12-F$,$S6-A$) and engaging ($S1-A$). Nonetheless they found the *ADAPTIVE* slightly more distracting than the *FIXED* one ($S9-A$). The effect is very small, and it is not evident to judge the reason as no specific comments were made. Our hypothesis is that, especially for naive users, the reduced set of possibilities offered by the *FIXED* exploration requires less learning and mental effort to manage navigation decisions, reducing the mental mode switches from fully guided to interactive. In any case, the unambiguous final statement demonstrates that the users prefer more the *ADAPTIVE* modalities than the *FIXED* one. In particular, 63.6% of the participants prefer the *ADAPTIVE* navigation, 9.1% does not have a preference, while 27.3% prefer the *FIXED* alternative. We assign a numerical score to both each single statement and a global score to the entire test similarly to the previous test. As done in the *User evaluation* of Sec. 7.3.1, we convert qualitative responses to numerical values, thus obtaining the final scores per statement in Fig. 16. Even with the presence of the inconsistently interpreted statements, most of the statements marked as $SX-A$ obtain a positive score, while most of those that judge positively the *FIXED* exploration (marked as $SX-F$), received a negative score. This, again, confirms the strong preference towards the *ADAPTIVE* solution. In order to compute the final global score, we sum all the single statement scores, by multiplying by -1 those marked as $SX-F$, and we remap between 0 and 1. A positive global score means that the users prefer our proposed automatic exploration system, otherwise they prefer the other one. The final global score is 0.58, showing a large preference. We found that the reliability of the questionnaire is very high, with a Cronbach's alpha equal to 0.91 for all questions, and 0.87 after the removal of redundant statements, i.e.,

2, 5, 10, and 12.

Usage statistics. The different perception of the two modalities is also reflected in a different usage pattern, despite the very similar control interface. On average, users spend considerably more time interacting using the *ADAPTIVE* solution. On average, 41.6s (median 40.5, min 0, max 108) are spent actively moving the lens, against the 29.2 for the *FIXED* solution (median 28, min 0, max 77.3). Interestingly, there has been a user that in both cases remained completely passive, just listening to the story without ever attempting to move the lens. The interactive exploration in the *ADAPTIVE* solution also leads to a slightly larger number of annotations visited. On average, 16.6 annotations (median 17, min 9. max 27) are presented (with overlay and audio explanation) against the 15.2 (median 14, min 8, max 51) for the *FIXED* version. The higher number of annotations is due to the dynamic activation of new annotations when the users explore new areas. Again, here, it is interesting to note the singular behavior of a user (one of the two that never attempted to move the lens) that continuously skipped to the next annotation at maximum speed (reaching the max of 51 annotations displayed and played in 3 minutes).

Free comments. After the experiments, we collected in the web forms a series of comments about both the *FIXED* and *ADAPTIVE* explorations. Several users explicitly stated that the *FIXED* modality is a little confusing, since they "have no control over the system. The system just explains things and the only thing the user can do is skip the explanations". They find it annoying that, when they move around, they can only explore the model without having any explanation of what they are looking at, so they find it of little use to let the user move around interrupting the guided tour. Similarly, other users complain about the fact that they don't find too intuitive how to guide the interaction; they could easily find regions of interest in the three statues, but couldn't find a way to visualize their details by themselves without waiting for the auto-tour to show them (if the *Autotour* decides to show them). Moreover, when they focus on a detail they continue to hear another audio explanation from the pre-defined series of visual/audio annotation. From this perspective they much preferred the *ADAPTIVE* configuration, finding it pretty nice and flexible. They can enjoy jumping from an annotation to another without waiting for the predefined path tour. So the *ADAPTIVE* exploration allows them to inspect much more details. We can also conclude that the *FIXED* exploration is more geared towards a mostly passive experience, with little differences than watching a video. Finally, several comments suggested possible changes in the interface implementation. For the *FIXED* modality they ask to provide more visual cues and colors. Conversely, for both modalities, they find that would be useful to speed up the annotation time (it takes too long to change the lens color), to allow users to move not only the lens but also the background scene, and to change the glyph for the `Done` button (they think that an X isn't the perfect button for this action). While in preparing the annotations we favored the audio and visual overlays to avoid clutter, a part of the users suggested us to add some more text to the annotated regions, since they think it could make the CH content easier to understand. Concerning the audio annotation, users suggest to fade that out smoothly when changing the annotation, rather than stopping it abruptly. Moreover, one user suggested to include a list of available annotations displayed somewhere in the screen (e.g. a thumbnail bar) to complement the current presentation display. Most of these suggestions point to aspects orthogonal to this work, and might be integrated in future versions of the system.

8. Conclusions and Future Work

We have proposed a framework that aims at presenting annotations in a structured way. The approach is meant to support casual users to explore, at their own pace, spatially annotated 2D models using an interactive lens that moves from an interesting area to the next, while also responding to user inputs, following shifts in interest and attention. The presentation order is dynamically dependent on lens position, navigation history and authoring information encoded in an annotation graph. The integration of a stochastic recommendation system that interprets context-dependent scores as transition probabilities makes it possible to increase the variability of exploration paths. Moreover, the user can freely mix personal/free exploration with automatic touring.

Our very preliminary evaluation has shown the potential interest of the approach, but also highlighted areas for future research. First of all, the current approach is targeted towards the exploration of areas that fit well on a circular lens, but should be refined when pointing at areas where linear or extended features should be explored. We plan to address this problem by storing at each node not only a single lens position, but a lens path for the exploration of the annotated area. Second, the dependencies presented here currently target the definition of simple precedence relations expressed by taking the fuzzy AND of values coming from enabling nodes. It is worth exploring whether fully supporting other logical operators (i.e., at least OR, XOR, and NOT) would be beneficial for improving the authoring expressiveness. Edges, in addition, might also benefit from being augmented with audio information, which could be played when a particular transition is activating, extending the current experience that limits audio clips to individual annotations. This latter feature, while interesting, is feasible within the current system for fully automatic transitions that move from one node to the next, but might require special care to be integrated with free-form lens motion.

The proposed annotation graph, state machine, and navigation interface have been applied in this work to interactions on an image plane. Such a 2D interactive exploration is natural for 2D objects, and is often applied also to fixed views of general 3D objects. The relightable 2.5D dataset used in this work is a typical example. A particularly interesting extension would be to apply this work to full 3D models using a less constrained interface. While, from the annotation point of view, our proposed concepts should already support a direct 3D extension, the interactive control and guiding components would need to be significantly extended. First of all, interactively manipulating lenses on 3D models require special care. Several solutions have been

proposed to control lenses in screen-space (e.g., [39, 40, 41] or object-space (e.g., [42, 43]), but none of these techniques seamlessly supports navigation on multiple models with coupled lens and camera control. How to control a lens while keeping an effective focus and context situation is an open problem in 3D. In terms of guidance, moreover, the various terms used for determining the next best lens would need to be adapted to 3D, in particular taking into account 3D visibility. A starting point could be the work done by Balsa et al. [17] for camera navigation.

Moreover, authoring, orthogonal to this work, also deserves attention, in particular in case of extension of the dependency logic. Finally, our current evaluation was very preliminary, and focused mostly on responding to our the main research questions, i.e., whether the presence of dependency among annotations perceivably improves the experience, and whether users enjoy our flexible interactive or mostly interactive tours better than the more standard fixed auto-touring features. More work is required to objectively assess the effectiveness of our user interface. It will be also interesting to evaluate whether the proposed approach, currently tuned to museum applications, can be extended to more complex situations requiring specific visualization tasks to be solved.

Acknowledgments The authors thank CRBC Sassari and SABAPC Cagliari for the access to the artworks for the purpose of digitization and for annotation information. We also acknowledge the contribution of Raffaella Chierici for content creation. The project received funding from the European Union's H2020 research and innovation programme under grant 813170 (EVOCATION), and from Sardinian Regional Authorities under project VDIC (POR FESR 2014-2020).

References

[1] Economou, M, Meintani, E. Promising beginnings? evaluating museum mobile phone apps. In: Proc. Rethinking Technology in Museums Conference. 2011, p. 26–27.

[2] Kuflik, T, Wecker, A, Lanir, J, Stock, O. An integrative framework for extending the boundaries of the museum visit experience: linking the pre, during and post visit phases. Information Technology & Tourism 2014;15:17–047. doi:10.1007/s40558-014-0018-4.

[3] Jankowski, J, Hachet, M. A survey of interaction techniques for interactive 3D environments. In: Eurographics 2013-STAR. 2013, p. 65–93. doi:10.2312/conf/EG2013/stars/065-093.

[4] Vanhulst, P, Evequoz, F, Tuor, R, Lalanne, D. A descriptive attribute-based framework for annotations in data visualization. In: Proc. International Joint Conference on Computer Vision, Imaging and Computer Graphics. 2018, p. 143–166. doi:10.1007/978-3-030-26756-8_7.

[5] Ponchio, F, Callieri, M, Dellepiane, M, Scopigno, R. Effective annotations over 3D models. Computer Graphics Forum 2020;39(1):89–105. doi:10.1111/cgf.13664.

[6] Camba, J, Contero, M, Johnson, M. Management of visual clutter in annotated 3D CAD models: A comparative study. In: Proc. International Conference of Design, User Experience, and Usability. 2014, p. 405–416. doi:10.1007/978-3-319-07626-3_37.

[7] Bettio, F, Ahsan, M, Marton, F, Gobbetti, E. A novel approach for exploring annotated data with interactive lenses. Computer Graphics Forum 2021;40(3):387–398. doi:10.1111/cgf.14315.

[8] Tominski, C, Gladisch, S, Kister, U, Dachselt, R, Schumann, H. Interactive lenses for visualization: An extended survey. Computer Graphics Forum 2017;36(6):173–200. doi:10.1111/cgf.12871.

[9] Segel, E, Heer, J. Narrative visualization: Telling stories with data. IEEE TVCG 2010;16(6):1139–1148. doi:10.1109/TVCG.2010.179.

[10] Healey, CG, Dennis, BM. Interest driven navigation in visualization. IEEE TVCG 2012;18(10):1744–1756. doi:10.1109/TVCG.2012.23.

[11] Ahsan, M, Marton, F, Pintus, R, Gobbetti, E. Guiding lens-based exploration using annotation graphs. In: Proc. Smart Tools and Applications in Graphics (STAG). 2021, p. 85–90. doi:10.2312/stag.20211477.

[12] Besançon, L, Ynnerman, A, Keefe, DF, Yu, L, Isenberg, T. The state of the art of spatial interfaces for 3D visualization. Computer Graphics Forum 2021;40(1):293–326. doi:10.1111/cgf.14189.

[13] Faraday, P, Sutcliffe, A. Designing effective multimedia presentations. In: Proc. SIGCHI. 1997, p. 272–278. doi:10.1145/258549.258753.

[14] Wang, Z, Shi, W, Akoglu, K, Kotoula, E, Yang, Y, Rushmeier, H. CHER-Ob: A tool for shared analysis and video dissemination. J Comput Cult Herit 2018;11(4). doi:10.1145/3230673.

[15] Potenziani, M, Callieri, M, Dellepiane, M, Corsini, M, Ponchio, F, Scopigno, R. 3DHOP: 3D heritage online presenter. Computers & Graphics 2015;52:129–141. doi:10.1016/j.cag.2015.07.001.

[16] Ellis, G, Dix, A. A taxonomy of clutter reduction for information visualisation. IEEE TVCG 2007;13(6):1216–1223. doi:10.1109/TVCG.2007.70535.

[17] Balsa Rodriguez, M, Agus, M, Marton, F, Gobbetti, E. Adaptive recommendations for enhanced non-linear exploration of annotated 3D objects. Computer Graphics Forum 2015;34(3):41–50. doi:10.1111/cgf.12616.

[18] Cockburn, A, Karlson, A, Bederson, BB. A review of overview+detail, zooming, and focus+ context interfaces. ACM Computing Surveys (CSUR) 2009;41(1):1–31. doi:10.1145/1456650.1456652.

[19] Ellis, G, Bertini, E, Dix, A. The sampling lens: making sense of saturated visualisations. In: CHI'05 extended abstracts on Human Factors in Computing Systems. 2005, p. 1351–1354. doi:10.1145/1056808.1056914.

[20] Jaspe-Villanueva, A, Ahsan, M, Pintus, R, Giachetti, A, Marton, F, Gobbetti, E. Web-based exploration of annotated multi-layered relightable image models. ACM Journal on Computing and Cultural Heritage (JOCCH) 2021;14(2):1–29. doi:10.1145/3430846.

[21] Furnas, GW. Generalized fisheye views. ACM SIGCHI Bulletin 1986;17(4):16–23. doi:10.1145/22627.22342.

[22] Van Ham, F, Perer, A. "search, show context, expand on demand": Supporting large graph exploration with degree-of-interest. IEEE TVCG 2009;15(6):953–960. doi:10.1109/TVCG.2009.108.

[23] Gladisch, S, Schumann, H, Tominski, C. Navigation recommendations for exploring hierarchical graphs. In: International Symposium on Visual Computing. 2013, p. 36–47. doi:10.1007/978-3-642-41939-3_4.

[24] Aoki, PM, Grinter, RE, Hurst, A, Szymanski, MH, Thornton, JD, Woodruff, A. Sotto voce: Exploring the interplay of conversation and mobile audio spaces. In: Proc. SIGCHI. 2002, p. 431–438. doi:10.1145/503376.503454.

[25] Hutchinson, R, Eardley, AF. Inclusive museum audio guides: 'guided looking' through audio description enhances memorability of artworks for sighted audiences. Museum Management and Curatorship 2021;36(4):427–446.

[26] Bekos, MA, Niedermann, B, Nöllenburg, M. External labeling techniques: A taxonomy and survey. Computer Graphics Forum 2019;38(3):833–860.

[27] Shneiderman, B. The eyes have it: a task by data type taxonomy for information visualizations. In: Proc. IEEE Symposium on Visual Languages. 1996, p. 336–343. doi:10.1109/VL.1996.545307.

[28] Floyd, RW. Algorithm 97: Shortest path. Commun ACM 1962;5(6):345–349. doi:10.1145/367766.368168.

[29] Kristensen, JW, Schjørring, A, Mikkelsen, A, Johansen, DA, Knoche, HO. Of leaders and directors: A visual model to describe and analyse persistent visual cues directing to single out-of view targets. In: Proc. VRST. 2021, p. 88:1–88:3. doi:10.1145/3489849.3489953.

[30] Bettio, F, Jaspe Villanueva, A, Merella, E, Marton, F, Gobbetti, E, Pintus, R. Mont'e Scan: Effective shape and color digitization of cluttered 3D artworks. ACM Journal on Computing and Cultural Heritage (JOCCH) 2015;8(1):4:1–4:23. doi:10.1145/2644823.

[31] Balsa Rodriguez, M, Agus, M, Bettio, F, Marton, F, Gobbetti, E. Digital Mont'e Prama: Exploring large collections of detailed 3D models of sculptures. ACM Journal on Computing and Cultural Heritage (JOCCH) 2016;9(4):1–23. doi:10.1145/2915919.

[32] Boninu, A, Usai, L, Costanzi Cobau, A, Minoja, M, Usai, A, editors. Le sculture di Mont'e Prama – Conservazione e restauro – La Mostra – Contesto, scavi e materiali. Gangemi; 2015. ISBN 9788849229844.

[33] Scholtz, J. Beyond usability: Evaluation aspects of visual analytic envi-

ronments. In: Proc. IEEE VAST. 2006, p. 145–150. doi:10.1109/VAST.2006.261416.

[34] Likert, R. A technique for the measurement of attitudes. Archives of psychology 1932;22(140):55.

[35] Othman, MK. Measuring visitors' experiences with mobile guide technology in cultural spaces. Ph.D. thesis; University of York Toronto, Canada; 2012.

[36] Hajjar, S. Statistical analysis: Internal-consistency reliability and construct validity. International Journal of Quantitative and Qualitative Research Methods 2018;6(1):46–57.

[37] Olhager, J, Selldin, E. Manufacturing planning and control approaches: market alignment and performance. International Journal of Production Research 2007;45(6):1469–1484.

[38] Artmyn, . Artmyn - new generation technological tools and services for the art ecosystem. 2021. URL: https://artmyn.com/; [Online; accessed 15-December-2021].

[39] Gasteiger, R, Neugebauer, M, Beuing, O, Preim, B. The FLOWLENS: A focus-and-context visualization approach for exploration of blood flow in cerebral aneurysms. IEEE Transactions on Visualization and Computer Graphics 2011;17(12):2183–2192.

[40] Pindat, C, Pietriga, E, Chapuis, O, Puech, C. JellyLens: Content-aware adaptive lenses. In: Proc. UIST. 2012, p. 261–270.

[41] Kluge, S, Gladisch, S, Freiherr von Lukas, U, Staadt, O, Tominski, C. Virtual lenses as embodied tools for immersive analytics. In: GI VR / AR Workshop. 2020, p. 1–12.

[42] Rocha, A, Silva, JD, Alim, UR, Carpendale, S, Sousa, MC. Decallenses: Interactive lenses on surfaces for multivariate visualization. IEEE transactions on visualization and computer graphics 2018;25(8):2568–2582.

[43] Mota, RCR, Rocha, A, Silva, JD, Alim, U, Sharlin, E. 3De interactive lenses for visualization in virtual environments. In: Proc. IEEE Scientific Visualization Conference (SciVis). 2018, p. 21–25.