IsoCam: Interactive Visual Exploration of Massive Cultural Heritage Models on Large Projection Setups

FABIO MARTON, MARCOS BALSA RODRIGUEZ, FABIO BETTIO, MARCO AGUS, ALBERTO JASPE VILLANUEVA, and ENRICO GOBBETTI, CRS4 Visual Computing, Italy

We introduce a novel user interface and system for exploring extremely detailed 3D models in a museum setting. Three-dimensional models and associated information are presented on a large projection surface controlled by a touch-enabled surface placed at a suitable distance in front of it. Our indirect user interface, dubbed *IsoCam*, combines an object-aware interactive camera controller with an interactive point-of-interest selector and is implemented within a scalable implementation based on multiresolution structures shared between the rendering and user interaction subsystems. The collision-free camera controller automatically supports the smooth transition from orbiting to proximal navigation, by exploiting a distance-field representation of the 3D object. The point-of-interest selector exploits a specialized view similarity computation to propose a few nearby easily reachable interesting 3D views from a large database, move the camera to the user-selected point of interest, and provide extra information through overlaid annotations of the target view. The capabilities of our approach have been demonstrated in a public event attended by thousands of people, which were offered the possibility to explore submillimetric reconstructions of 38 stone statues of the Mont'e Prama Nuragic complex, depicting larger-than-life human figures, and small models of prehistoric Nuraghe (cone-shaped stone towers). A follow-up of this work, using 2.5m-high projection screens, is now included in permanent exhibitions at two Archeological Museums. Results of a thorough user evaluation, involving quantitative and subjective measurements, are discussed.

Categories and Subject Descriptors: I.3.6 [**Computer Graphics**]: Methodology and Techniques—*Interaction techniques*; H.5.2 [**Information Interfaces and Presentation (HCI)**]: User Interfaces—*Interaction styles, Input devices and strategies*; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism

General Terms: Cultural Heritage

Additional Key Words and Phrases: Interaction techniques, camera controls, 3D navigation, 3D viewers, 3D visualization

ACM Reference Format:

Fabio Marton, Marcos Balsa Rodriguez, Fabio Bettio, Marco Agus, Alberto Jaspe Villanueva, and Enrico Gobbetti. 2014. IsoCam: Interactive visual exploration of massive cultural heritage models on large projection setups. ACM J. Comput. Cult. Herit. 7, 2, Article 12 (June 2014), 24 pages.
DOI: http://dx.doi.org/10.1145/2611519

1. INTRODUCTION

Three-dimensional shape and material acquisition, as well as modeling techniques, are now able to produce highly detailed and accurate 3D representations of cultural heritage artifacts. While this

This work is partially supported by the EU FP7 Program under the DIVA project (290277) and by *Soprintendenza per i Beni Archeologici per le Province di Cagliari ed Oristano (ArcheoCAOR)*.
Authors' addresses: F. Marton, M. Balsa Rodriguez, F. Bettio, M. Agus, A. Jaspe Villanueva, and E. Gobbetti, CRS4, POLARIS, Ed. 1, 09010 Pula (CA), Italy; email: {marton, mbalsa, fabio, magus, ajaspe, gobbetti}@crs4.it.

DOI: http://dx.doi.org/10.1145/2611519

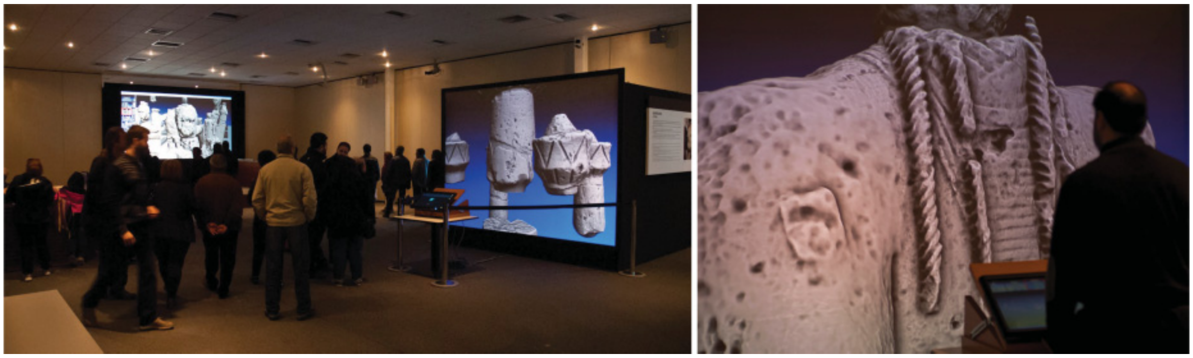


Fig. 1. Museum exhibition. The method described in this work has been employed for the virtual stands in a large exhibition attended by approximately 3,500 people who had the possibility to freely inspect 38 highly detailed models. A larger setup is now installed in two permanent exhibitions at the archeological museums in Cagliari and Cabras (Italy).

digitization process has a variety of applications, including archival, study, and restoration, visual communication is by large the most common utilization. Until recently, the most successful and widespread use of 3D reconstructions for exploration have been through mostly passive visual presentation modalities, such as videos or computer-generated animations. Interest is, however, now shifting toward more flexible active presentation modalities, such as virtual navigation systems, which let users directly drive navigation and inspection of 3D digital artifacts. These active presentation approaches are known to engage museum visitors and enhance the overall visit experience, which tends to be personal, self-motivated, self-paced, and exploratory [Falk and Dierking 2000]. In general, visitors do not want to be overloaded with instructional material, but they do want to receive the relevant information, learn, and have an overall interesting experience. To serve this goal, user-friendly and flexible systems are needed, and many challenges need to be addressed in parallel [Kuflik et al. 2011]. Our work deals with the particular problem of inspecting extremely detailed visual representations of 3D models, such as statues reconstructed using state-of-the-art 3D scanning pipelines. This article principally focuses on a novel user interface integrated in a museum presentation system designed in collaboration with domain experts in the context of a large cultural heritage visualization project (see Section 3).

Our approach. Three-dimensional models and associated information are presented on a large projection surface (projection wall). We let users focus their attention exclusively on the large screen, while controlling the application through a touch-enabled surface placed at a suitable distance in front of it (see Figure 1, right). In this setting, the display of the touch-enabled screen is used only to provide minimal help information on the application. We exploit this simple setup with an indirect user interface, dubbed *IsoCam*, which combines an object-aware interactive camera controller, to incrementally explore the 3D model, with an interactive point-of-interest selector. Our camera controller exploits a multiresolution distance-field representation of the 3D object, which allows it to be scalable and to smoothly transition from orbiting to proximal navigation (see Section 4). The simple point-of-interest selector exploits a specialized view-similarity computation to propose a few nearby easily reachable interesting 3D views from a large database, move the camera to the user-selected point of interest, and provide extra information through overlaid annotations of the target view (see Section 5). A scalable implementation is realized on top of specialized multiresolution structures shared between rendering and user interaction subsystems.

Novelty. Our system combines and extends a number of state-of-the-art results. The main novel contribution is an intuitive 3D camera controller, handling objects with multiple components while smoothly transitioning from orbiting to proximal navigation, integrated with a simple user interface for point-of-interest selection and overlaid information display. Thanks to view-adaptive structures, object-aware user-interface components are capable to support models composed of various billions of primitives and large numbers of points of interest, and they have been used by thousands of inexperienced users in a real exhibition with a massive object collection (see Section 8).

Advantages. The proposed setup is simple, low in cost, and suitable for exhibitions and museums; it supports large-scale display surfaces where multiple users can focus their attention and objects can be displayed at an imposing scale. The proposed navigation technique, which can also be employed in other more standard settings, lets inexperienced users inspect topologically complex 3D shapes at various scales and without collisions, integrating panning, rotating, and zooming controls into simple low-degree-of-freedom operations. Our approach, based on distance-field isosurfaces, significantly improves over previous constrained orbiting and proximal navigation methods [Khan et al. 2005; McCrae et al. 2009; Moerman et al. 2012; Marton et al. 2012] by better supporting disconnected components, more seamlessly transitioning between orbiting and proximal navigation, and providing smoother camera motion, while being simpler to implement, as it is based on a single unifying concept and does not require complex algorithms (e.g., motion prediction) to handle special cases. On-demand context-based selection of "nearby" points of interests is well integrated with the IsoCam interface, reduces clutter, does not require a pointing device for hot-spot selection, and supports overlaid information associated to target views. Our scalable implementation supports giga-triangle–sized models and hundreds of points-of-interest on commodity platforms.

Limitations. As for all current approaches, our approach has also limitations. The presented approach targets the traditional nonimmersive setting with 2D screens and low-DOF input devices, and might be not suitable for more immersive setups (e.g., CAVEs). In addition, we only target the problem of 3D model exploration with image/text overlays. Using other associated multimedia information and/or supporting narratives are orthogonal problems not treated in this work. The proposed navigation technique achieves simplicity by constraining camera/model motion. As for many other constrained navigation techniques, not all view positions are thus achievable.

2. RELATED WORK

Creating an interactive projection-based system for natural exploration of massive models in museums requires combining and extending state-of-the-art results in a number of technological areas. In this section we briefly discuss the methods that most closely relate to ours. For a wider coverage, we refer the reader to well-established surveys on 3D interaction techniques [Jankowski and Hachet 2013], 3D camera control [Christie and Olivier 2009], massive model rendering [Dietrich et al. 2007; Gobbetti et al. 2008; Yoon et al. 2008], and visualization strategies for cultural heritage [Foni et al. 2010].

Motion control for natural exploration of massive surface models. In the context of visualization of massive complicated scenes, users require interactive control to effectively explore the data. Most of the work in this area is connected to camera/object motion control [Christie and Olivier 2009; Jankowski and Hachet 2013]. In this article, we deal with exocentric camera motion techniques for object inspection. Mapping the limited DOF input to a 6DOF transformation space, and assisting the user to generate "good" motions are two of the main problems. Variations of the virtual trackball [Chen et al. 1988; Shoemake 1992; Henriksen et al. 2004], which decompose motion into pan, zoom, and orbit, are the most commonly employed approaches. To solve the *lost-in-space problem* and avoid collisions with the environment, Fitzmaurice et al. [2008] have proposed the Safe Navigation Technique, which, however, requires explicit positioning of rotation pivot, and thus needs precise pointing. Moreover, motion decomposition and pivot positioning can be difficult for novice users. For this reason, a number of authors have proposed techniques for effective object inspection that constrain viewpoint by using precomputed/authored camera properties [Hanson and Wernert 1997; Burtnyk et al. 2002, 2006], increasing authoring time and limiting effective view selection. Proximal surface navigation methods constrain the camera to stay in a region around the object and with a specific orientation with respect to the visible surface [Khan et al. 2005; McCrae et al. 2009; Moerman et al. 2012; Marton et al. 2012]. These methods, following a hovercraft metaphor, slide a target point on the surface and compute camera positions so as to look at this point from a constant distance and at a good orientation. Surface/depth smoothing and algorithmic solutions are used to reduce jerkiness and deal with singularities and out-of-view field collisions. Moreover, disconnected surfaces are difficult to handle with this method. Our approach solves these problems by exploiting the structure of the distance field and by computing camera parameters by decoupling view position and view direction computation. This solution significantly reduces jerkiness, guarantees a collision-free motion, handles topologically complex surfaces, and smoothly blends from hovering to orbiting as distance to the surface increases.

Multitouch interfaces for 3D model exploration. In recent years, researchers started to combine multipoint input technology with interactive 3D graphics. Hancock et al. [2009, 2007] proposed different approaches using direct and indirect multitouch interaction techniques for tabletop surfaces to manipulate 2D and 3D content. Martinet et al. [2012] have studied the effect of DOF separation in a 3D manipulation task on a direct multitouch display. In their work, Reisman et al. [2009] proposed co-location constraints for direct-touch object manipulation. The Cubtile has been designed by de la Riviere et al. [2008], where users interact with 3D content from an indirect multitouch box-shaped device. Recently, direct touch interfaces have been created for interacting with visualizations in application domains ranging from astrophysics [Fu et al. 2010] to oceanography [Butkiewicz and Ware 2011], fluid mechanics [Klein et al. 2012], and medicine [Lundstrom et al. 2011]. In particular, Yu et al. [2010] proposed a general direct-touch technique allowing users to explore 3D data representations and visualization spaces. All these systems are targeted to scientific visualization of complex data, while our system is focused on cultural heritage, and designed to be effective in a museum setting, mostly for naïve users. Recent work on multitouch surfaces has also focused on indirect interaction, which is not guided by colocation concerns, and seems particularly interesting for interactive 3D visualization [Moscovich and Hughes 2008]. In particular, Knoedel and Hachet [2011] showed that direct-touch shortens completion times, but indirect interaction improves efficiency and precision, and this is particularly true for 3D visualizations. In our system, we considered an indirect gestural multitouch approach and we coupled it with an assisted user interface to reduce the number of DOFs and gestures. Our method is proved to be particularly effective for users without experience in 3D interaction techniques.

Dual display setups. Recently, a number of visualization systems have been presented exploiting two display surfaces. In general, these systems employ an horizontal interactive touch table, and a large display wall visualizing the data, sometimes combining them in a continuous curved display [Weiss et al. 2010; Wimmer et al. 2010]. Our system layout is similar to the two-display volumetric exploration system of Coffey et al. [2012], which, however, employs a World-In-Miniature metaphor, simultaneously displaying a large-scale detailed data visualization and an interactive miniature. Differently from all these systems, we use the horizontal surface exclusively for indirect interaction so that users keep their focus on virtual exploration on the main display.

Integration of additional information. The use of auxiliary information to integrate and enhance the presentation of complex data has a long history [Faraday and Sutcliffe 1997], and 3D environments where textual information is attached to the models in a spatially coherent way have been thoroughly studied [Bowman et al. 2003; Sonnet et al. 2005; Jankowski et al. 2010; Polys et al. 2011]. Recently, a number of authors have proposed techniques for presenting 3D models and textual information in an integrated fashion [Götzelmann et al. 2007; Jankowski and Hachet 2013; Callieri et al. 2013]. In this work, we only target the problem of 3D model exploration with image/text overlays, and associate overlays to points of interest selection in order to overcome the limitations of the classic hot-spot solution, which require pointing devices.

Point-of-interest exploration. Thumbnail-bars are tools that enable the navigation of a dataset by employing a set of precomputed images. At any given time, one image of the dataset can be selected by the user as current focus. The focus image can be regarded as the current position where the user sits within the dataset. The exact meaning of selecting a focus depends on the application: In an image viewer, the focus can be the image being shown at full resolution in the main area of the screen; in a 3D navigation application [Pintore et al. 2012], where images are linked to viewpoints (the physical camera positions of the shot), the focus image can drive the setup of the virtual camera used to render the 3D scene inside the main screen area. Often, these images are also linked to additional information, which is displayed when the user reaches them, as an alternative to the usage of hot-spots [Andujar et al. 2012; Besora et al. 2008]. The organization of the images in this kind of tools can be considered a challenging problem in many CH scenarios because simple grid layout approaches do not scale up well enough with the number of images. A trend consists in clustering images hierarchically, according to some kind of image semantic, like combining time and space [Ryu et al. 2010], spatial image-distances [Epshtein et al. 2007; Jang et al. 2009], or a mixture of them [Mota et al. 2008] to automatically, or interactively [Girgensohn et al. 2009; Crampes et al. 2009] compute image-clusters. Most of these works strive to identify good clustering for images, rather than good way to dynamically present and explore the clustered dataset. Our approach instead is navigation-oriented and it is organized in a way that, in any moment, users can decide to change the point of view and trigger display of overlaid information by browsing a limited number of thumbnails radially organized and automatically chosen according to the similarity to the current view point. Similar concepts can be found in the hyperlink system of Goetzelmann et al. [2007], which presented a system to link textual information and 3D models, where links on the text are associated to predefined points of view, and definition of a point of view activates a mechanism proposing contextual information.

3. OVERVIEW

The design of our method has taken into account requirements gathered from domain experts. Section 3.1 briefly illustrate the design process and the derived requirements, while Section 3.2 provides a general overview of our approach, justifying the design decisions in relation to the requirements.

3.1 Requirements

While our methods are of general use, our work is motivated by the *Digital Mont'e Prama* project, a collaborative effort between our center and the *Soprintendenza per i Beni Archeologici per le Province di Cagliari ed Oristano (ArcheoCAOR)*, which aims to present to the public a large unique collection of prehistoric statues of the Mont'e Prama complex, including larger-than-life human figures and small models of prehistoric nuraghe (cone-shaped stone towers) (see Figure 1). The project covers aspects ranging from 3D digitization to exploration.

In order to design our setup and interactive technique, we started with a detailed analysis of the requirements of our chosen application domain, involving domain experts in a participatory design process. The experts include one museum curator, two archaeologists, and one museum architect from ArcheoCAOR, plus two restorations experts from the *Centro di conservazione e restauro dei beni culturali* of Li Punti. The design process took approximately four months, for a total of 16 meetings.

From interactions with the domain experts, we derived a list of requirements to guide our development. Additional requirements stem from our analysis of related work (see Section 2) and our own past experience developing interactive systems for cultural heritage [Bettio et al. 2009; Marton et al. 2012; Balsa Rodriguez et al. 2013]. We now describe the main requirements that we used for guiding the development process, briefly summarizing how they were derived:

R1. Large-scale visualization (real-world-sized macro-structure). The specific application case for our design is the presentation of larger-than-life human statues, ranging from 2.0m to 2.5m of height. The statues were constructed at this imposing scale on purpose, and this macrostructure information should be immediately conveyed to the visitor through a real-scale (or larger-than-real) presentation. This means, in particular, that we need to support large (wall-sized) displays.

R2. High-resolution details (magnified microstructure). As almost any decorated Cultural Heritage object, the statues present information at multiple scales (global shape and carvings). Even the finest material microstructure carries valuable information (e.g., on the carving process). For instance, the Mont'e Prama statues have millimeter-sized carvings, and thus require submillimetric model precision. This carving information should be clearly perceivable at all scales, and should be magnified for close inspection.

R3. Seamless interactive exploration and zooming (macro- and microstructure). Comprehension of the artifacts impose the capability to seamlessly move from macrostructure analysis, providing information on function and context, and microstructure analysis, providing information on nature, shape and signification of decorations. Camera navigation should thus support both global object motion and proximal navigation (panning over a surface to inspect details). The control modes should be active with real-time feedback in order to engage visitors providing them the sense of control, and they support smooth and seamless object inspection, going back and forth from shape inspection to detail inspection.

R4. Annotation display. In the case of presentation of 3D artworks, the user interface should support 3D model exploration connected to accompanying information (e.g., rich textual descriptions). This information should not be obtrusive and directly linked to the 3D model (e.g., to provide hints on reconstruction hypotheses for missing parts).

R5. Fast learning curve and assisted navigation. In a museum installation, where *walk-up-and-use* interfaces are expected, the visitors' experience could be easily frustrated if the proposed interaction paradigm does not allow them to immediately explore the content, through a natural user interface with an extremely short learning curve. Moreover, because museums must manage large amounts of visitors, long training times and/or guided training are not affordable. The user interface should thus be perceived as simple, immediately usable ("please touch"), and provide guidance in complex operation (e.g., to avoid lost-in-space situations during camera control), while not being perceived as overly obtrusive.

R6. Single-user control, multiuser fruition. Visitor experience tends to be personal, self-motivated, self-paced, and exploratory. At the same time, the support of multiuser viewing enables visitors to benefit from other people's experiences in terms of user interaction behavior and object exploration, also encouraging collaboration and discussions. While a single-user interface is considered appropriate, the physical setup should comfortably support at least 5 to 10 observers.

R7. Focus on work of art (avoid occlusion from people and/or interaction widgets). The important information is the visualized object itself, which, as in a real exhibition should not be obstructed by other people or general clutter (e.g., interaction widgets). Visitor focus should thus be guided to the presentation medium.

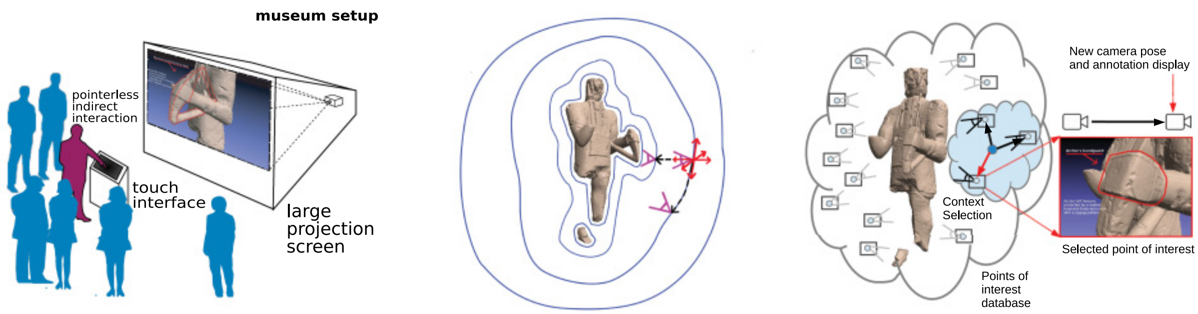
R8. Flexibility and low cost (possibility of multiple installations). For practical purposes and to ensure wide adoption, the system should be flexibly configurable and adjustable, of reasonably low cost and easy to maintain by nontechnical staff (e.g., simple automated switch on/off).

3.2 Approach

Requirements **R1** through **R8, as well as our analysis of related work presented in Section 2**, were taken as guidelines for our development process, which resulted in the definition of an approach based on indirect touch-based interactive control of a large projection surface through an assisted user-interface. The main components of our user-interface are an object-based assisted system for interactive camera motion tuned for both orbiting and proximal object inspection, and a context-based point-of-interest selector for moving the camera to precomputed viewpoints and display associated information in overlay. Multiresolution technology is used both in the user-interface and the rendering subsystem to cope with performance constraints.

Proposed setup. The widespread diffusion of touch devices, such as tablets or smartphones, has made people used to touch-based user interfaces based on direct manipulation of on-screen objects (**R5**). This kind of setting is increasingly being used at a larger scale, also for museum presentations, with users exploring models on touch-enabled monitors, tables, or walls [Hachet et al. 2013]. Instead of using a large touch screen as input/output device for direct manipulation, we have preferred to decouple the devices for interaction and for rendering as to allow for large projection surfaces and enable multiple users to watch the whole screen without occlusion problems (**R1**, **R6**) and staying at a suitable distance from it when viewing large objects of imposing scale (**R1**). By contrast, using co-location of touch interaction and display would force users to stay within reach of the display, limiting its size. Many embodiments of this concept are possible. In this work, we present a low-cost solution (**R8**) based on a single PC connected to a small touch screen for input and display of help information and a projector illuminating a large screen for application display (see Figure 2(a)). We emphasize that the touch screen does not display any visual feedback, encouraging the users to focus solely on the large display (**R7**) where the objects of interest are rendered (see Figure 8(a)). Both single- and multitouch controls have been implemented (see Section 6).

Camera control. Three-dimensional variations of the well-known 2D multitouch RST technique, that allows the simultaneous control of Rotations, Scaling, and Translations from two finger inputs are becoming increasingly used as viewpoint manipulation technique in this context [Kurtenbach et al. 1997; Jankowski and Hachet 2013]. While no real standard for 3D touch-based interaction exists [Keefe and Isenberg 2013], touch surfaces are now so common that people are encouraged to immediately start interacting with them, which is an important aspect of *walk-up-and-use* interfaces. Moreover, even if the mapping between 2D and 3D motions is nontrivial and varies for a user interface to the next, users are encouraged to learn by trial and error while interacting. Inspection of complex objects, especially for novice users, can be difficult, as, similarly to virtual trackball approaches, users are continuously shifting between pan, orbit, and zoom operations, and, in the case of topologically complex objects presenting many cavities and protrusions, the camera eye can cross the object surface, resulting in



(a) Setup (b) Assisted camera control (c) Points of interest

Fig. 2. Method overview. Users focus on a large projection screen. Object exploration is achieved by moving the camera on isosurfaces of the object's distance field, and point-of-interest selection is context sensitive and associated to display of auxiliary information.

an undesirable effect. General 7DOF controllers also require special handling to avoid lost-in-space effects. These problems occur because the navigation interface is not context-aware, but simply operates on the model transformation, leaving the burden to adapt to the surface to the user. We have thus designed an object-aware technique explicitly targeted to constrained macroscale and microscale object exploration (**R5**). Among the possible solutions discussed in Section 2, we focus on techniques that do not required on-screen widgets, in order to have a clean view of the work of art (**R7**). IsoCam (see Section 4) automatically avoids collisions with scene objects and local environments, while trying to maintain the camera at a fixed distance around the object while maintaining a good orientation (**R3**). When hovering around the surface, our method maintains the camera position on the current isosurface of the distance field generated by the model, and, once computed the new viewpoint, devises a good orientation based on temporal continuity and surface orientation at multiple scales. Zooming instead is obtained by changing the current isolevel. With this approach, collisions are avoided, and the camera accurately follows the model shape when close to the object, while smoothly evolving to move around a spherical shape when moving away from it (see Figure 2(b)). This approach permits to browse in a smooth way the model always facing the camera in a proper direction, but also allows the user to move among disconnected parts, due to the topological structure of a distance field. User interaction does not require to provide a visual feedback or to implement picking, as needed (e.g., to select a pivot for rotation over a model). The user can update the camera position by interacting with the touch screen, while always looking at the large display where the object of interest is shown (**R7**). Among the many non-standardized mappings of 3D touch gestures to 3D manipulations [Keefe and Isenberg 2013], we choose the simple approach of mapping common 2D RST gestures, such as pinch-to-zoom, one- and two-fingers pan, and two-finger rotate to analogue 3D behaviors, as this encourages people to quickly learn the mapping by looking at the effect of their actions. We note that this is made possible because of the constrained nature of our interface, which reduces degrees of freedom in order to favor ease of use at the expense of generality.

Points of interest. Three-dimensional hot-spots, popularized by VRML/X3D browsers and geoviewers, are the most common methods for associating auxiliary information to the 3D model [Callieri et al. 2008; Jankowski and Decker 2012]. These methods, however, require pointing methods and/or produce clutter. Instead of using hot-spots, we thus employ a context-based radial menu to select among a subset of context-selected views. The system thus supplies the user, on demand, a set of precomputed points-of-interest, corresponding to the best ones with respect to the observer position and view similarity metric (see Figure 2(c) and Section 5). Each view is also enriched with an associated overlaid information, which is presented to the viewer (**R6**). As for constrained navigation, point-of-interest selection does not require any visual feedback on the interaction touch screen, nor the availability of a specific pointing method (see Section 6) (**R7**).

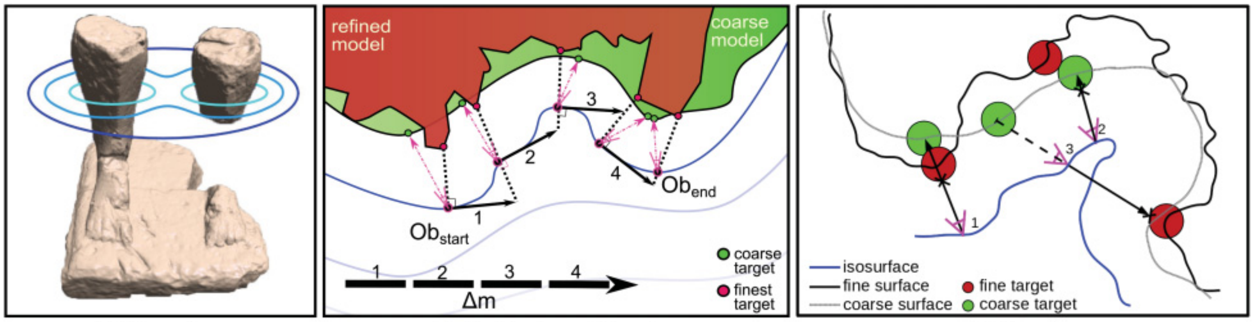
Scalability. To ensure real-time performance (**R3**) on large datasets (**R1**) presented at high resolution (**R2**) (e.g., billions of triangles and hundreds of points of interest per model), we employ specialized multiresolution structures and adaptive algorithms. In our approach, the renderer and the user-interface subsystem share structure and work on a view-adapted representation of the dataset (see Section 7). In particular, we have extended the Adaptive TetraPuzzles (ATP) approach [Cignoni et al. 2004] for massive model multiresolution visualization to implement all required geometric queries, and employ kd-trees to organize the points of interest. Since the complexity of all operations depend only on the (bounded) view complexity, rather than on the (potentially unbounded) model complexity, we can guarantee high display accuracy and real-time performance both for macro- and microstructure inspection (**R1**, **R2**, **R3**).

4. CAMERA CONTROL

Previously proposed object-based orbiting approaches, well suited for indirect interaction, are based on the concept of sliding a target point on the object surface, and determining the camera orientation based on the local surface normal, using ad-hoc techniques to manage cavities and sharp turns. These requirements have two implications: First, the user can easily move only on connected parts of the model, and second, the movement is highly sensible to the surface roughness and thus needs to be filtered properly to avoid abrupt accelerations.

We start from a concept similar to the one of the sliding surface, but the basic idea is to exploit the structure of the object's distance field. Thus, during hovering operations, the camera position is maintained at a constant isolevel, while zooming is achieved by changing the isolevel. Camera orientation is determined in this approach only after position is incrementally updated. One of the benefits is that isosurfaces of the distance field, being offset surfaces, are much smoother than the surface itself and change smoothly from the model shape to a spherical shape with increasing distance, combining increased smoothness with topological simplification. This permits to follow all the fine detail when the distance is small, while smoothly orbiting and moving across disconnected parts when the distance increases. For example, when inspecting a human figure, users can hover over a single leg when looking from a few centimeters, while orbiting around both legs from a further distance (see Figure 3(a)).

As we will see, orientation is computed by considering the gradient of a smoothed version of the distance field, an up-vector model, and temporal continuity considerations. Decoupling position and orientation computation allows us to have precise guarantees in terms of collision-free motion, which are not shared by hovering techniques that smooth the surface to reduce jittering [Moerman et al. 2012]. In particular, distance field information is extracted with on-the-fly nearest-neighbor computation on multiresolution approximations of the current view-dependent approximation of the model (see Section 7). It should be noted here that position computation must be precise enough to avoid collisions, while orientation computation can trade precision with smoothness. Thus, a precise nearest neighbor search is used to keep the user at a fixed distance from the model surface, while a coarser nearest neighbor computation is preferred to compute the view direction, toward having smooth camera orientation variations. In the following, we define *fine target* as the nearest neighbor on the refined model with respect to the eye, while the *coarse target* corresponds to the nearest neighbor on a smoother coarse model (see Figure 3(b)). Section 7 details how fine and coarse targets can be efficiently computed using the same multiresolution structure that is used for rendering.

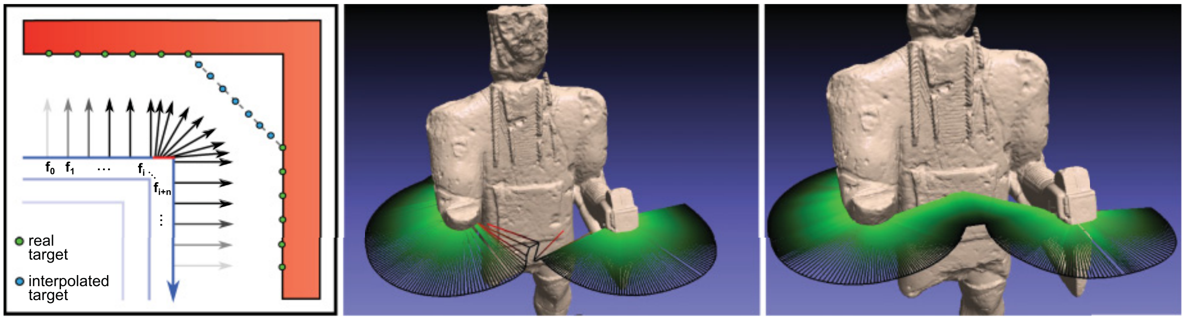


(a) Isosurface (b) Positioning and hovering (c) Zooming

Fig. 3. IsoCam. (a) Some line of the isosurface. (b) Hover movement orthogonal to gradient, user input Δm split into four steps to follow the isosurface. (c) Zooming stopped in three situations: (case 1) approaching fine target, (case 2) approaching coarse target, (case 3) zooming backward (circle radius represents the minimum permitted distance from surface).

Hovering. The hovering operation tries to maintain the camera on the current isolevel, looking straight at the model. Hovering is performed through an incremental update of the current position on the isosurface, followed by an update of the view direction. As user input, the camera controller takes a 2D vector indicating the desired direction and amount of motion in the camera plane. We assume in the following that the increment is small. Therefore, large user inputs (large velocities) imply multiple update steps. ITo update the position, we achieve motion by tracing an isoline starting from the current position. The tracing algorithm moves the observer position tangentially to the isosurface, reprojects it onto the isosurface, and finally updates the view direction (see Figure 3(b)). For that, each step is unprojected from screen to world space using the current projection and view transformation looking toward the fine target, thus producing a movement tangential to the isosurface. Projecting the observer onto the isosurface consists of moving the observer position towards the fine target to adjust the distance, and then recomputing the fine target. Working exclusively on fine targets for position computation ensures that we cannot have collisions with the model. Since the gradient direction changes from point to point, a single translation would not be enough to reach the original isosurface, and the projection algorithm is applied iteratively until the correct isolevel is reached within a prescribed relative tolerance (a small fraction of the isosurface value) or a maximum small number of iterations has been done. After this iterative adjustment, a new coarse target is recomputed for orientation computation. At this point the observer position and the view target are available, but we still need an up vector to define the current view transformation. Up-vector definition is a classic problem in camera motion control, and many solutions are available in the literature (see Khan et al. [2005] for up-vector techniques suited for proximal navigation). In this work, we use a global up vector model, which is particularly suitable for exploring standing sculptures. To handle singularities, we limit the camera tilt with respect to the up vector, generating a camera motion similar to the one a human would perform around a statue.

Zooming. Zooming is achieved by switching between the different isosurfaces. The user input modifies the distance of the observer from the coarse target. The coarse target position, which defines the view direction, is not updated during the zooming, thus leaving unchanged the orientation and producing a smooth and intuitive movement. Moving between isosurfaces has the benefit that we always know the distance from the model surface, thus we can easily avoid collisions, simply limiting the distance from the model. We perform the movement only if the new distances from both the fine target and the coarse target are greater or equal than a minimum distance. We obviously need to check the



(a) Temporal smoothing process (b) Free motion (c) Temporally smoothed orientation

Fig. 4. Temporal smoothing. Position and orientation interpolation when reaching coarse target discontinuity. In the 3D views, lines identify view direction in different frames. Large view direction discontinuities are highlighted in red.

fine target to avoid collisions. At the same time, we check the coarse target distance to avoid abrupt rotations when keeping facing the coarse target (see Figure 3(c) (case 1, 2)). Zooming in and zooming out use the same algorithm, since these checks are enough to prevent forward as well as backward collisions (see Figure 3(c) (case 3)).

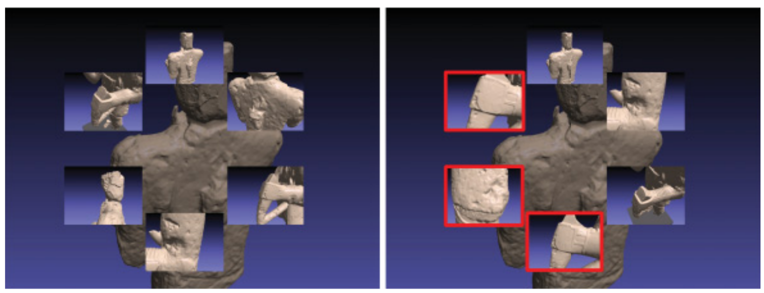
Sudden orientation changes and temporal smoothing. A benefit of moving over an isosurface is the ability of early detecting collisions with invisible parts, as could happen in the HoverCam approach [Khan et al. 2005] when browsing a surface which presents a sudden change of orientation, as in Figure 4(a). HoverCam solves this problem with prediction in the movement direction to avoid hitting the hidden surface. In our case, this is implicit when moving over an isosurface that covers the entire model. It is important to note that our view-dependent representation is complete, that is, it covers the entire model without clipping it outside the view frustum (resolution of far/out-of-view nodes are only reduced) (see Section 7). Sudden surface orientation changes could cause, however, accelerations in user motion, due to fast changes of view direction (coarse target) even in a smoothed surface. This is a simpler situation to handle, as it does not need to modify the path of the camera and only requires introducing acceleration control in the orientation. Working on coarse targets for orientation computations already introduces a degree of smoothness in orientation changes. Moreover, limiting the screen space movement of the coarse target produces a smooth change of orientation also in case of sudden coarse neighbor changes. To achieve this goal, we introduce hysteresis in the process, achieving temporal smoothing. Positions and orientations are thus adaptively interpolated between previous and current view transformation. The interpolation value is given by the ratio between the user input movement and the coarse target screen movement (see Figures 4(b) and 4(c)). Because orientation does not change linearly, we recursively apply this procedure until the coarse target screen movement is within a certain tolerance from the user input. This procedure converges in a few steps, generally 2 or 3. Finally, the current camera view transformation is interpolated at each step with previous camera transformation to produce a smoother movement.

5. IMAGE-BASED NAVIGATION AND POINTS OF INTEREST

We provide the user also with a context-based guided navigation relying on nearest point-of-interest selection. This alternative control method allows the user to explore the object by traveling through a set of interesting views that have been previously defined and annotated. For that, the user is presented with a small set of images consisting of interesting views that can be reached from its position and can be selected with a radial selection interface shown in the projection screen (see Figure 5).

(a) Current view (b) Context-sensitive view selector (c) Final image with overlay

Fig. 5. Radial view selector. A small set of images is presented to the user corresponding to the nearest views available from the current viewpoint. Selecting one of the presented images triggers a camera animation that moves the viewpoint to the location associated to the images and then presents in overlay the associated information.



(a) Selection with similarity filtering (b) Selection without similarity filtering

Fig. 6. Context-based selection. Selected images with (a) and without (b) similarity filtering images from the point of view corresponding to the top view in the radial view selector. In the image without similarity filtering (b), we have highlighted in red the views less related to the current point of view, like the knee and the two closeup of the hand, which are close in terms of viewpoint but not in terms of similarity.

Selecting one of the presented images triggers a camera animation that moves the viewpoint to the location associated to the images and then presents in overlay the associated information (see Figure 5). The information remains visible until the user decides to move to another position. This approach, based on static overlays, which can freely contain 3D sketches to be superimposed to rendered 3D models, greatly simplifies the authoring of annotations by domain experts, which can use standard 2D graphics tools to generate vector graphics representations enhancing and commenting particularly interesting views.

No *hot-spots* or *hyperlinks* are visible during the navigation, but activated on demand by the user (see Section 6). Upon activation, a search is performed for determining the best view candidate set from the user viewpoint. The search of the best view candidates is performed on a separate kd-tree that contains all the views (see Section 7). For each view we store the associated image previously rendered, a description of the view contents, which is then rendered as overlay, and the viewing transformation from which we derive the point of view. Context-based selection selects from the database images that are similar to the current view and easily reachable from the current position (see Section 7).

By using a radial selection interface, the user is no longer required to look at the touch screen for switching between the different views, which are distributed onto a circumference around the center of the display. The upper position is held by the the nearest view with the following views in distance

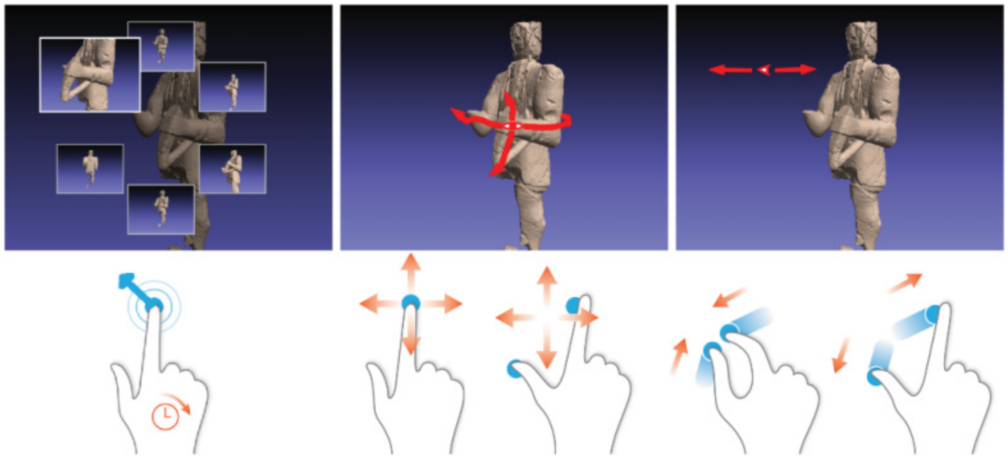


Fig. 7. Multitouch gestures. Device mapping using a multitouch device. Hand images adapted from Wikipedia (contributor GRPH3B18) under a Creative Commons Attribution-ShareAlike license.

order being distributed in a clockwise fashion. Relative displacements from the initial touch position determine the selection in a natural way.

6. DEVICE MAPPING

Our user interface for model manipulation and point-of-interest selection requires minimal user input, and can be mapped from input devices in a variety of ways. In this article, we are interested in a situation where a touch screen is used for motion control and the rendering is displayed onto a big projective screen. For such a setup, we avoid using the touch screen to display content-related information, in order to encourage the user to focus on the visualization screen instead of concentrating on the user interface (see Figure 8(a)). It should be noted that, when using touch input, we not only have to deal with 2D-input-to-3D-output mapping, but we also have to be aware of modality and precision issues [Keefe and Isenberg 2013]. The IsoCam interface helps in this respect because it uses a constrained navigation approach that reduces degrees of freedom to simplify common operations, limits operations to zooming (1D), hovering (2D), and optional twist (1D), employs a fully incremental mode of operation that does not require pointing. We have implemented the interface both with a single-touch as well as a multitouch input. The single-touch interface can be operated also with a one-button 2D mouse. Incremental camera motion control requires only to differentiate zoom versus hovering and to continuously specify the associated 1D or 2D amounts. Point-of-interest selection requires the activation of the circular menu, and the specification of an angle to select the desired point of interest.

Multitouch mapping. Multitouch is our preferred mode of operation. To reduce training times, we considered common single-touch and multitouch gestures, as used for 2D actions in commodity products such as smartphones or tablets, and mapped them to analogue 3D behaviors in our constrained controllers. To the same end, we also found useful to associate multiple gestures to the same action. While no standard exists for mapping 3D touch gestures to 3D manipulations [Keefe and Isenberg 2013], one-finger pan and two-finger rotate-scale-translate gestures are well established and accepted in the 2D realm. Because our controller is constrained and maps 3D motions to locally 2D actions, it is possible to map common 2D RST gestures to analogue 3D motions. This has the advantage of encouraging walk-up-and-use behaviors, with people learning the 2D to 3D mapping while interacting. The one-finger long press is thus mapped to the activation of the point-of-interest menu, and a radial motion in the direction of the desired thumbnail followed by a release performs selection. The single-finger panning gesture is used to control hovering. Direction and velocity are specified by the vector from the first touch position to the current finger position. Two-finger gestures can also be used to control hovering, as well as zooming and up-vector control. Two-finger panning has the same effect as single-finger panning (i.e., hovering control), while two-finger pinch controls zooming, and two-finger rotate controls twist (i.e., up-vector modification).

Single-touch mapping. The single-touch interface is used on desktop applications and in cases where a multitouch screen is not available. Device mapping is similar to the multitouch interfaces, with long press (or right button press for a mouse) for menu activation, and dragging gestures for hovering control and point-of-interest selection. Similarly to other works on unified camera control [Zeleznik and Forsberg 1999], zooming in this case is activated by reserving a specific area of the touch device (a vertical bar placed on the left of the screen), and using dragging in this area for zoom-in (drag up) or zoom-out (drag down).

7. SCALABILITY

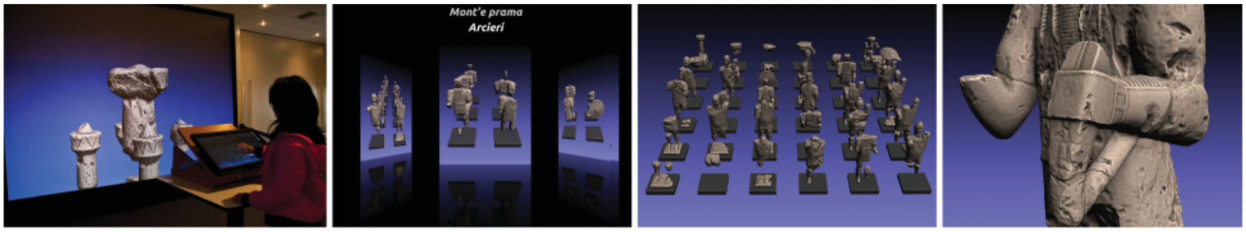
Both the rendering and user interface require specialized spatial indexing and multiresolution structures and adaptive algorithms to ensure real-time performance on large datasets (billions of triangles and hundreds of points of interest per model).

We employ kd-trees to organize the points of interest, and have extended the Adaptive TetraPuzzles (ATP) approach [Cignoni et al. 2004], which already provides the features and performance required for massive model rendering, to implement the required geometric queries. The ATP coarse-grained multiresolution structure is based on a hierarchical tetrahedral subdivision of the input volume. Each tetrahedron contains at leaf level a portion of the original model, while in inner levels, coarser representations are constructed by constrained simplification from children tetrahedra. Each node is made of a few thousands of triangles, leveraging the cost of the CPU traversal, amortizing node selection over many graphics primitives, and properly exploiting current GPU bandwidth and processing capabilities. The algorithm is able to produce view-dependent adaptive seamless representations at high frame rates. The original algorithm has been improved, similar to what has been done by Balsa et al. [2013] by using a diamond structure that is refined with the use of a double heap refinement/coarsening strategy, to produce time-critical interruptible view-dependent cut adaptations. At each frame, the view-adapted cut is used for all operations.

Nearest neighbor computation for camera control. The use of the multiresolution structure is mandatory to be able to perform, multiple times per frame, fast nearest neighbor search at multiple scales on massive models made up of hundreds of millions of samples, as required by our constrained camera controller (see Section 4). We perform all searches on the view-adapted tetrahedron graph. Each node of the tetrahedral hierarchy has been enriched with an internal bounding box hierarchy, which permits nearest neighbor computation to be performed at a fast pace. This box tree is constructed by hierarchically bi-sectioning the node's vertex array and calculating the associated bounding box, exploiting the fact that the vertices in the triangle strip are sorted in a spatially coherent order, similarly to ray strips methods [Balsa Rodriguez et al. 2013; Lauterbach et al. 2007], which, however, construct a similar hierarchy on triangle strip. This is not required for our meshless approximation. Each node of the axis-aligned box hierarchy contains, in addition the implicitly defined bounding box, the first and last indices of the vertex array portion it contains in order to be able to refine the search. The per-node bounding box hierarchy gives us the ability to perform nearest neighbor search at different levels of resolution, just by specifying the maximum level of the box hierarchy at which to perform the search.

If the level is coarser than the box's maximum level, the search will return the point contained at half of the vertex array of that node. Returning a point belonging to the vertex array gives a better surface approximation with respect to returning, for example, the center of the bounding box, which could be placed away from the surface, depending on the point distribution inside the box. The fine search will look for the nearest neighbor among the leaf vertices (in our implementation there are $N < 16$ vertices at leaf level). Hence, coarse target search is performed near the root of the bounding box hierarchy, whereas fine target search is performed deeper in the hierarchy. To remove discrete sampling effects, each nearest neighbor search returns not only a single sample but also a Gaussian weighted average $p^* = \frac{\sum_{i=1}^{K} w_i \cdot p_i}{\sum_{i=1}^{K} w_i}$ of the first K-nearest neighbors identified by the knn-search, where $w_i = (1 - \frac{d_i^2 - d_{\min}^2}{d_{\max}^2 - d_{\min}^2})^4$ is the Gaussian-like weight for each of the K points, d_i is the distance of p_i to the eye, and $d_{\max}$ and $d_{\min}$ are the minimum and maximum distances among all d_i values. In our implementation, we use $K = 8$ for fine target computation and $K = 16$ for coarse target computation to provide smoothing of sampling discontinuities at small CPU costs. Note, again, that this weighted averaging process is aimed at reducing discontinuities due to discrete sampling, not at smoothing the surface. Surface smoothing is not required for the fine target, used for position computation, while it is achieved for the coarse target through the sampling sparsification (subsampling) obtained by controlling search depth in the box hierarchy.

Scalable selection of points of interest. Our context-based user interface requires scalable methods to efficiently select a good set of points of interest from a large database using a view similarity metric. We start from the fact that our situation is different from just image matching, since we have full information on the camera parameters as well as on the imaged 3D model. Thus, instead of computing image correlations or extracting and matching feature sets, we can compute similarity based on the fact that, for two similar views, a given point model would project to about the same image pixel. To efficiently perform the operation, we split the search into two phases. We organize all views in a kd-tree, indexed by the observer position, the selection of the M view candidates starts with a traversal of the kd-tree, looking for the $N \gg M$ points of view nearest to the user viewpoint that satisfy the following two constraints: (a) the angle ϕ between the user view direction and the candidate view direction must be below a certain threshold (in our case $\phi \leq 90^\circ$), and (b) there should not be any obstacle between the user viewpoint and the candidate view (we perform a raycasting between the two points of view to check for that constraint). Constraint (a) ensures that selected viewpoint has approximately the same orientation of the current view, while (b) ensures that the current viewpoint sees the target viewpoint, and a simple linear path can move the camera to the target point without collisions. Once the N-nearest neighbors set satisfying both constraints has been determined, the set is further filtered to select the most similar views. Intuitively, two views are similar if surface samples in 3D model geometry project approximately to the same pixels in the two images. We perform the computation in Normalized Device Coordinates (NDC) (see Figure 6), using minimal information. For that, a small random set of 3D points is extracted from the view-adapted tetrahedron graph. This is done using a simple traversal of the graph leafs, selecting a few points per node. These points are then projected using the view parameters both from the candidate view and the user viewpoint, in order to calculate the average screen space distance between the two point sets, defined as $d = \frac{1}{S} \cdot \sum_{i=1}^{S} \| PV_j p_i - PV_{current} p_i \|$, where P is the projection matrix, V_j is the candidate's view matrix, $V_{current}$ is the current view matrix, and p_i corresponds to each of the S random selected points. Only $N \leq M$ views pass this last filter where $d \leq 2$ is required (as we work in NDC coordinates, a distance of two corresponds to screen width or height). A major advantage is this a purely geometry-driven technique and the only required information on the stored views are the viewing parameters.



(a) Museum setup (b) Main menu (c) Mont'e Prama database (d) Rendered model

Fig. 8. Application setup. (a) Setup of touch screen large projection device on a exhibition. (b) Main menu interface. (c) Mont'e Prama full database. (d) Detail of a single Archer statue.

8. IMPLEMENTATION AND RESULTS

The techniques presented in this work have been implemented in a system for exploration of highly detailed models in museum settings. The system has been successfully used in a variety of settings, in particular for the exploration of a set of 3D models derived from the 3D scan acquisition of the statues of Mont'e Prama, ancient stone sculptures created by the Nuragic civilization of Sardinia, Italy (see Figures 8(c) and 8(d)). The 3D models of these statues are highly detailed and often made of a few disconnected parts, posing important problems to navigation techniques. See Bettio et al. [2013, 2014] for details on the Mont'e Prama dataset.

8.1 Reference Implementation

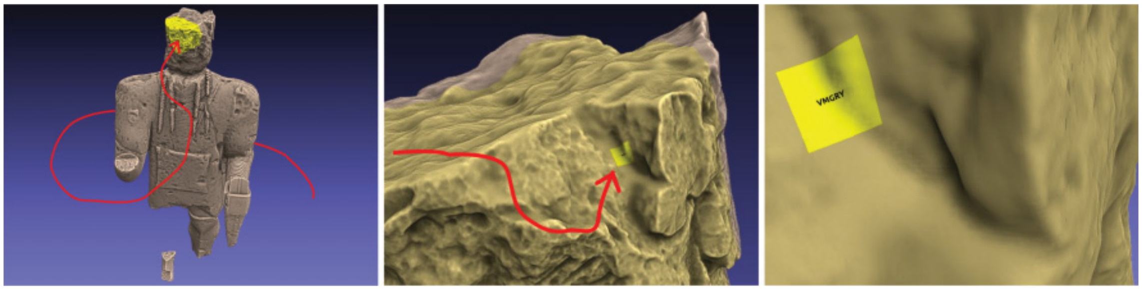
A reference system integrating all techniques described in this article has been implemented on Linux using OpenGL and GLSL for the rendering, and Qt 4.7 for the interface, while the database component relies on BerkeleyDB. The hardware setup for the interactive stations was composed of a 2.5m-diagonal back-projection screen, a 3000 ANSI Lumen *projectiondesign F20* SXGA+ projector, and a 23" ACER T230H optical multitouch screen, both controlled by a PC with with Ubuntu Linux 12.10, a Intel Core i7-3820 @ 3.6Ghz with 8GB of RAM, and a NVIDIA GTX 680 GPU. The multitouch screen was designed to be placed 1.5m away from the projection screen in order to allow visitors to see the whole display without being occluded by the user that was interacting.

The system has been tested on a variety of models including the Mont'e Prama statues. Models were rendered using a target resolution of 0.5 triangles/pixel, leading to graphs cuts containing 250 leaf nodes, on average, with approximately 8K triangles/node. Interactive performance was achieved for all datasets, with frame rates constantly above 30fps and negligible interaction delays. Reusing the multiresolution structure used for rendering also for object queries during interaction proved successful, as camera transformation computation in our camera controller always took just between 10% and 30% of the graph adaptation time required for rendering.

8.2 User Study

To assess the IsoCam exploration controller we carried out a thorough user evaluation involving quantitative and subjective measurements based on interactive tasks carried out by a number of volunteers.

Goal. The main goal of the evaluation was to assess whether the proposed camera controller is adequate for usage in the typical scenario of virtual museums, where many users with different skills and experiences try to interactively explore digital models in order to highlight details at various scales. Given the large number of 3D object exploration techniques, it is impossible to carry out a fully comprehensive comparison considering all the existent navigation systems. We thus limited our comparison to the virtual 3D trackball [Henriksen et al. 2004], and the HoverCam proximal navigation



(a) Global positioning task (b) Local positioning task (c) Target writing in the model

Fig. 9. User interface evaluation. Tasks consisted of (a) manipulating the model until reaching given specified global positions highlighted in yellow, (b) exploring a search area to find meaningful details, (c) identifying a five-letter writing textured over the model.

controller [Khan et al. 2005]. The virtual trackball was chosen because it is the interactive camera controller most commonly employed in virtual environments and is considered easy and immediate to learn. HoverCam was considered because it is the most similar to IsoCam between the object-aware techniques available in literature.

Setup. The experimental setup considered the reference system implementation described in Section 8.1. All user interfaces were operated using the same precise optical touch screen device using a multitouch device mapping. IsoCam and HoverCam used the same device mapping presented in Section 6. The Virtual TrackBall was implemented by rigidly attaching the object to the user's hand and uses a typical RST approach: With one finger, the user generates 2-DOF rotations on the model's bounding sphere, whereas with two fingers (pinch), the user performs zoom in or out by increasing or reducing the distance between the fingers, or pan the model by moving the fingers inside the workspace. As a testing model, we selected a 80M triangles Archer statue from the Mont'e Prama collection, which has detailed carvings and is missing some parts: half bow, one hand, and half leg (see Figure 9(a)). The reconstructed model is thus composed by two disconnected parts (whole body and disconnected right foot), and presents many sharp features, surface saddles peaks and valleys, whose exploration represents a nontrivial task.

Tasks. The experiments consisted in letting users try the three different manipulation controllers (IsoCam, TrackBall [Henriksen et al. 2004], and HoverCam [Khan et al. 2005]) in the context of a target-oriented interaction task [Martinet et al. 2009]. We designed our task to measure performance in the macrostructure and microstructure inspections tasks typical of cultural heritage model explorations. Participants were asked to manipulate the model until reaching given specified global positions and to perform local small-scale explorations of the model surface with the goal of finding meaningful information. Targets were indicated by yellow patches lying on the Mont'e Prama bowman model, and the goal was to reach them as quickly as possible by starting from a fixed position and adequately manipulating the model in order to adequately position the viewpoint over the indicated areas (see Figure 9(a)). When the target area was reached, the yellow patch area became an exploring area (see Figure 9(b)) and users had to search and identify a five-letter writing textured above the surface of the bowman (see Figure 9(c)). Users had to read correctly the writing by moving locally the model up to reach an adequate orientation and scale, and identify it between five different choices indicated in buttons appearing under the touch area.

Participants. For the user analysis, 20 participants (17 males and 3 females, with ages ranging from 17 to 65, mean 36.5 ± 12.5 years) were recruited between the participants of the various exhibitions, CRS4 employees (including researchers, network administrators and administrative staff), and high school students. They were subdivided in two groups according to their experience with 3D interfaces (12 novices and 8 experts).

Design. Each participant saw the three interfaces in randomized order. Users were first allowed to become familiar with the controller by watching a brief video showing how it works (part of the help system of the museum installation), trying it for 2 minutes, and performing one task just for testing. After the training session, the measured tests consisted of five trials, where targets were randomly selected from a list of 15 potential candidates, as to avoid any bias due to a priori knowledge of target positions. For a complete testing session, users needed times ranging from 15 to 20 minutes. A trial was considered aborted if the target area was not reached within 40 seconds, or the inscription was not correctly identified within 40 seconds from reaching the target area. The times needed for reaching the target areas (global positioning) and for identifying the correct writings (local browsing) were measured and recorded. In summary, the complete test design consisted of 20 participants, each one testing three camera controllers with two tasks (positioning and identification) on five targets for a total of 600 time measurements. At the end of the experiments, participants were also asked to fill a questionnaire comparing the performance of the three controllers by indicating a score in a 5-point Likert scale with respect to the following characteristics: ease of learning, ease of reaching desired positions, and perceived 3D exploring quality. Finally, participants were asked to indicate their preferred controllers.

Performance evaluation. Before collecting the results, we expected that IsoCam would perform similarly to HoverCam and TrackBall for global positioning, while we expected a significant improvement in the local positioning task. We also expected that expert users might have similar performances for all the three camera controllers, and significant differences when the controllers are used by novice users. We performed an analysis of task completion times for global positioning and identification, and total times. Figure 10 show the boxplots of the task completion times, for novice and expert users, as rendered by the *R* package [R Core Team 2013]. The bottom and top of each box are the first and third quartiles, the band inside the box is the second quartile (the median), and the ends of the whiskers extending vertically from the boxes represent the lowest datum still within 1.5 interquartile range (IQR) of the lower quartile, and the highest datum still within 1.5 IQR of the upper quartile. Outliers are indicated as small circles. Analysis of boxplots in Figure 10 (bottom row) reveals that experts had similar performances with all interfaces, even if they were slightly faster with IsoCam. Through direct observation of expert users while performing the tasks, we noticed that the common strategy for completing the global positioning tasks consisted of moving the model at macroscale before performing one zooming operation toward the target area. For the identification tasks, most expert users zoomed out the area to recognize the detail, and finally zoomed in to the target writing. In this way, for most of the targets, performances were more or less similar independently from the interface employed. However, for targets positioned in sharp areas of the model or close to disconnected parts, a number of users experienced problems with HoverCam and TrackBall. This explains the presence of some distant outliers in the boxplot of identification times for HoverCam and TrackBall. On the other hand, from top row of Figure 10, it appears evident that novice users had better performances and felt more comfortable with IsoCam, especially during the identification task. Direct observation of interaction during the tasks revealed that naïve participants did not follow any common strategy for completing the tasks, but the movement was mostly instinctive, with a gradual usage of zooming and the tendency of trying to slide along the surface when performing the exploration and identification task. In most of the trials, independently from the position of the target, novice users performed continuous motion corrections, especially when using HoverCam and TrackBall, while with IsoCam they

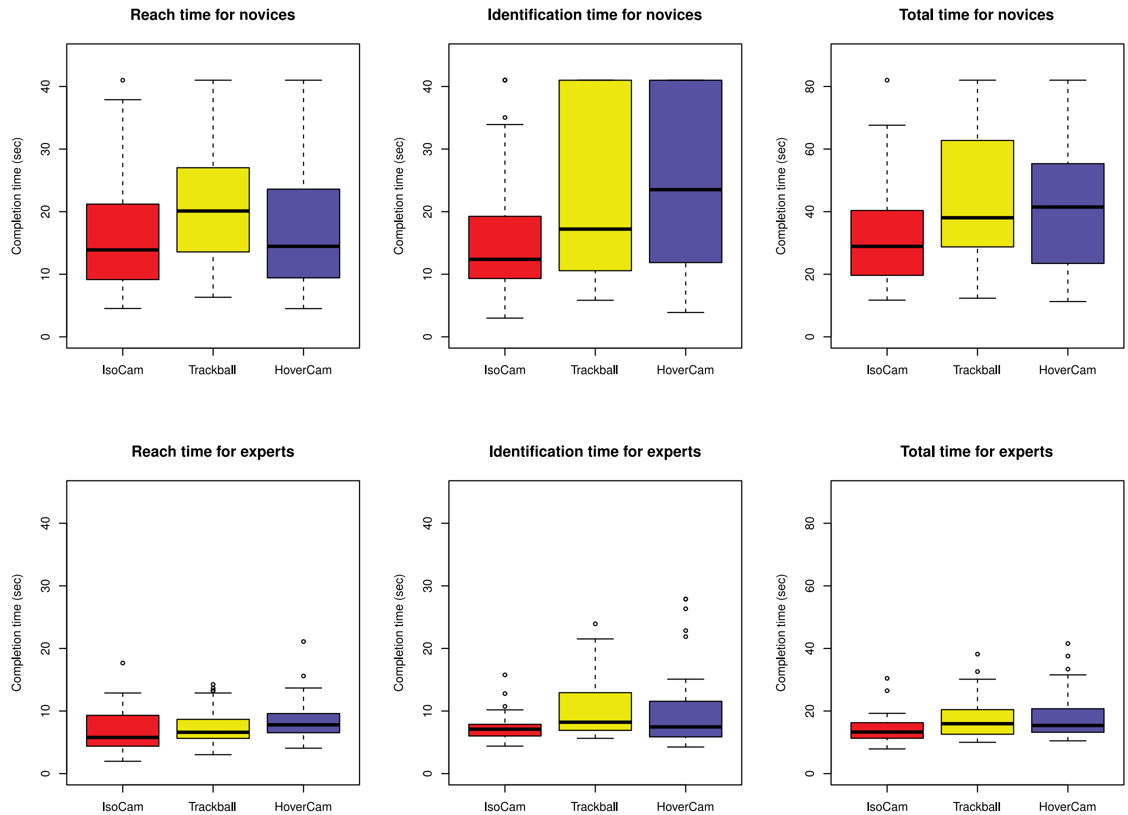
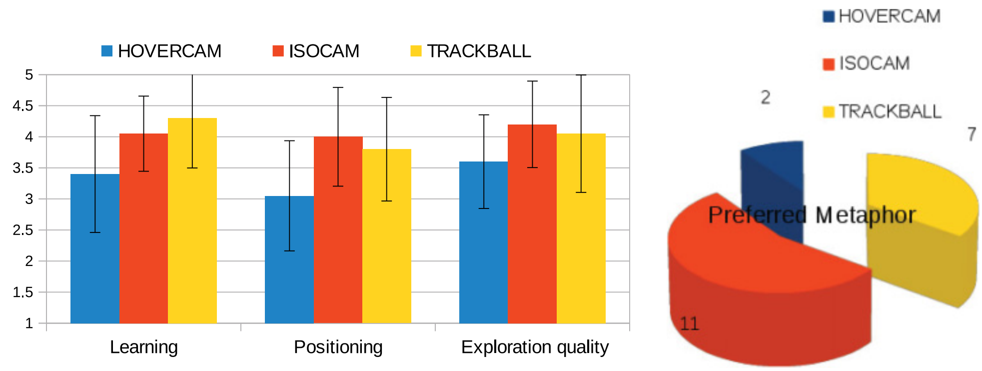


Fig. 10. Performance evaluation. The proposed IsoCam camera controller was compared to the Virtual TrackBall [Henriksen et al. 2004] and HoverCam [Khan et al. 2005] with respect to performance and perceived navigation quality during the exploration of a complex Mont'e Prama statue model. In boxplots, red indicates IsoCam, yellow indicates TrackBall, and blue indicates HoverCam.

were able to keep smooth trajectories. To this end, particularly interesting is the size of IQR for the identification time (see second column of Figure 10), from which it appears evident that both novices and experts had homogeneous performances during the local browsing task with IsoCam, but not with HoverCam and TrackBall. These results can be explained by considering that a direct surface orbiting technique like HoverCam can suffer when dealing with sharp features and disconnections, such as the ones contained the Mont'e Prama bowman model, while TrackBall needs many mode switches between rotations, translations, and zoom to be able to inspect highly detailed surface models while remaining at a constant distance.

Qualitative evaluation. Figure 11(a), showing histograms with error bars for qualitative results, provides indication that users found the Virtual TrackBall easier to learn but not simple to operate, since they perceived that with IsoCam they could more easily reach targets, and they also felt more comfortable during exploration. The easy-to-learn property of the TrackBall is in large part due to the fact that the technique is (falsely) perceived as standardized because most users have already been exposed to it, at least in standard 2D RST interfaces. It appears also evident that IsoCam revealed to be strongly appealing for users (see Figure 11(b)). In addition, from think-a-loud comments, we derived



(a) Subjective comparison of camera controllers (b) Preferred camera controller

Fig. 11. Qualitative evaluation. (a) Users were asked to evaluate the performance of the three camera controllers by indicating a score in a 5-point Likert scale with respect to ease of learning, ease of reaching desired positions, and perceived 3D exploring quality. (b) Users were asked to indicate the preferred controller.

that the large majority of novice users noted and appreciated the superior smoothness of the IsoCam camera controller with respect to the HoverCam.

8.3 Exhibition

The system has been successfully employed in a large exhibition held during the 65th International Trade Fair of Sardinia, held from April 25 to May 6, 2013 (see Figure 1). The Mont'e Prama restoration project was presented to the visitors, together with virtual stands for interactively exploring the entire set of statues. In that setting, visitors were presented with a simple touch interface to select the statue to be explored. A gallery of models was shown on the multitouch screen, and specific information about the currently selected object was shown in the projective screen. Sliding to left and right allowed the user to change the selected model, while a click on the central image invokes the associated action: showing the objects under the selected hierarchy group or exploring the selected object. Once the model to explore has been selected, the visualization application came in, providing the user with an almost empty multitouch screen (only the buttons for help and going back were shown), and the high-resolution model being displayed in the projective screen (see Figure 8(b)). In order to interact with it, users simply had to perform touch gestures on the multitouch screen without any visual feedback other than the visualization. Being that these gestures are relative to the initial touch position, the user simply had to associate their movement on the screen surface with the behavior of the navigation, as would happen with a joystick, for instance. A total of approximately 3,500 visitors attended the exhibition, and many of them interacted with the models using our interface (see accompanying video). Typically, groups of people gathered in front of the interactive stations and freely interacted with the system. The unobtrusive interface and setup was very effective in this setting. Informal feedback received during the exhibition was very positive. In particular, all users particularly appreciated the size of projection, the interactivity of the navigation, and the fine details of the statues provided by the multiresolution renderer.

A follow-up of this work is now the basis for permanent exhibitions on Mont'e Prama statues at the National Archaeological Museum in Cagliari, Italy and at the Civic Museum in Cabras, Italy. Each of these two installations employ a 7500 Lumen projector illuminating a 2.5m-high 16:10 screen positioned in portrait mode. This configuration permits rendering the statues at full scale. By matching projection parameters with the position of the viewer operating the console, we can achieve a natural viewing experience.

9. CONCLUSIONS

We have presented an interactive system for natural exploration of extremely detailed surface models with indirect touch control in a room-sized workspace, suitable for museums and exhibitions. Furthermore, a museum installation has been set up and thousands of visitors had the possibility, during 2 weeks, of browsing high-resolution 3D laser scan models of the 38 restored statues of the Mont'e Prama complex. We have presented a novel camera controller based on distance field surfing, enriched with a dynamic selection of precomputed views and point of interest, which does not require the visualization of hot-spots over the model, but is able to dynamically identify the best views near the current user position and to display connected information in overlay. The presented IsoCam camera controller has been compared with other two consolidated approaches, collecting quantitative and qualitative results from a series of tests involving 20 people. The camera controller appears to be intuitive enough even for casual users who quickly understand how to browse statue models in a short trial period. Our indirect control low-degree-of-freedom controller allows users to pan, rotate and scale, with simple and natural gestures, massive models, while always facing the model surface, without requiring precise pointing methods, and without the monolithic surface limitation of previous orbiting and hovering techniques. The resulting virtual environment, which combines ease of use with high-fidelity representation and low-cost setup, appears to be well suited for installations at museums and exhibition centers. Our current work concentrates on improving interaction quality and on extending the system for supporting bidirectional connection between multiple multimedia types as well as narrative contents.

ACKNOWLEDGMENTS

We also acknowledge the contribution of Sardinian Regional Authorities. We thank the personnel of *ArcheoCAOR* and of the *Centro di conservazione e restauro dei beni culturali* of Li Punti for their participation in the design process.

REFERENCES

C. Andujar, A. Chica, and P. Brunet. 2012. Cultural heritage: User-interface design for the Ripoll monastery exhibition at the National Art Museum of Catalonia. *Computers and Graphics* 36, 1 (2012), 28–37.

Marcos Balsa Rodriguez, Enrico Gobbetti, Fabio Marton, and Alex Tinti. 2013. Compression-domain seamless multiresolution visualization of gigantic meshes on mobile devices. In *Proc. ACM Web3D*. ACM, 99–107.

Isaac Besora, Pere Brunet, Marco Callieri, Antoni Chica, Massimiliano Corsini, Matteo Dellepiane, Daniel Morales, Jordi Moyés, Guido Ranzuglia, and Roberto Scopigno. 2008. Portalada: A virtual reconstruction of the entrance of the Ripoll monastery. In *Proc. 3DPVT*. 89–96.

Fabio Bettio, Enrico Gobbetti, Fabio Marton, Alex Tinti, Emilio Merella, and Roberto Combet. 2009. A point-based system for local and remote exploration of dense 3D scanned models. In *Proc. the 10th International Symposium on Virtual Reality, Archaeology and Cultural Heritage*. 25–32.

Fabio Bettio, Enrico Gobbetti, Emilio Merella, and Ruggero Pintus. 2013. Improving the digitization of shape and color of 3D artworks in a cluttered environment. In *Proc. Digital Heritage*. To appear.

Fabio Bettio, Alberto Jaspe Villanueva, Emilio Merella, Ruggero Pintus, Fabio Marton, and Enrico Gobbetti. 2014. Mont'e scan: Effective shape and color digitization of cluttered 3D artworks. *Submittted for publication* (2014).

Doug A. Bowman, Chris North, Jian Chen, Nicholas F. Polys, Pardha S. Pyla, and Umur Yilmaz. 2003. Information-rich virtual environments: Theory, tools, and research agenda. In *Proc. ACM VRST*. ACM, 81–90.

Nicholas Burtnyk, Azam Khan, George Fitzmaurice, Ravin Balakrishnan, and Gordon Kurtenbach. 2002. StyleCam: Interactive stylized 3D navigation using integrated spatial and temporal controls. In *Proc. ACM UIST*. ACM, 101–110.

Nicolas Burtnyk, Azam Khan, George Fitzmaurice, and Gordon Kurtenbach. 2006. ShowMotion: Camera motion based 3D design review. In *Proc. ACM I3D*. ACM, 167–174.

T. Butkiewicz and C. Ware. 2011. Multi-touch 3D exploratory analysis of ocean flow models. In *Proc. IEEE Oceans*. IEEE.

Marco Callieri, Chiara Leoni, Matteo Dellepiane, and Roberto Scopigno. 2013. Artworks narrating a story: A modular framework for the integrated presentation of three-dimensional and textual contents. In *Proc. ACM Web3D*. ACM, 167–175.

Marco Callieri, Federico Ponchio, Paolo Cignoni, and Roberto Scopigno. 2008. Virtual inspector: A flexible visualizer for dense 3D scanned models. *IEEE Computer Graphics and Applications* 28, 1 (2008), 44–54.

Michael Chen, S. Joy Mountford, and Abigail Sellen. 1988. A study in interactive 3-D rotation using 2-D control devices. In *Proc. SIGGRAPH*. ACM, 121–129.

Marc Christie and Patrick Olivier. 2009. Camera control in computer graphics: models, techniques and applications. In *ACM SIGGRAPH ASIA Courses*. ACM, 3:1–3:197.

Paolo Cignoni, Fabio Ganovelli, Enrico Gobbetti, Fabio Marton, Federico Ponchio, and Roberto Scopigno. 2004. Adaptive tetrapuzzles: Efficient out-of-core construction and visualization of gigantic multiresolution polygonal models. *ACM Transactions on Graphics* 23, 3 (2004), 796–803.

Dane Coffey, Nicholas Malbraaten, Trung Bao Le, Iman Borazjani, Fotis Sotiropoulos, Arthur G. Erdman, and Daniel F. Keefe. 2012. Interactive slice WIM: Navigating and interrogating volume data sets using a multisurface, multitouch VR interface. *IEEE Transactions on Visualization and Computer Graphics* 18, 10 (2012), 1614–1626.

Michel Crampes, Jeremy de Oliveira-Kumar, Sylvie Ranwez, and Jean Villerd. 2009. Visualizing social photos on a Hasse diagram for eliciting relations and indexing new photos. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (Nov. 2009), 985–992.

Jean-Baptiste de la Rivière, Cédric Kervégant, Emmanuel Orvain, and Nicolas Dittlo. 2008. CubTile: A multi-touch cubic interface. In *Proc. ACM VRST*. ACM, 69–72.

Andreas Dietrich, Enrico Gobbetti, and Sung-eui Yoon. 2007. Massive-model rendering techniques: A tutorial. *IEEE Computer Graphics and Applications* 27, 6 (Nov./Dec. 2007), 20–34.

Boris Epshtein, Eyal Ofek, Yonatan Wexler, and Pusheng Zhang. 2007. Hierarchical photo organization using geo-relevance. In *Proc. ACM GIS*. ACM, 18:1–18:7.

H. J. Falk and L. D. Dierking. 2000. *Learning from Museums: Visitor Experience and the Making of Meaning*. Rowman & Littlefield.

Peter Faraday and Alistair Sutcliffe. 1997. Designing effective multimedia presentations. In *Proc. ACM SIGCHI*. ACM, 272–278.

George Fitzmaurice, Justin Matejka, Igor Mordatch, Azam Khan, and Gordon Kurtenbach. 2008. Safe 3D navigation. In *Proc. ACM I3D*. ACM, 7–15.

Alessandro E. Foni, George Papagiannakis, and Nadia Magnenat-Thalmann. 2010. A taxonomy of visualization strategies for cultural heritage applications. *Journal on Computing and Cultural Heritage* 3, 1 (July 2010), 1:1–1:21.

Chi-Wing Fu, Wooi-Boon Goh, and Junxiang Allen Ng. 2010. Multi-touch techniques for exploring large-scale 3D astrophysical simulations. In *Proc. ACM SIGCHI*. ACM, 2213–2222.

Andreas Girgensohn, Frank Shipman, Lynn Wilcox, Thea Turner, and Matthew Cooper. 2009. MediaGLOW: Organizing photos in a graph-based workspace. In *Proc. ACM IUI*. ACM, 419–424.

Enrico Gobbetti, David Kasik, and Sung-eui Yoon. 2008. Technical strategies for massive model visualization. In *Proc. ACM Solid and Physical Modeling Symposium*. ACM, 405–415.

Timo Götzelmann, Pere-Pau Vázquez, Knut Hartmann, Andreas Nürnberger, and Thomas Strothotte. 2007. Correlating text and images: Concept and evaluation. In *Proc. Smart Graphics*. Springer-Verlag, Berlin, 97–109.

Martin Hachet, Jean-Baptiste de la Rivière, Jrmy Laviole, Aurlie Coh, and Sbastien Cursan. 2013. Touch-based interfaces for interacting with 3D content in public exhibitions. *IEEE Computer Graphics and Applications* 33, 2 (March 2013), 80–85.

Mark Hancock, Sheelagh Carpendale, and Andy Cockburn. 2007. Shallow-depth 3D interaction: Design and evaluation of one-, two- and three-touch techniques. In *Proc. ACM SIGCHI*. ACM, 1147–1156.

Mark Hancock, Otmar Hilliges, Christopher Collins, Dominikus Baur, and Sheelagh Carpendale. 2009. Exploring tangible and direct touch interfaces for manipulating 2D and 3D information on a digital table. In *Proc. ITS*. ACM, 77–84.

Andrew J. Hanson and Eric A. Wernert. 1997. Constrained 3D navigation with 2D controllers. In *Proc. IEEE Visualization*. IEEE, 175–182.

Knud Henriksen, Jon Sporring, and Kasper Hornbæk. 2004. Virtual trackballs revisited. *IEEE Transactions on Visualization and Computer Graphics* 10, 2 (March 2004), 206–216.

Chuljin Jang, Taijin Yoon, and Hwan-Gue Cho. 2009. A smart clustering algorithm for photo set obtained from multiple digital cameras. In *Proc. ACM SAC*. ACM, 1784–1791.

Jacek Jankowski and Stefan Decker. 2012. A dual-mode user interface for accessing 3D content on the world wide web. In *Proc. WWW*. ACM, 1047–1056.

Jacek Jankowski and Martin Hachet. 2013. A survey of interaction techniques for interactive 3D environments. In *Eurographics STAR*.

Jacek Jankowski, Krystian Samp, Izabela Irzynska, Marek Jozwowicz, and Stefan Decker. 2010. Integrating text with video and 3D graphics: The effects of text drawing styles on text readability. In *Proc. ACM SIGCHI*. ACM, 1321–1330.

D. F. Keefe and T. Isenberg. 2013. Reimagining the scientific visualization interaction paradigm. *Computer* 46, 5 (May 2013), 51–57. DOI:http://dx.doi.org/10.1109/MC.2013.178

Azam Khan, Ben Komalo, Jos Stam, George Fitzmaurice, and Gordon Kurtenbach. 2005. HoverCam: Interactive 3D navigation for proximal object inspection. In *Proc. I3D*. ACM, 73–80.

Tijmen Klein, Florimond Guéniat, Luc Pastur, Frédéric Vernier, and Tobias Isenberg. 2012. A design study of direct-touch interaction for exploratory 3D scientific visualization. *Computer Graphics Forum* 31 (2012), 1225–1234.

Sebastian Knoedel and Martin Hachet. 2011. Multi-touch RST in 2D and 3D spaces: Studying the impact of directness on user performance. In *Proc. ACM 3DUI*. ACM, 75–78.

Tsvi Kuflik, Oliviero Stock, Massimo Zancanaro, Ariel Gorfinkel, Sadek Jbara, Shahar Kats, Julia Sheidin, and Nadav Kashtan. 2011. A visitor's guide in an active museum: Presentations, communications, and reflection. *Journal on Computing and Cultural Heritage* 3, 3, Article 11 (2011), 11:1–11:25 pages.

Gordon Kurtenbach, George Fitzmaurice, Thomas Baudel, and Bill Buxton. 1997. The design of a GUI paradigm based on tablets, two-hands, and transparency. In *Proc. ACM SIGCHI*. ACM, 35–42.

Christian Lauterbach, Sung-Eui Yoon, and Dinesh Manocha. 2007. Ray-Strips: A compact mesh representation for interactive ray tracing. In *Proc. IEEE/EG Symposium on Interactive Ray Tracing*. IEEE, 19–26.

Claes Lundstrom, Thomas Rydell, Camilla Forsell, Anders Persson, and Anders Ynnerman. 2011. Multi-touch table system for medical visualization: Application to orthopedic surgery planning. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (Dec. 2011), 1775–1784.

Anthony Martinet, Géry Casiez, and Laurent Grisoni. 2009. 3D positioning techniques for multi-touch displays. In *Proc. ACM VRST*. ACM, 227–228.

Anthony Martinet, Gery Casiez, and Laurent Grisoni. 2012. Integrality and separability of multitouch interaction techniques in 3D manipulation tasks. *IEEE Transactions on Visualization and Computer Graphics* 18, 3 (March 2012), 369–380.

Fabio Marton, Marco Agus, Enrico Gobbetti, Giovanni Pintore, and Marcos Balsa Rodriguez. 2012. Natural exploration of 3D massive models on large-scale light field displays using the FOX proximal navigation technique. *Computers & Graphics* 36, 8 (December 2012), 893–903.

James McCrae, Igor Mordatch, Michael Glueck, and Azam Khan. 2009. Multiscale 3D navigation. In *Proc. I3D*. ACM, 7–14.

Clement Moerman, Damien Marchal, and Laurent Grisoni. 2012. Drag'n Go: Simple and fast navigation in virtual environment. In *Proc 3DUI*. IEEE, 15–18.

Tomer Moscovich and John F. Hughes. 2008. Indirect mappings of multi-touch input using one and two hands. In *Proc. ACM SIGCHI*. ACM, 1275–1284.

João Mota, Manuel J. Fonseca, Daniel Gonçalves, and Joaquim A. Jorge. 2008. Agrafo: A visual interface for grouping and browsing digital photos. In *Proc. ACM AVI*. ACM, 494–495.

Giovanni Pintore, Enrico Gobbetti, Fabio Ganovelli, and Paolo Brivio. 2012. 3DNSITE: A networked interactive 3D visualization system to simplify location recognition in crisis management. In *Proc. ACM Web3D*. ACM, 59–67.

Nicholas F. Polys, Doug A. Bowman, and Chris North. 2011. The role of depth and gestalt cues in information-rich virtual environments. *International Journal of Human-Computer Studies* 69, 1–2 (Jan. 2011), 30–51.

R Core Team. 2013. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Jason L. Reisman, Philip L. Davidson, and Jefferson Y. Han. 2009. A screen-space formulation for 2D and 3D direct manipulation. In *Proc. ACM UIST*. ACM, 69–78.

Dong-Sung Ryu, Woo-Keun Chung, and Hwan-Gue Cho. 2010. PHOTOLAND: A new image layout system using spatio-temporal information in digital photos. In *Proc. ACM SAC*. ACM, 1884–1891.

Ken Shoemake. 1992. ARCBALL: A user interface for specifying three-dimensional orientation using a mouse. In *Proc. SIGGRAPH*. Morgan Kaufmann, San Francisco, CA, 151–156.

Henry Sonnet, Sheelagh Carpendale, and Thomas Strothotte. 2005. Integration of 3D data and text: The effects of text positioning, connectivity, and visual hints on comprehension. In *Human-Computer Interaction (INTERACT'05)*, Maria Francesca Costabile and Fabio Patern (Eds.). Lecture Notes in Computer Science, Vol. 3585. Springer, Berlin, 615–628. DOI:http://dx.doi.org/10.1007/11555261_50

Malte Weiss, Simon Voelker, Christine Sutter, and Jan Borchers. 2010. BendDesk: Dragging across the curve. In *Proc. ACM International Conference on Interactive Tabletops and Surfaces*. ACM, 1–10.

Raphael Wimmer, Fabian Hennecke, Florian Schulz, Sebastian Boring, Andreas Butz, and Heinrich Hussmann. 2010. Curve: Revisiting the digital desk. In *Proc, NordiCHI*. ACM, 561–570.

Sung-eui Yoon, Enrico Gobbetti, David Kasik, and Dinesh Manoc ha. 2008. *Real-time Massive Model Rendering*. Synthesis Lectures on Computer Graphics and Animation, Vol. 2. Morgan and Claypool.

Lingyun Yu, Pjotr Svetachov, Petra Isenberg, Maarten H. Everts, and Tobias Isenberg. 2010. FI3D: Direct-touch interaction for the exploration of 3D scientific visualization spaces. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (Nov. 2010), 1613–1622.

Robert Zeleznik and Andrew Forsberg. 1999. UniCam: 2D gestural camera controls for 3D environments. In *Proc. Symposium on Interactive 3D Graphics*. ACM, 169–173.

Received July 2013; revised February 2014; accepted March 2014