Natural exploration of 3D massive models on large-scale light field displays using the FOX proximal navigation technique

Fabio Marton, Marco Agus, Enrico Gobbetti, Giovanni Pintore, Marcos Balsa Rodriguez

CRS4, POLARIS Ed. 1, 09010 Pula (CA), Italy – www.crs4.it/vic/

Abstract

We report on a virtual environment for natural immersive exploration of extremely detailed surface models on multi-projector light field displays, which give multiple, freely moving, naked-eye viewers the illusion of seeing and manipulating 3D objects with continuous horizontal parallax. Our specialized 3D user interface, dubbed *FOX* (*Focus Sliding Surface*), allows inexperienced users to inspect 3D objects at various scales, integrating panning, rotating, and zooming controls into a single low-degree-of-freedom operation. At the same time, FOX takes into account the requirements for comfortable viewing on the light field display hardware, which has a limited field-of-view and a variable spatial resolution. Specialized multi-resolution structures, embedding a fine-grained, per-patch spatial index within a coarse-grained patch-based mesh structure, are exploited for fast batched I/O, GPU-accelerated rendering, and user-interaction-system-related geometric queries. The capabilities of the system are demonstrated by the interactive inspection of a giga-triangle dataset on a large-scale, 35 MPixel light field display controlled by wired or vision-based devices. Results of a thorough user evaluation, involving quantitative and subjective measurements, are discussed.

Keywords: virtual reality, 3D interaction, input and interaction technologies, visualization

1. Introduction

Museums are evolving into one of the principal components of the leisure and education industry. In recent years, the classical concept of a museum as a room showcasing objects is starting to give way to that of an environment in which the visitor not only observes and contemplates, but also interacts and interprets. The rapid evolution of automatic shape acquisition technologies is making large amounts of sampled 3D data available, especially in the field of cultural heritage where artifacts are nowadays routinely scanned for preservation, study, or presentation. This data availability provides opportunities to provide users with realistic and accurate visual depictions of cultural artifacts that are controlled by real-time navigation/interaction tools. However, a visitor of a museum exhibit should not be expected to be a proficient computer user; and if he is, the museum cannot expect to keep the visitor's attention long enough to train him in the use of a sophisticated user interface.

In this paper, we describe an approach for natural immersive exploration of extremely detailed but topologically simple surface models, such as those acquired by modern 3D scanning technology. Typical examples are 3D reconstruction of statues and other cultural heritage artifacts.

Recent advances in 3D display design make it possible to reproduce natural light fields with high-resolution [1], allowing a modern display to closely reproduce the perceptual quality and the unique aura of a real 3D artifact. Such devices can give multiple, freely placed naked-eye viewers the illusion of seeing and manipulating 3D objects with continuous horizontal parallax. While previous work has demonstrated the possibility of rendering life-like massive models on such displays [2], this display technology raises specific user interface issues which have been, so far, neglected.

The main contributions of this work is a user-interface approach – with implementation – that is specifically designed for the inspection of massive models rendered on advanced light field displays, allowing users to view detailed 3D objects at various scales. This method integrates panning, rotating, and zooming controls into a single low-degree-of-freedom operation, while automatically keeping the user within the optimal display workspace. Our method, called *FOX*, allows the user manipulate the model while adapting the motions to the limited field-of-view and variable spatial resolution of the light field display; taking these factors into account is essential for comfortable viewing on such hardware. Moreover, the interaction method is well suited to a variety of input devices, including vision-based techniques for full hands-free interaction. In order to maintain interactive frame rates with multi-gigabyte models, we employ specialized multi-resolution structures which embed a fine-grained, per-patch spatial index within a coarse-grained patch-based mesh structure. These structures, in addition to being exploited for fast batched I/O and GPU-accelerated rendering, are important for real-time model exploration through fast multi-scale geometric queries.

Email addresses: marton@crs4.it (Fabio Marton), magus@crs4.it (Marco Agus), gobbetti@crs4.it (Enrico Gobbetti), gianni@crs4.it (Giovanni Pintore), mbalsa@crs4.it (Marcos Balsa Rodriguez)

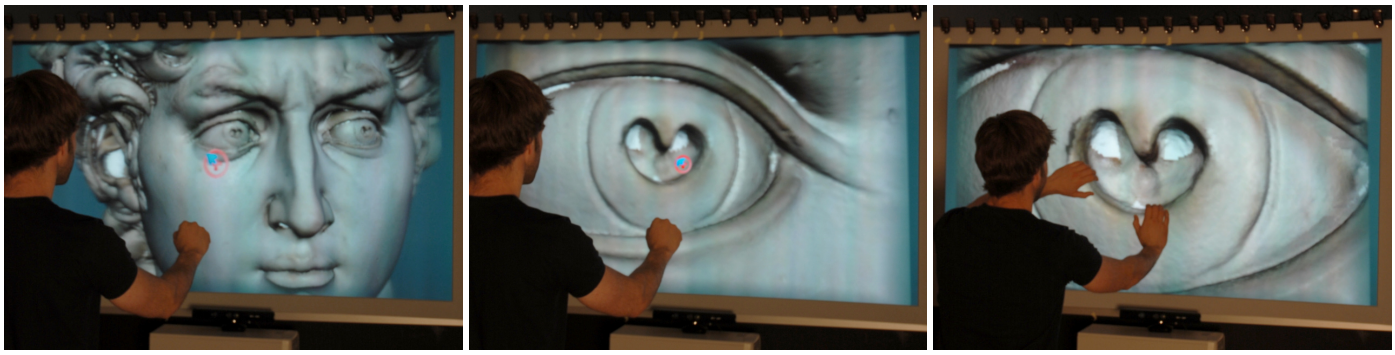

Figure 1: **Natural immersive exploration of the David 0.25mm model (1GTriangle) on a 35MPixel light field display.** Images taken with a hand-held camera. The 3D user interface allows casual users to inspect 3D objects at various scales, integrating panning, rotating, and zooming controls into a single low-degree-of-freedom operation, while taking into account the requirements for comfortable viewing on the light field display hardware. The model appears to be floating in the display workspace, providing correct parallax cues to multiple naked-eye observers.

In this article, we extend the work presented at VRCAI 2011 [3] by adding significant new features to FOX, including an automatic depth adjustment method for maintaining the scene within the comfortable view range of the light field display, and an extensive qualitative and quantitative user evaluation. So, while FOX takes inspiration from previous work, it is particularly motivated/customized by the constraints posed by the display environment. Moreover, we present the first thorough user evaluation of an exploration metaphor in such a context, which we expect will be helpful for future work in this area. Finally, we provide a thorough explanation of system.

Combining the techniques implemented by FOX in a single system is not trivial and represents a substantial advancement of the state-of-the-art. We claim that this is the first system providing controlled navigation in the context of 3D massive model exploration on light field displays. The performance and possibilities of the system are demonstrated by the interactive inspection of a giga-triangle model on a large scale 35MPixel light field display driven by 19 PCs. As demonstrated by our user evaluation, the system can be effectively used with little or no training even by novice users.

2. Related work

A system for the interactive inspection of massive surface models on light field displays requires the application of multiple state-of-the-art techniques in a number of technological areas. In this section we briefly discuss the techniques that most closely relate to ours.

Motion control for virtual exploration. In the context of massive model visualization, users require interactive control to effectively explore the data. Automatic or assisted navigation control has the potential to greatly enhance the interactive experience with large data sets by simplifying navigation – something especially important in the context of virtual museums where novice users are expected and non-negligible training times must be avoided. An approach for assisted navigation is to limit the user's degrees of freedom. For instance, surface orbiting methods constrain the camera to stay in a region around the object and with a specific orientation with respect to the surface [4, 5, 6]. In fact, most of the work in this area is connected to camera motion control (see Christie and Olivier [7] for a survey). Conversely, in this paper we propose an *object motion control* metaphor, combining the advantages of Speed-Dependent Adaptive Zooming [8] and Adaptive Surface Orbiting [4, 9]. In addition, we introduce constraints for comfortable viewing on the light field display and implementing them specifically for massive and detailed models. In particular, we strive to reduce visual discomfort by constraining large portions of the model within the limited depth-of-field of the display. The resulting interface exploits the granularity of the multi-resolution representation to provide a smooth, natural and easy to use exploration tool, able to provide users with fast access to fine details and a compelling model surfing experience. As input, the interface only requires two degrees of freedom and a status button. Thus, the method can be used with many input devices, including standard mice, 3D pointing devices, and computer-vision-based tracking systems.

Supporting massive models. Given the potentially massive size of high-resolution digital models and the wide range of devices at which an interactive renderer and user interaction system have to operate, it is essential for the system to be based on an adaptive level-of-detail (LOD) structure maintained out-of-core. For mesh rendering, state-of-the-art systems achieve maximum performance by shifting the granularity of the representation from triangles to triangle patches [10, 11, 12]. While the coarse-grained approach improves performance for rendering, it is insufficient for fast point queries, which are required by our 3D navigation system for finding anchor positions during object motion. To enable fast point queries we augment each node of the coarse-grained rendering structure with a fine-grained partitioning structure which spatially indexes each of the triangles it contains. The coarse multi-resolution structure is based on a diamond hierarchy [13], similar to the one used in Batched Multi-triangulation [12] and constructed with an Adaptive Tetrapuzzles approach [10]. The fine spatial index structure, by contrast, is based on an axis-aligned bounding box

hierarchy. A similar approach, based on BSP trees, has been proposed by Lauterbach et al. [14] for Interactive Ray Tracing applications.

3D rendering for light field displays. Light field displays provide unrestricted stereoscopic viewing and parallax effects without special glasses or head tracking. They are intrinsically multi-user and can be built by using high-resolution displays or, alternatively, multi-projector systems with parallax barriers or lenticular screens. The light field display hardware employed for this work is manufactured by Holografika (see `www.holografika.com`) and is commercially available. It uses a specially arranged projector array, driven by a cluster of PCs, and a holographic screen. Large, multi-view light field displays require generating multiple images, one for each available perspective. As in other state-of-the art rendering methods for such displays, we exploit multiple center of projection (MCOP) geometries [15] and adaptive sampling [16] to fit with the display geometry and the finite angular resolution of light beams. In addition, we employ a multi-pass rendering method which allows us to implement depth-dependent filtering [17]. For cluster-parallel rendering, we use a sort-first parallel rendering approach with an adaptive out-of-core GPU renderer for each back-end node, rather than using an object-based server-push philosophy as in previous light field display rendering systems [2]. This method reduces server load and supports the use of different refinement strategies for the rendering and control system.

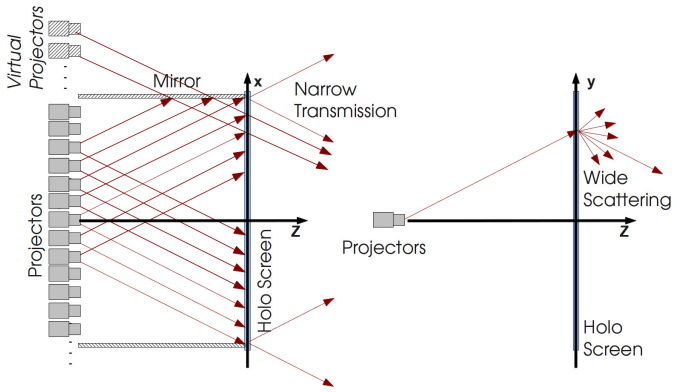

Figure 2: **Light field display concept.** The display is uses a specially arranged projector array, a holographic screen, and side mirrors to increase the field of view. Left: horizontally, the screen is sharply transmissive and maintains separation between views. Right: vertically, the screen scatters widely so the projected image can be viewed from essentially any height.

3. Light field display: concepts and consequences

The light field display employed for this work uses a specially arranged projector array driven by a cluster of PCs and a holographic screen (see Fig. 2 left). The projectors are densely arranged at a fixed, constant distance from a curved (cylindrical section) screen. The projectors cast their respective images onto the holographic screen to create the light field. Mirrors positioned at the sides of the display reflect back onto the screen the light beams that would otherwise be lost, thus creating virtual projectors that increase the display field of view. The holographic screen has a holographically recorded, randomized surface relief structure able to provide controlled angular light divergence: horizontally, the surface is sharply transmissive, to maintain a sub-degree separation between views determined by the beam angular size. Vertically, the screen scatters widely, hence the projected image can be viewed from essentially any height. Thus, this approach creates a display with only a horizontal parallax.

In order to cope with the parallax-only design, we employ a multiple-center-of-projection (MCOP) technique [15, 16] to generate images with good stereo and motion parallax cues. The method is based on the approach of fixing the viewer's height and distance from the screen to those of a virtual observer in order to cope with the horizontal parallax. We assume that the screen is centered at the origin with the y axis in the vertical direction, the x axis pointing to the right, and the z axis pointing out of the screen. Given a virtual observer at $\mathbf{V}$, the ray origin passing through a point $\mathbf{P}$ is then determined by $\mathbf{O} = (E_x + \frac{P_x - E_x}{P_z - E_z}(V_z - E_z), V_y, V_z)$, where $\mathbf{E}$ is the position of the currently considered projector. The ray connecting $\mathbf{O}$ to $\mathbf{P}$ is then used as projection direction to transform the model in normalized projected coordinates. The parameters used for mapping screen pixels to screen 3D points can be determined by automated multi-projector calibration techniques [16].

By appropriately modeling the display geometry, the light beams leaving the various pixels can be made to propagate in specific directions, as if they were emitted from physical objects at fixed spatial locations. Freely moving, naked eye users can thus have the illusion of seeing virtual objects floating in the display workspace. It is important to note that the images of these objects are sharp only near the holographic screen, since the spatial resolution of the display is variable with respect to depth, approximately according to the equation $s(z) = s_0 + 2\|z\| \tan(\frac{\Phi}{2})$, where z is the distance to the holographic screen, and s_0 is the pixel size on the screen surface [16] (see Fig. 3 left). While blurred images are acceptable on the background, far from the viewer, excessive blurring near the viewer leads to discomfort.

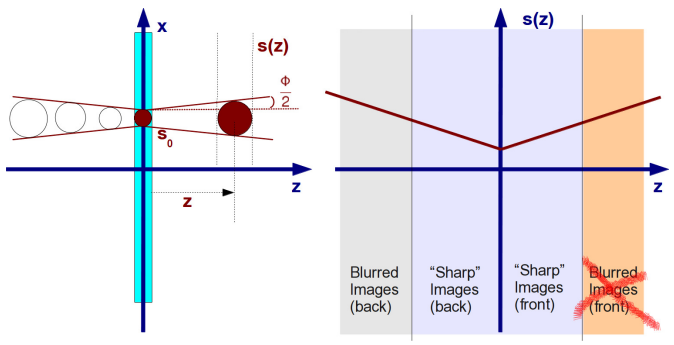

Figure 3: **Light field display spatial resolution.** The spatial resolution of the display varies with the depth. Only the region near the holographic screen is rendered sharply.

Thus, the 3D display and related rendering methods have peculiar characteristics which impose constraints to the interaction and rendering system in order to generate compelling visualizations and reduce rendering artifacts. Specifically, the following characteristics have to be taken into account for the implementation of a natural interactive rendering system for massive models on a light field display:

- the spatial resolution of the display is variable with respect to depth, and objects far from the display's holographic screen appear blurred; thus, points of interest of the objects should be rendered near the screen surface;
- the calibration technique minimizes errors only on the surface of the screen; thus, the effective depth of field of the display is reduced not only because of the diminishing spatial resolution, but also because of the spatially varying calibration accuracy;
- because of the display geometry, the angular field of view is limited and allows presentation of objects only within well defined angular bounds.

Thus, the best viewing experience is obtained when: (a) the scene is kept centered with respect to the screen; (b) the scene remains inside a limited depth range (at least in the front area of the display); and (c) the frequency details of the objects are adapted to the display's spatial accuracy. While (c) can be obtained by suitable rendering methods (see Sec. 5), (a) and (b) are best met by taking special care to position the scene within the display workspace.

4. Focus sliding surface (FOX) interactive navigation metaphor

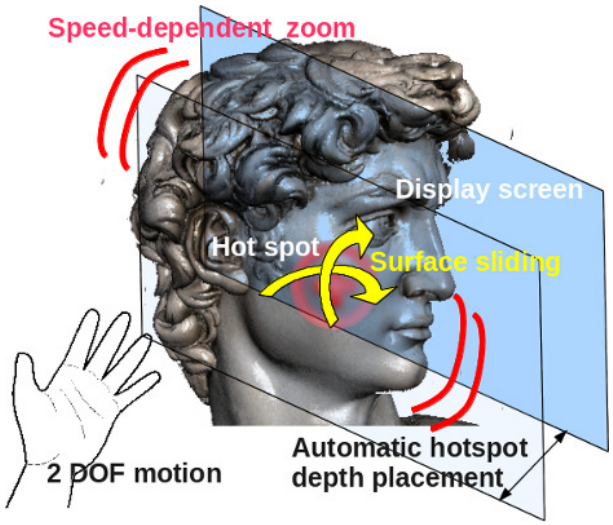

Figure 4: **Focus sliding surface (FOX) navigation metaphor.** The proposed metaphor combines 2 DOF surface sliding which constrains the model to be attached to the display hot spot, speed-dependent automatic zooming and automatic hot spot depth placement. With this metaphor, users can explore the model at various scales, while always maintaining a portion of the object, which becomes the focal point, in the optimal viewing position

In general, natural exploration of 3D objects can be a difficult task for a novel user, even with common 2D displays. Using a light field display further complicates the situation since objects need to stay within a certain depth range to produce good quality images. General interaction metaphors, like rotate, pan and zoom, in addition to not being trivial to master, can easily move the part of the surface of interest out of the display's effective rending volume, thus further increasing the complexity of the navigation task and visual discomfort.

We introduce a 3D user interaction technique which allows casual users to inspect 3D objects of various scales, integrating panning, rotating, and zooming controls into a single low-degree-of-freedom operation, while taking into account the requirements for comfortable viewing on a light field display. The technique is dubbed "focus sliding surface" (FOX). We attempt to use as many constraints as possible to simplify the number of controls needed for interaction. The method does not require learning specialized gestures, and is well adapted to a variety of input devices, including vision-based techniques for full hands-free exploration. The basic idea behind FOX is that navigation actions should move and scale the inspected object so that its surface remains in contact with the display hotspot, placed near the center of the screen, with the local (smoothed) surface plane parallel to the screen, and (optionally) that the object's up direction remains oriented upwards in the real world. In this manner, a user can explore the model at various scales while FOX keeps a portion of the object, which becomes the focal point, in the optimal viewing position. The object is constrained to slide on an anchor point placed near the center of the screen. This approach nicely handles simple convex surfaces, slightly concave surfaces and, through the usage of multi-resolution models for approximating the surface (see later), jumps across gaps or holes. Objects with these kinds of surfaces, possibly with significant protrusions and cavities but nonetheless with relatively simple topologies, correspond well to the typical cultural heritage models (e.g., statues) targeted by our application. Specifically, our FOX interaction metaphor is obtained by composing the following motion primitives:

- translation and rotation, obtained with a two DOF pointer motion, which constrains the model surface to slide on the display hot point;
- speed-dependent automatic zooming, coupled with the user's motion speed, which enables zoom-in (for lower speeds) and zoom-out (for higher speeds) concurrently with surface sliding;
- automatic display hotspot placement, which tunes the depth of the hotspot during interaction in order to maintain the manipulated surface in a good viewing position.

Figure 4 provides a schematic diagram of the components of FOX interaction metaphor.

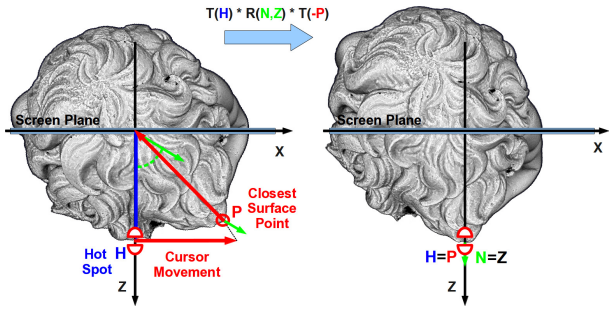

Figure 5: **Constrained panning and rotation.** Left: the red horizontal arrow represents the cursor movement into the plane identified by the hotspot and the front direction. **p** is the closest point of the model surface to this new cursor position; the model will be translated to the origin from here, rotated to transform the point normal (green arrow) into the front direction, then translated to the hotspot. Right: the model transformed by this incremental matrix ($T \times R \times T$), with the pair $\mathbf{p}, \mathbf{n}$ satisfying the hotspot constraint.

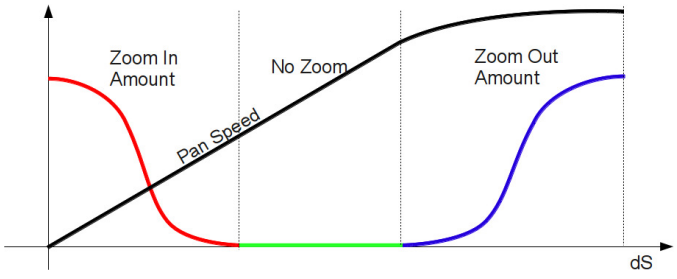

Figure 6: **Zoom and pan.** Zoom and pan as functions of cursor movement. Zoom amount: before the left threshold, the amount of zoom-in is a decreasing smooth-step of the dS; in central part there is no zoom; after the higher threshold, the amount zoom-out is an increasing smooth-step of the dS. Pan speed linearly grows with the cursor movement, until the zoom-out region where it saturates.

4.1. Translations and rotations

Since we constrain the surface to slide on the display hot point, panning and rotation can be specified with only two degrees of freedom. A smooth path can be achieved by first applying user input to the current surface point, moving it in the plane parallel to the display screen. We then search for the closest point and normal on a smoothed version of the object surface (see later). This point and its normal are then transformed to align them with the hotspot and the front direction by an incremental matrix which will update the model placement, as shown in Fig. 5. An additional constraint on the transformation can be introduced to keep the model oriented along its preferential up direction. In order to apply this constraint, the surface normal **n** is projected into the plane orthogonal to the up vector before computing the rotation to avoid changing the vertical axis. It is then averaged with the front direction to smooth out the resulting movement, limiting abrupt rotations due to the model surface roughness. The incremental model transformation $\overline{\delta}$ given by the closest surface position and normal $(\mathbf{p}, \mathbf{n})$ pair is computed by $\overline{\delta} = \mathbf{T}(\mathbf{s_c}) \times \mathbf{R}(\mathbf{n} \rightarrow \mathbf{Z}) \times \mathbf{T}(-\mathbf{p})$, where $\mathbf{s_c}$ is the screen center. The new modeling matrix is then computed by $\overline{\mathbf{M}}_{\mathbf{i+1}} = \overline{\delta} \times \overline{\mathbf{M}}_{\mathbf{i}}$.

4.2. Automatic zooming

We employ speed-dependent automatic zooming to couple the user's rate of motion with the zoom level – the faster the user moves the smaller the object is made (see Fig. 6). The rationale behind this approach is as follows. When the user begins a motion and then stops, he is focusing on something and wants to see it in more detail. Thus, we slowly start to scale up the object, increasing zoom rate with time. On the other hand, when the user moves slowly, we can infer that he is interested in inspecting the region around the current focal position. Thus, we incrementally navigate over the object's surface while remaining at the same scale. Finally, if the user starts moving very fast, he probably wants to quickly reach a new target of interest, thus we scale down the object, making incremental navigation over the object's surface faster. With this approach, both translate, rotate, and zoom can be specified by a single 2D vector input. This 2D vector represents the velocity with which we intend to move the anchored point. As illustrated in Fig. 6, for incremental navigation, the norm of the vector is filtered by the pan speed function, which grows linearly up to the start of the zoom-out state. Then it saturates since zooming-out starts to help panning by reducing the scale of the object. The velocity vector multiplied by the elapsed time between two consecutive steps with a proper scale factor produces the amount of movement to apply to the anchor point.

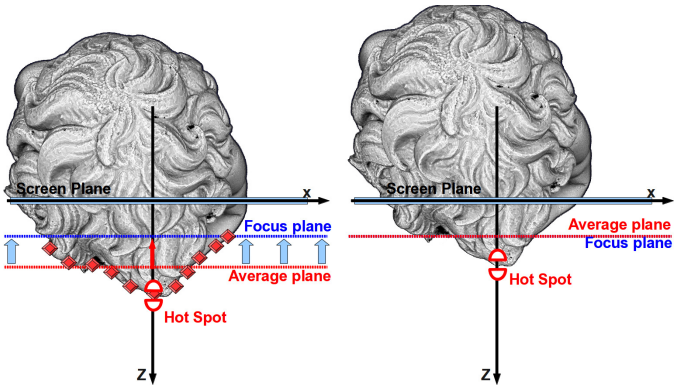

Figure 7: **Automatic hotspot placement.** The depth of hotspot is tuned automatically during interaction to keep the manipulated surface in a good viewing position. To do so, a least square plane of the points in the neighborhood of the hotspot is computed.

4.3. Automatic model depth adjustment

The system always places the model in contact with the display hotspot, which should be at the center of the screen. Another requirement imposed by display characteristics is to keep the surface being manipulated at a good viewing depth. The display achieves its best resolution on its surface ($z = 0$). However, we found that users prefer to have the object slightly protruding from the screen in order to be able to virtually touch it (see Fig. 1 right). Thus, we would like the system to place the surface approximately at a depth H^* a few centimeters out of the screen. Simply placing the hotspot at a fixed depth H^*

is not sufficient, since the model can have complex asymmetric shapes around the hotspot.

To implement this depth adjustment feature, we developed a feedback correction scheme that automatically updates the model's position (and thus the hotspot depth) during interaction. For each user interaction step, our depth correction method extracts a coarse approximation of the surface in contact with the display hotspot (see Fig. 7 left). This coarse point cloud, $(P_0, P_i, ..., P_N)$, quickly extracted from our multi-resolution model representation (see Sec. 5), is then used to compute a weighted average depth s_z of the surface in the neighborhood of the hotspot $H = (h_x, h_y, h_z)$:

$$s_z = \frac{\sum_i w^{(i)} P_z^{(i)}}{\sum_i w^{(i)}} \quad (1)$$

where the weight of each point $w^{(i)} = \Phi\left(\frac{\|(p_x^{(i)}, p_y^{(i)}) - (h_x, h_y)\|_2}{R}\right)$ is computed by a smooth, radially decreasing weight function for which we use the following compactly supported polynomial: $\Phi(x) = \max(0, (1 - x^2))^4$. Since the function has local support, only points within an xy-distance of R from the hotspot contribute to determining the desired visible model surface depth s_z. For the purposes of this work, R was set to half the height of the display.

At this point, the amount of depth correction theoretically required is the difference between the average depth s_z and the comfortable depth H^* a few centimeters out of the screen (see Fig. 7 right).

In order to avoid abrupt changes in depth due to any surface discontinuities in the model and to reduce high-frequency vibrations, the depth correction is temporally low-pass filtered by applying at each frame only a fraction λ of the full displacement (in our implementation $\lambda = 50\%$ adequately cut all undesired vibrations while still effectively correcting the scene depth). The overall model (and thus also the surface hotspot) is thus translated at each frame by an amount $\lambda(s_z - H)$ in the z direction.

With this scheme, FOX is able to automatically keep the position of the approximated surface in a comfortable viewing position (close to the focal depth). Points near the hotspot are therefore rendered at a good resolution and, since they are placed out of the screen, the surface can also be "touched" by users, increasing the quality of experience. In our tests, a sampling rate twenty times coarser with respect to the original surface employed for rendering resulted to be computationally effective and sufficiently accurate for automatic model depth adjustement.

4.4. Input mapping

We can use a variety of devices to capture user input since FOX only requires simple a 2D vector and a binary state (pressed/released). One simple approach is to use a single button 2D (or 3D) mouse and use mouse drag to specify motion. Motion is applied when the button is pressed, and the velocity vector is computed by the distance from the position at button press time to the current position. In the case of a 3D mouse, the motion is projected to the plane parallel to the display surface.

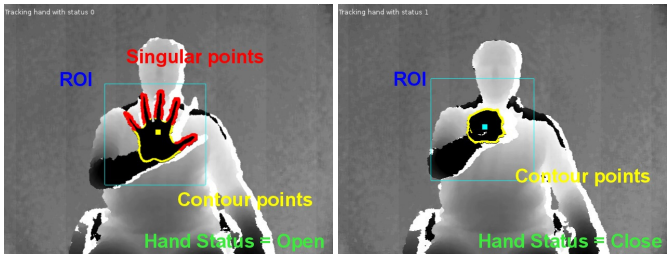

Figure 8: **Hand tracking using depth sensors.** Open or closed hand state is detected by estimating contour curvature of the hand as to identify singular points.

Another input approach, especially ideal in a museum setting, is hands-free object manipulation. We have implemented a tracker using a Kinect depth sensor to recognize hand movements. At each frame from the sensor, the hand point cloud in world coordinates is used to compute the cursor position and the hand state (open/closed). The cursor position is simply the centroid of the hand point cloud, while the hand status is recognized by analysis of contour curvature. Evaluation of curvature information on blob contours point has been demonstrated to be a robust way to detect fingertips [18]. In FOX, we apply this method to depth images from the Kinect and detect critical points by using an eigen analysis of the covariance matrix of a local neighborhood. A Region Of Interest (ROI) is employed to continuously track the hand point cloud in world coordinates. A prediction filter is also applied to the ROI, as to compensate abrupt motion changes that could compromise tracking. Singular points – i.e., points whose curvature values exceed a given threshold – are then used to identify the hand fingers [19]. Although this method is robust enough to track finger motion, we are only interested in recognizing the status of the hand (closed or open); we obtain the status by thresholding the number of singular points (see figure 8).

4.5. Cursor glyphs

The interaction experience is improved by providing visual feedback indicating the current interaction mode through cursor glyphs (see Fig. 9).

An icon, drawn at the anchored point position, provides a visual representation of the function presented in Fig. 6. A red circle containing a plus sign indicates zoom-in, while zoom-out is indicated by a blue circle containing a minus sign. When panning, an arrow showing the direction of movement is superimposed on the zoom glyph. All glyph sizes are proportional to the norm of the represented quantities.

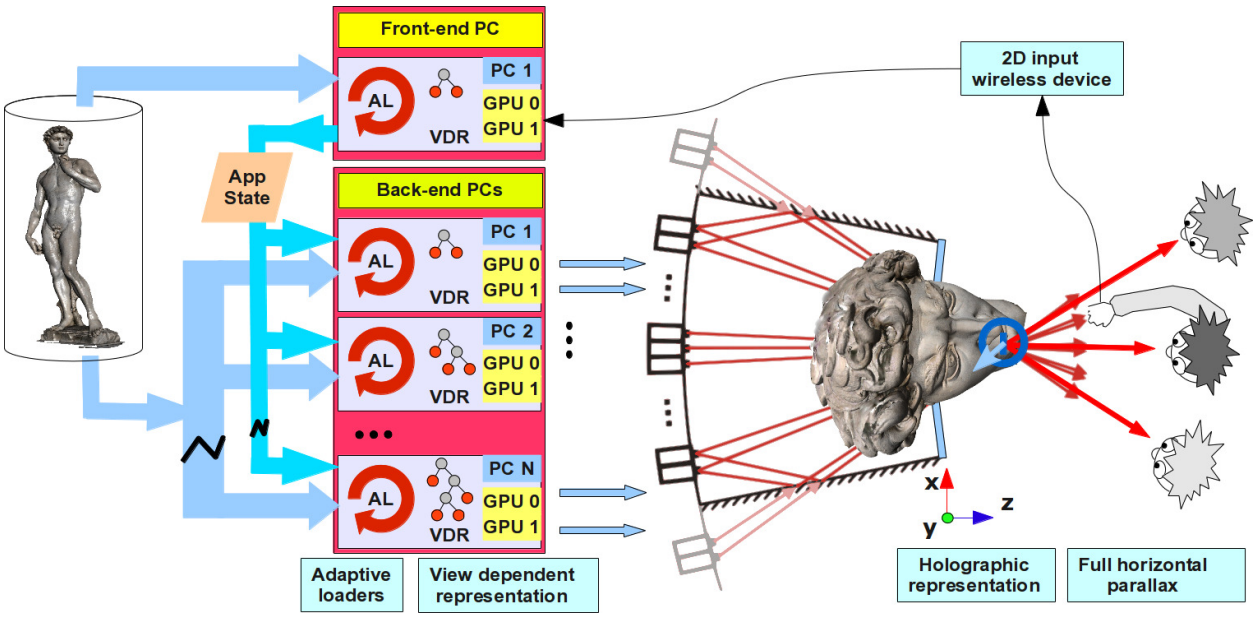

Figure 10: **Virtual environment architecture.** A user moves the model using a 2D device whose input is elaborated by the front-end PC, which computes new modeling transformation and sends them to the rendering back-ends. Back-end nodes update their view-dependent representations by asynchronously fetching data from the out-of-core database. Multiple users perceive the model as floating in space in the new updated position.

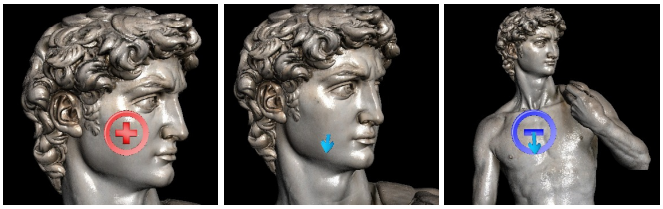

Figure 9: **Cursor glyphs.** In the left image, a red cross inside a circle indicating zoom-in. In the central image, an arrow shows the direction of movement, while the symbol in the right image indicates zoom-out with panning.

5. Handling massive models

Our integrated system has to allow multiple naked-eye users to see detailed giga-triangle models floating in space while quickly responding to the actions performed through the FOX interface. Given the size of the model, adaptive out-of-core structures must be used both for rendering the models and for the geometric queries required by our interaction paradigm. These structures are integrated within a parallel system that drives the multi-projector display.

5.1. Overall parallel system architecture

As is the case with other multi-screen displays, we use a distributed image generation system implemented on a cluster, with a front-end PC coordinating many rendering back-end PCs. The overall system architecture is illustrated in Fig. 10.

The front-end PC is connected to one or more input devices for the FOX user interface and manages model motion by delivering to the back-end PCs the current model position, orientation, scale and rendering parameters. An adaptive loader is used within the front-end PC to maintain in core only the part of the surface required for the geometric queries (see Sec. 5.2).

The rendering system uses a sort-first parallel rendering approach, in which each back-end PC is responsible only for the images associated with its connected projectors. Even though in principle it is possible to use, for maximum performance, one PC per projector, benefit/cost analysis leads to a configuration in which each PC drives multiple projectors through multiple graphics boards. Each back-end process controls a portion of the frame buffer, where it renders the multi-resolution model, adaptively loaded from out-of-core, and some visual feedback for the motion control. Unlike previous light field display rendering systems [2], we do not push data from the front-end to rendering nodes, but let each back-end node manage an adaptively refined version of the model. A multi-pass rendering approach is used in which a first geometry pass uses vertex shaders that implement the display-specific projection, and a series of full-image passes implemented by fragment shaders realize deferred shading and filter the image to produce the required visual effects. In particular, we apply a depth-dependent blur to adapt the frequency content of the scene to the display's spatial resolution, reducing aliasing artifacts. The filtering is implemented by applying an image-based, two-pass, depth-of-field method [20], with a circle of confusion corresponding to the depth-dependent spatial resolution of the light field display. Specifically, we employ a post-processing pixel shader which takes as input the original image and a downsampled and pre-blurred version of the same image, and uses a variable size kernel approximating the circle of confusion to blend between the original and the pre-blurred image. The filtering reduces aliasing and extends the usable depth range of the light field display for model backgrounds.

5.2. Multi-resolution structure

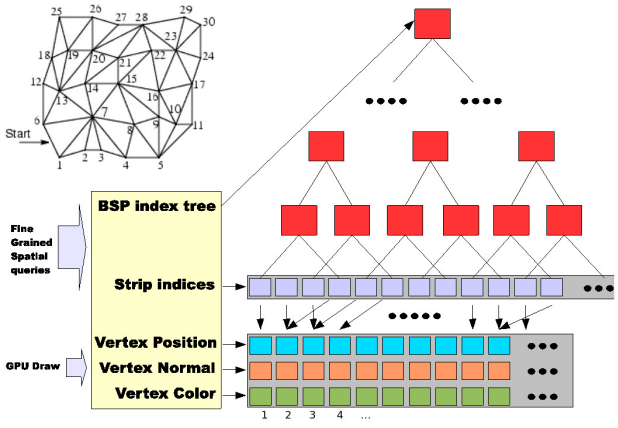

Figure 11: **Patch structure.** Each patch is coarse-grained, represented as a generalized triangle strip, but also contains a fine-grained spatial index.

In order to be able to interactively render and query massive models, we employ a multi-resolution structure based on a modification of the Adaptive Tetra Puzzles [10] method, which allows us to efficiently select nearest neighbor points at different levels of resolution, in addition to supporting adaptive rendering. The structure is used both in the front-end (for geometric queries) and in the back-ends (for rendering).

The underlying idea of the Adaptive Tetra Puzzles method is to adopt a patch-based data structure constructed by spatial decomposition, from which view-dependent conforming mesh representations can be efficiently extracted by combining precomputed patches arranged in a DAG. Since each patch is itself a mesh composed of a few thousand triangles, the multi-resolution extraction cost is amortized over many graphics primitives, and CPU/GPU communication can be optimized to fully exploit the complex memory hierarchy of modern graphics platforms. In addition, to accelerate spatial queries we augment this coarse-grained structure with a per-patch spatial index that organizes individual triangles in a patch triangle strip.

The patch-size granularity of such a method is efficient enough to ensure interactive and high-quality rendering and is effective for batched I/O operation, but is too coarse for the spatial queries. For this reason, we introduce a fine-grained BSP structure which is kept within each coarse-grained node in order to spatially index individual triangles accelerating the spatial search (see Fig. 11). To reduce storage overhead, this BSP structure is constructed on-the-fly at patch loading time using a fast recursive split procedure which exploits the fact that each node contains triangles arranged in a spatially coherent manner. Given N triangles in a patch, these are organized in a single generalized cache-coherent triangle strip of $M \geq N + 2$ vertices. We recursively split each strip at the median edge in order to define a balanced tree on the strip. At each step, we record the left and right bounding boxes. This procedure defines a balanced spatial bounding box tree on the patch mesh, such that only the two bounding boxes must be stored. At runtime, the tree can be used for search queries implemented with top-down descents. The model's closest point **p** and its normal **n** are computed by extracting the k-nearest neighbor points on the model's surface ($k = 64$ in our current implementation), which are in turn blended together with Gaussian weights that fall out with the distance from the search point, with a standard deviation equal to the median of the distances of the neighborhood points. By pre-computing a geometric simplification of the multi-resolution model and blending multiple points our system smoothly handles non-trivial models.

Our parallel system uses, in core, two adaptively refined versions of the model: one for the rendering (in the back-ends), and one for interaction support (in the front-end).

In the rendering back-ends, error computation for the level of detail selection is different from what is done for standard displays, since we must consider the geometric properties of the display screen (see Fig. 2). In order to select the appropriate level of detail, we compute the nearest distance z_{min} between the current node and the display screen, and decide to refine the node if its average edge length is bigger than the local spatial display resolution $s(z_{min})$. Since the level of detail selection purely depends on distances to the display screen, and it is independent of any specific projector parameters, all back-ends converge to the same representation without the need to exchange information, and the overall image is fully continuous.

On the other hand, nearest neighbor queries in the front-end use a graph cut in which the level of detail is determined by a radial function centered at the search hot spot and decreasing with the distance to the center. This approach allows us to perform filtered spatial queries consistently with the current viewing scale.

6. Implementation and evaluation

Our system has been implemented on Linux using OpenGL and GLSL. Our 3D display is capable of visualizing 35Mpixels by composing images generated by 72 SVGA LED commodity projectors illuminating a 160×90cm holographic screen. The display provides continuous horizontal parallax within a 50° horizontal field-of-view with 0.8° angular accuracy. The pixel size on the screen surface is 1.5mm. The rendering back-end is currently running on an array of 18 Athlon64 3300+ Linux PCs equipped with two NVIDIA 8800GTS 640MB (G80 GPU) graphics boards running in twin-view mode. Therefore, each back-end PC generates $4 \times 800 \times 600$ pixels using two OpenGL graphics boards based on an old G80 chip. Front-end and back-end nodes are connected through Gigabit Ethernet and communicate through OpenMPI 1.2.6.

We have tested our system with a variety of high-resolution models and settings. In this paper, we discuss the results obtained with the inspection of the *David*0.25*mm* model, composed of 970M triangles. The model can be considered a good test case for the method since it has a non-trivial topology and

it can be inspected at a variety of scales. For instance, viewing the full figure of the model requires fitting the 5.17 meter marble statue within the 90cm display height, while looking at the details of an eye, clearly visible at the scan resolution, requires increasing scale by a hundred zoom levels.

It is obviously impossible to fully convey the impression provided by our interactive 3D system on paper or video. As a simple illustration of our system's current status and capabilities we recorded interactive sessions using a hand-held video camera. Representative video frames are shown in Fig. 12. Please refer to the accompanying video for further results.

6.1. Performance

Our multi-resolution system is capable of sustaining interactive performance with an accuracy of 1 triangle/view-dependent pixel for the rendering back-ends and 1 triangle/cm on the front-end for k-nearest-neighbor searches near the hotspot. The frame rate of typical inspection sequences varies between 15 Hz for extreme close-up views to over 60 Hz for overall views. We tested our interactive system by asking users to perform a variety of inspection tasks, including looking at the back of the object, rapidly moving from top to bottom, and closely inspecting several very distant details.

6.2. User evaluation

In order to assess the FOX manipulation metaphor we performed an extensive user evaluation, involving quantitative and subjective measurements based on interactive exploration tasks. The main goal of the evaluation was to assess whether the proposed interaction metaphor 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 manipulation techniques and controlling devices, we do not try to compare all the possible navigation systems, but focus on providing qualitative and quantitative measures on FOX, as well as comparing it with a representative approach.

To this end, we evaluated FOX and compared it with a standard 5-DOF object-in-hand manipulation metaphor [21], using a precise inertial ultra-sonic 6-DOF tracking device (Intersense IS-900 3D Mouse). The 5-DOF object-in-hand rigidly attaches the object to the user's hand when a button is pressed. Two other buttons are used for zoom-in and zoom-out. The 5-DOF object-in-hand technique was chosen because it is the interaction metaphor most commonly employed in virtual environments and because it is easy and immediate to learn.

Furthermore, we wanted to evaluate whether a practical free hand implementation of our metaphor is reasonable, and how the reduced tracking accuracy with respect to other 3D sensors (e.g., ultrasound, magnetic or optical tracking systems) impacts on user experience. Thus, we compared the performance of FOX with a free hand Kinect control and the Intersense 3D mouse.

Methodology. The evaluation methodology considered both quantitative and qualitative criteria to assess performance in the user tasks as well as the overall user experience and satisfaction. For quantitative evaluation, users were asked to perform guided and exploratory manipulation tasks. Their performance was evaluated with respect to task completion time and 3D image quality maintained during interaction. We wanted to propose to participants a difficult but compelling virtual interaction scenario, similar to what they could find in a virtual museum exhibit. Specifically, the experiment consisted in letting users try the two different manipulation metaphors (FOX and 5-DOF) in the context of two different interaction tasks: a guided target-reaching task, where participants were asked to manipulate the model until reaching a given specified position with a specified zoom level, and an exploring task, where users were asked to manipulate the model until they found a small sphere target randomly placed on the model's surface. For comparing device performance, users were asked to perform guided reaching tasks, by employing FOX metaphor with both test devices. With respect to the subjective qualitative evaluation, participants were asked to fill a questionnaire comparing the performance of the two metaphors by indicating a score from 0 (very weak) to 4 (very strong) with respect to the following characteristics: ease of learning, ease of reaching desired positions, and perceived 3D image quality. Users were also asked to score the two devices' performance with respect to the ease of learning and ease of reaching the desired position. Finally, participants were asked to indicate their preferred metaphor and device.

Participants. The evaluation procedure involved 33 participants (24 males and 9 females), which were subdivided into two categories: 26 novice and 7 experts, according to their experience with 3D virtual reality or videogames involving 3D tracking interfaces (such as Nintendo Wii or Microsoft Kinect). Ages ranged between 20 and 58, with average 38 years (standard deviation 7 years). Only one participant was left-handed. All subjects had normal or corrected to normal vision.

Procedure. Participants completed four successive blocks of trials. The 3D docking tasks performed are similar to what has been proposed by Zhai et al. [22] and more recently by Martinet et al. [23] to evaluate speed and precision for object positioning in 3D space. The first two blocks consisted of 5 guided "reaching position" tasks, to be performed by employing the Intersense 3D mouse with FOX and 5-DOF. The initial metaphor was randomly chosen as to avoid any potential bias due to training. Targets were indicated by red spheres lying on the David model and users were verbally informed of the position of each target before starting any task. The goal was to reach the specified target in the as quickly as possible by manipulating the model in order to drive the red sphere to fully include a centered transparent white sphere placed at the screen hot spot. When the target was reached, the sphere turned green (see accompanying video and Fig. 13). The targets were posi-

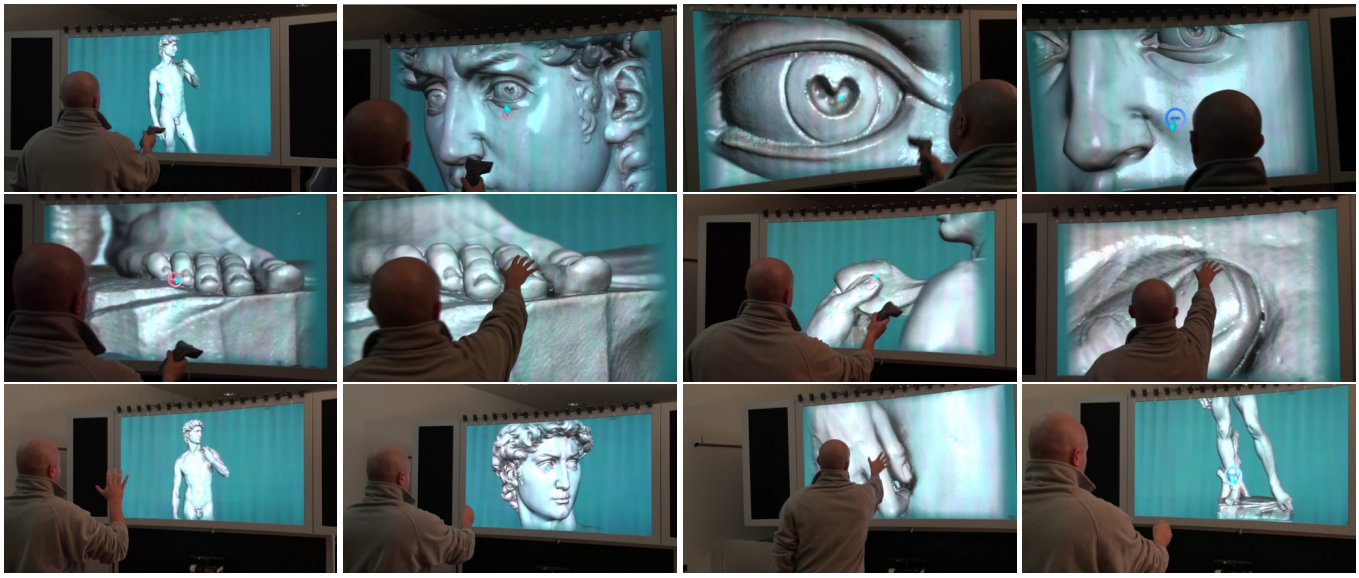

Figure 12: **Live capture.** Representative frames recorded using a hand-held video camera freely moving in the camera and display workspace.

Figure 13: **User evaluation: manipulation task.** Users tested both the FOX and 5-DOF metaphors with an Intersense 3D mouse, and the FOX metaphor with the Kinect hands-free device. Targets were red spheres lying on the model, and the user's task consisted in moving the object to make the target fully include a transparent white sphere at the screen hot spot.

tioned in the following parts of the statue: belly button, behind the neck, left foot, right hand, and left hand. Users had to manipulate the model from the previous target position to the successive one, and some of the target positions were designed in order to force users to follow difficult paths when using the FOX metaphor (e.g., from the right hand to the left hand, by sliding through the arm). The third block of tasks consisted in reaching the same five targets by employing the FOX metaphor with our Kinect-based tracker. Finally, the fourth block consisted of 5 exploring tasks where subjects were asked to find 5 hidden targets. In this case, spheres were drawn with the same color of the statue and no indications of their position were provided to subjects. For this exploration experiment only a metaphor was used for each subject and it was chosen according to the results obtained in the previous tasks (generally subjects were asked to choose according to their preference). The starting position was the same for all targets and the presentation order was randomly shuffled for the two metaphors considered. The times needed for reaching each target were measured and recorded. When subjects were not able to complete a task, we considered a maximum task completion time (60 seconds for guided manipulation, and 120 seconds for exploratory manipulation). We also recorded an estimation of the average 3D image degradation as well as its standard deviation during each task. For this evaluation criteria, we used as metric the normalized integral of the square of the depth of each visible point of the statue, as drawn from the central projector and from the two extreme left and right projectors of the light field display:

$$Q = \sum_i (z_{Ci}^2 + z_{Li}^2 + z_{Ri}^2) \qquad (2)$$

This value provided a clear indication of the distribution of the mass of the model with respect to the light field screen and can be considered as a good estimate of the perceived blurriness.

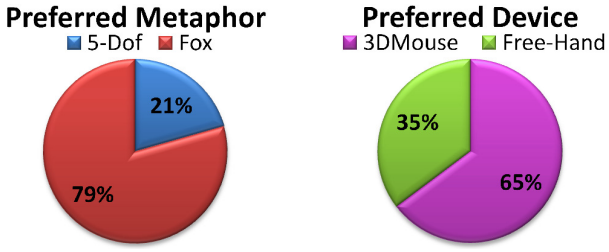

Figure 15: **User preferences** Left: pie chart showing the users' preferred manipulation metaphor between our FOX and classical 5-DOF object-in-hand manipulation metaphors. Right: pie chart showing the users' preferred interface between Intersense 3D mouse and free-hand Kinect-based devices, both tested with FOX metaphor.

Quantitative analysis. We conducted an extensive analysis of variance (ANOVA) of quantitative results in order to find the

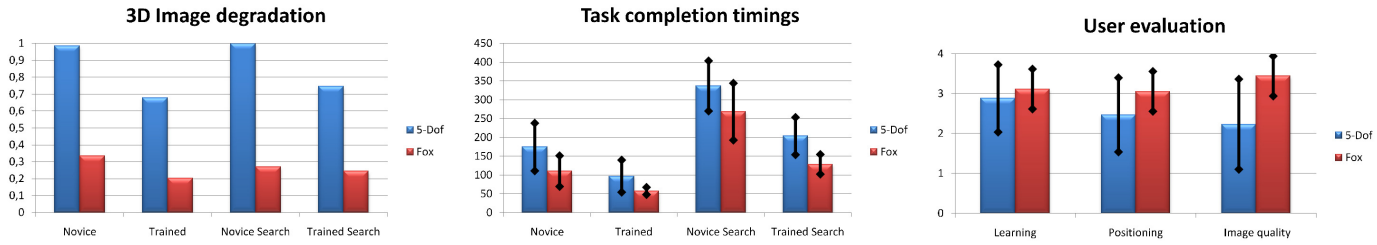

Figure 14: **Test evaluation.** Left: normalized 3D Image degradation for guided manipulation tasks (novice/trained) and exploration tasks (novice/trained). 5-DOF and FOX interfaces are respectively represented by the blue and red color. Center: total task completion timings for guided manipulation tasks (novice/trained) and exploration tasks (novice/trained). Right: users' qualitative evaluation comparing ease of learning, ease of positioning and 3D image quality for 5-DOF and FOX manipulation metaphors.

possible effects affecting the completion times and the image degradation. As possible effects, we investigated the manipulation metaphor (5-DOF and FOX), for both the guided manipulation tasks and for the hidden target searching tasks, and the device employed for guided manipulation tasks performed with FOX. The goal of the analysis was to compare the performance of the different manipulation metaphors and the performance of the different devices. Considering the results of guided manipulation tasks, the main significant effect was found in the metaphor: in fact, for novice users we had $F_{1,52} = 19.497$ with $p < 0.0001$ for total completion time and $F_{1,52} = 41.526$ with $p < 10^{-7}$ for image degradation, while for trained users we had $F_{1,12} = 4.9292$ with $p < 0.05$ for total completion time and $F_{1,12} = 8.4796$ with $p < 0.013$ for degradation. This fact indicates that, especially for novice users, the exploration task was much easier with FOX metaphor and the overall 3D image quality was sensibly better. As expected, this difference was greatly reduced when tasks were performed by trained users, since 5-DOF metaphor provides a greater manipulation control and subjects tend to avoid uncomfortable and blurred model configurations. Moreover, novice users were unable to complete the task within the allocated time in 14% of the trials using the 5-DOF interface, and in 4.8% of the trials using FOX. Expert users always succeeded in both cases. With respect to the device performance, for novice users the Kinect interface resulted in the worse performance when compared to the 3D mouse ($F_{1,42} = 5.8128, p = 0.02$ for total completion time and $F_{1,42} = 5.8418$ with $p = 0.02$ for image degradation), while the device effect was not particularly significant for trained users ($F_{1,14} = 3.078$ with $p = 0.1$ for total completion time and $F_{1,14} = 1.8486$ with $p = 0.2$ for image degradation). Moreover, novice users were unable to complete the task within the allocated time in 8% of the trials using the Kinect while only in 4.8% of the trials using the 3D mouse. These results indicate that the usage of a less precise free-hand device considerably affected the performances of novice users, but not the performances of trained users. Expert users always succeeded using both configurations. Fig. 14 shows the average values of image degradation and total completion times, subdivided for novice and trained users, for the guided manipulation task (left) and for the hidden targets searching task (right). It appears evident, as also highlighted by the ANOVA, that better performances were obtained with the FOX metaphor especially by novice users. These better performance is mostly attributable to the automatic depth correction feedback, which keeps most of the scene inside the comfortable viewing range. Free-hand manipulation results, not present in the graphs, are just slightly worse than the ones for the FOX interface with the 3D mouse. The proof-of-concept vision-based interface appears to work well enough to permit full free-hand interaction. However, users sometimes experience delays in the beginning/ending of navigation, as well as spurious interruptions, because of the imprecision in reliably detecting opening/closing hand events. Nevertheless, the use of the free-hand device does not degrade performance excessively, especially for trained users.

Qualitative analysis. With respect to qualitative results, an analysis of variance of the responses of the subjects indicated that there was no significant effect in the ease of learning ($F_{1,64} = 0.945, p = 0.335$), while there was a main significant effect in the ease of positioning ($F_{1,64} = 7.2746, p = 0.009$) and in the 3D image quality perception ($F_{1,64} = 24.7, p < 10^{-5}$). These results provide evidence that users found it easy to learn both metaphors, but they perceived that with FOX they could more easily reach targets, and also the felt that 3D image quality was considered superior. Average values, together with standard deviations, are show in Fig. 14 right, and highlight the perceived differences between the two metaphors. Finally, an analysis of variance on the subjective responses comparing the devices revealed that there was a significant effect in the ease of learning ($F_{1,52} = 3.609, p = 0.063$) and a main significant effect in the ease of positioning ($F_{1,52} = 15.716, p = 0.0002$). This fact indicates that subjects felt more comfortable when using the Intersense 3D mouse. We think that it is due to the greater precision of Intersense device and to the workspace limitations of the free hand manipulation system. Finally, Fig. 15 shows user preferences with respect to the metaphor and to the device. It appears evident that FOX metaphors sounds strongly appealing for novice users while 5-DOF manipulation is more complex for naïve users.

7. Conclusions and future work

We have presented an interactive system for intuitive exploration of extremely detailed surface models, which appear float-

ing in space to multiple freely moving, naked-eye viewers in a room-sized workspace.

Our cluster-parallel system achieves interactive performance for multi-gigabyte sized models, and its 3D user interface allows casual users to inspect 3D objects at various scales, integrating panning, rotating, and zooming controls into a single low-degree-of-freedom operation, while taking into account the requirements for comfortable viewing on light field displays. The resulting virtual environment, which combines ease of use with high representation fidelity, appears well suited for creative installations at exhibition centers. The low-DOF interaction method is well adapted to a variety of input devices, including vision-based techniques for full hands-free interaction.

Our evaluation points out that the navigation interface appears to be reasonably intuitive even to casual users, which quickly understand how to manipulate the object after a very short trial and error period. The simple markerless 3D tracker is well accepted, even if a more reliable markerless hand tracking system remains an important area for future research. Another possibility worth looking at is the flexible switching between metaphors, e.g., for easier handling of models with many disconnected components.

Our current work is concentrating on improving our proof-of-concept vision-based tracking system to allow it to handle multiple simultaneous users while providing more reliable input. In addition, we are in the process of complementing the application of the FOX method with orthogonal rendering techniques for retargeting 3D content to the display workspace by adaptively warping the 3D model shape. Preliminary results have been recently presented [24]. We are also planning to perform user studies in a real museum setting.

Acknowledgments. We thank Luca Pireddu for his helpful comments and suggestions. This work is partially supported by the EU FP7 Program under the DIVA project (290277). We also acknowledge the contribution of Sardinian Regional Authorities. The David dataset is courtesy of the Digital Michelangelo Project.

References

[1] Agocs T, Balogh T, Forgacs T, Bettio F, Gobbetti E, Zanetti G. A large scale interactive holographic display. In: Proc. IEEE VR Workshop on Emerging Display Technologies. 2006,.

[2] Bettio F, Gobbetti E, Marton F, Pintore G. Scalable rendering of massive triangle meshes on light field displays. Computers & Graphics 2008;32(1):55–64.

[3] Marton F, Agus M, Pintore G, Gobbetti E. FOX: The Focus Sliding Surface metaphor for natural exploration of massive models on large-scale light field displays. In: Proc. VRCAI. 2011, p. 83–90.

[4] Khan A, Komalo B, Stam J, Fitzmaurice G, Kurtenbach G. Hovercam: interactive 3d navigation for proximal object inspection. In: Proc. I3D. 2005, p. 73–80.

[5] Burtnyk N, Khan A, Fitzmaurice G, Kurtenbach G. ShowMotion: camera motion based 3D design review. In: Proc. I3D. 2006, p. 167–74.

[6] Burtnyk N, Khan A, Fitzmaurice G, Balakrishnan R, Kurtenbach G. Stylecam: interactive stylized 3d navigation using integrated spatial & temporal controls. In: Proc. UIST. 2002, p. 101–10.

[7] Christie M, Olivier P. Camera control in computer graphics: models, techniques and applications. In: ACM SIGGRAPH ASIA Courses. 2009, p. 1–197.

[8] Igarashi T, Hinckley K. Speed-dependent automatic zooming for browsing large documents. In: Proc. UIST. 2000, p. 139–48.

[9] McCrae J, Mordatch I, Glueck M, Khan A. Multiscale 3D navigation. In: Proc. I3D. 2009, p. 7–14.

[10] Cignoni P, Ganovelli F, Gobbetti E, Marton F, Ponchio F, Scopigno R. Adaptive tetrapuzzles: efficient out-of-core construction and visualization of gigantic multiresolution polygonal models. ACM Trans Graph 2004;23(3):796–803.

[11] Yoon SE, Salomon B, Gayle R, Manocha D. Quick-VDR: Interactive View-Dependent Rendering of Massive Models. In: Proceedings of IEEE Visualization 2004. 2004, p. 131–8.

[12] Cignoni P, Ganovelli F, Gobbetti E, Marton F, Ponchio F, Scopigno R. Batched multi triangulation. In: Proc. IEEE Visualization. 2005, p. 207–14.

[13] Weiss K, De Floriani L. Simplex and diamond hierarchies: Models and applications. In: Eurographics State of the Art Reports. 2010, p. 113–36.

[14] Lauterbach C, Yoon SE, Manocha D. Ray-strips: A compact mesh representation for interactive ray tracing. In: IEEE/EG Symposium on Interactive Ray Tracing. 2007, p. 19–26.

[15] Jones A, McDowall I, Yamada H, Bolas MT, Debevec PE. Rendering for an interactive 360 degree light field display. ACM Trans Graph 2007;26(3):40–.

[16] Agus M, Gobbetti E, Guitián JAI, Marton F, Pintore G. GPU accelerated direct volume rendering on an interactive light field display. Computer Graphics Forum 2008;27(2):231–40.

[17] Zwicker M, Vetro A, Yea S, Matusik W, Pfister H, Durand F. Resampling, antialiasing, and compression in multiview 3-D displays. IEEE Signal Processing Magazine 2007;24(6):88–96.

[18] Argyros AA, Lourakis MIA. Vision-based interpretation of hand gestures for remote control of a computer mouse. In: In Computer Vision in Human-Computer Interaction. Springer-Verlag; 2006, p. 40–51.

[19] Pauly M, Gross M, Kobbelt LP. Efficient simplification of point-sampled surfaces. In: Proceedings of the conference on Visualization '02. VIS '02; Washington, DC, USA; 2002, p. 163–70.

[20] Zhou T, Chen J, Pullen M. Accurate depth of field simulation in real time. Computer Graphics Forum 2007;26(1):15–23.

[21] Ware C. Using hand position for virtual object placement. Vis Comput 1990;6:245–53.

[22] Zhai S, Milgram P. Quantifying coordination in multiple dof movement and its application to evaluating 6 dof input devices. In: Proc. CHI. 1998, p. 320–7.

[23] Martinet A, Casiez G, Grisoni L. 3d positioning techniques for multi-touch displays. In: Proc. VRST. 2009, p. 227–8.

[24] Agus M, Pintore G, Marton F, Gobbetti E, Zorcolo A. Visual enhancements for improved interactive rendering on light field displays. In: Eurographics Italian Chapter Conference. 2011, p. 1–7.