FOX: The Focus Sliding Surface Metaphor for Natural Exploration of Massive Models on Large-scale Light Field Displays

Fabio Marton* CRS4
Marco Agus CRS4
Giovanni Pintore CRS4
Enrico Gobbetti CRS4

Abstract

We report on a virtual environment for natural immersive exploration of extremely detailed surface models on light field displays. Our specialized 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 a light field display hardware. Specialized multiresolution 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 35MPixel light field display controlled by wired or vision-based devices.

CR Categories: I.3.1 [Computer graphics]: Hardware architecture—Three-dimensional displays I.3.6 [Computer graphics]: Methodologies and techniques—Interaction techniques I.3.7 [Computer graphics]: Three-dimensional graphics and realism—Virtual reality

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 the last few years, the classical concept of museum space as a room containing showcases full of objects is starting to give way to that of an environment in which the visitor not only reads 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. Users need to be provided of realistic and accurate visual representations of such data controlled by real-time navigation/interaction tools. However, the public in a museum does not need to be familiar with the use of computers, and even if it is, the museum cannot afford wasting a user's precious time of attention in training her in the use of a certain user interface.

In this paper, we report on 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 demonstrate that high resolution 3D display technology able to reproduce natural light fields is practically achievable [Agocs et al. 2006], making it possible to closely reproduce the perceptual quality and the unique aura of a real 3D artifact. Such devices give multiple freely moving 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 [Bettio et al. 2008], this display technology raises specific user interface issues which have been, so far, neglected.

Our main contributions are methods and systems, specifically designed for massive models rendered on such a light field display, which allow users to inspect detailed 3D objects at various scales, integrating panning, rotating, and zooming controls into a single low-degree-of-freedom operation, while automatically maintaining them within the optimal display workspace. Our method, dubbed *FOX* (*focus sliding surface*), takes into account the requirements for comfortable viewing on light field display hardware, which has a limited field-of-view and a variable spatial resolution. Moreover, the interaction method is well adapted to a variety of input devices, including vision-based techniques for fully unencumbered interaction. In order to maintain interactive rates for multi-gigabyte models, we employ specialized multiresolution structures, which embed a fine-grained per-patch spatial index within a coarse-grained patch-based mesh structure. These structures, in addition of being exploited for fast batched I/O and GPU accelerated rendering, are important for real-time virtual exploration through fast multi-scale geometric queries.

The elaboration and combination in a single system of these techniques is non trivial and represents a substantial enhancement to 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. While FOX takes inspiration from previous work, the technique is particularly motivated/customized by the peculiarity of the display environment. 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 graphics 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

Developing a rendering system targeting natural interactive inspection of massive surface models on light-field displays requires the combination of state-of-the-art results in a number of technological areas. In the following, we briefly discuss only the approaches most closely related to ours.

Motion control for virtual exploration. In the context of real-time massive models visualization applications, users require interactive control to explore the data. To this end, the interaction is usually restricted to a small number of navigational metaphors,

*CRS4 Visual Computing Group, POLARIS Edificio 1, 09010 Pula, Italy. www: `http://www.crs4.it/vic/` e-mail: {`marton|magus|gianni|gobbetti`}`@crs4.it`

Figure 1: ***Natural immersive exploration of the David 0.25mm model (1GTriangles) on a 35MPixel light field display.*** *Images taken with a hand-held camera. The model appears floating in the display workspace, providing correct parallax cues to multiple naked-eye observers. 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.*

for example, the identification of a number of interesting points or regions in the data, and the exploration of the remaining data in relation to these. Automatic or assisted navigation control has the potential to greatly enhance interaction with large data sets, especially in the context of Virtual Museums systems where novice users are supposed to be involved, and non-negligible training times must be avoided. Usability is often improved by assisting the computation of some degrees of freedom during navigation. In this category, surface orbiting methods constrain the camera to stay in a region around the object and with a specific orientation with respect to the surface [Khan et al. 2005; Burtnyk et al. 2006; Burtnyk et al. 2002]. Most of the work in this area is connected to camera motion control (see [Christie and Olivier 2009] for a survey). In this paper we propose an *object motion control* metaphor, combining the advantages of Speed-dependent Adaptive Zooming [Igarashi and Hinckley 2000] and Adaptive Surface Orbiting [Khan et al. 2005; McCrae et al. 2009], introducing constraints for taking into account the requirements for comfortable viewing on the light field display peculiarities, 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 fast access to fine details as well as a compelling model surfing experience. We require as input only two degrees of freedoms and a status button. The method can thus employed with many 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 scales at which an interactive renderer and user interaction system has 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 [Cignoni et al. 2004; Yoon et al. 2004; Cignoni et al. 2005]. While the coarse-grained approach improves performance for rendering, a fine grained structure is still necessary for point queries, which are used by our 3D navigation system for finding anchor positions during object motion. To provide this, we augment the coarse-grained structure, which organizes the rendering nodes, with a fine-grained space partitioning structure within each node, which spatially indexes each of the contained triangles. The coarse multiresolution structure is based on a diamond hierarchy [Weiss and De Floriani 2010], similar to the one used in Batched Multi-triangulation [Cignoni et al. 2005] and constructed with an Adaptive Tetrapuzzles approach [Cignoni et al. 2004]. 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. [Lauterbach et al. 2007] 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 multiprojector systems and parallax barriers or lenticular screens. The light-field display hardware employed in this work has been realized by Holografika (see `www.holografika.com`) and is commercially available. It is based on projection technology and uses a specially arranged projector array controlled by a PC cluster and a holographic screen. Large multi-view light field displays require generating multiple views for delivering perspective correct images. As in other state-of-the art rendering methods for such displays, we exploit multiple center of projection (MCOP) geometries [Jones et al. 2007] and adaptive sampling [Agus et al. 2008] to fit with the display geometry and the finite angular resolution of light beams. In addition, we employ a multipass rendering method, which allows us to realize depth-dependent filtering [Zwicker et al. 2007]. For cluster-parallel rendering, we use a sort-first parallel rendering approach with an adaptive out-of-core GPU renderer per back-end node, rather than using an object-based server-push philosophy as in previous light field display rendering systems [Bettio et al. 2008]. This makes it possible to reduce server burden and supports the use of different refinement strategies for the rendering and control system.

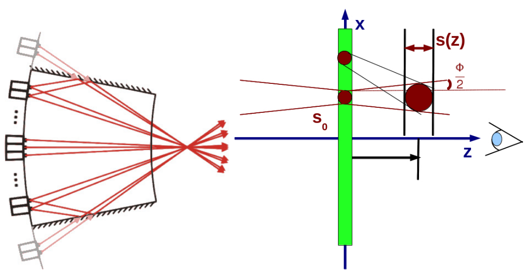

Figure 2: ***Light field display concept.*** *Left: the display is based on projection technology and uses a specially arranged projector array and a holographic screen. Right: the finite angular size of the light beams determines the voxel dimension as a function of distance from the screen.*

3 Context and system overview

Lightfield display overview The light-field display hardware employed in this work uses a specially arranged projector array controlled by a PC cluster and a holographic screen (see Fig. 2 left). The projectors are densely arranged at a fixed constant distance from a curved (cylindrical section) screen. All of them project their specific image onto the holographic screen to build up a light field. Mirrors positioned at the side 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 Φ. Vertically, the screen scatters widely, hence the projected image can be viewed from essentially any height. With this approach, a horizontal-parallax-only display is obtained. 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. Following [Jones et al. 2007; Agus et al. 2008], we employ a multiple-center-of-projection (MCOP) technique for generating images with good stereo and motion parallax cues. Our technique 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 only design (see Fig. 3). 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 [Agus et al. 2008].

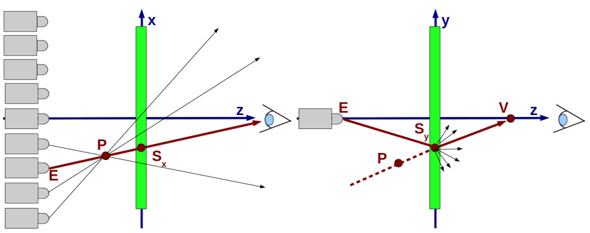

Figure 3: ***Light field display geometry.*** *Left: horizontally, the screen is sharply transmissive and maintains separation between views. Center: vertically, the screen scatters widely so the projected image can be viewed from essentially any height.*

Driving the display Our integrated system has to enable multiple naked-eye users to perceive detailed giga-triangles models as floating in space, responsive to their actions. Given the size of the model, adaptive out-of-core structures must be used both for rendering and for the geometric queries required by our interaction paradigm. These structures are integrated within a parallel system that drives the multi-projector display. The overall system concept is illustrated in Fig. 4. 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 front-end PC is connected to one or more input devices able to produce a 2D vector plus a state (3D sensor, or free-hand recognition) and manages the user-interface delivering to the back-end PCs the information for the current model position and rendering parameters. The front end PC uses an adaptive loader to maintain in core only the part of the surface required for the geometric queries. For nearest neighbor queries, we use a graph cut in which the level of detail is determined by a radial function centered at the search hot spot and decreases the required accuracy proportionally to the distance to the center This allows us to perform filtered spatial queries consistently with the current viewing scale. The system uses a sort-first parallel rendering approach, in which each back-end PC is responsible only for the images associated to the 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 multiple projectors are controlled by a single PC using multiple graphics boards. Each back-end process controls a frame-buffer portion where it renders the multiresolution model, adaptively loaded from out-of-core, and some visual feedback for the motion control. Differently from previous light field display rendering systems [Bettio et al. 2008], we do not push data to rendering nodes from the front end, 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 adapt the frequency content of the scene to the display spatial resolution using an image-based two-passes depth-of-field method implemented in a post-processing fragment shader [Zhou et al. 2007], with a circle of confusion corresponding to the depth-dependent spatial resolution of the light-field display.

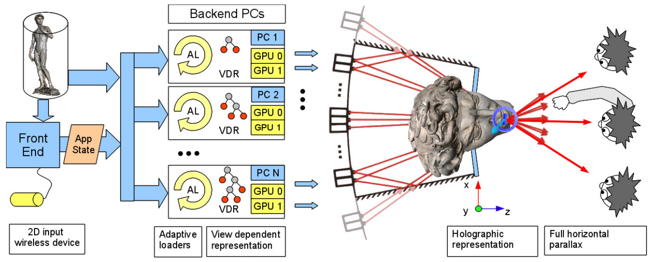

Figure 4: ***Virtual environment concept.*** *A user moves the model through some 2D device, whose input is elaborated from front-end PC, which computes new modeling transformation and sends them to the rendering back-ends. Back-end nodes update their view-dependent representations by asynchronously fetching data from the out-of-core database. Multiple users perceive the model as floating in space, from the new updated position.*

4 Natural massive models exploration

The 3D display, and the related rendering methods, have peculiar characteristics, which impose constraints to the interaction and rendering system in order to have a compelling visualization 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:

- 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 [Agus et al. 2008] (see Fig. 2 right);
- the calibration technique minimizes errors only on the screen surface; thus, the effective field of depth of the display is re-

duced not only because of the diminishing spatial resolution but also 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 performances are obtained when (a) the scene is kept centered with respect to the screen; (b) the scene remains inside a limited depth range; and (c) the frequency details of the objects are adapted to the display spatial accuracy.

4.1 Focus sliding surface (FOX) interactive navigation metaphor

In general, natural 3D object exploration can be a difficult task for a novel user, also with a common 2D display. Things get also more complicated when using a light field display, because object should remain within a certain depth range to produce a good quality image. General interaction metaphors, like rotate pan and zoom, besides being non trivial to master, can easily move the part of the surface on which one is interested in away from the hotspot of the display, with a further increase in complexity of the navigation task and visual discomfort.

We introduce a 3D user interaction technique which 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 technique is dubbed "focus sliding surface" (FOX). We attempt to use as many constraints as possible to simplify the number of controls needed to position models within the light field display workspace during typical object inspection. The method does not require learning specialized gestures, and is well adapted to a variety of input devices, including vision-based techniques for fully unencumbered exploration. The basic idea is that all navigation commands should move and scale the inspected object in such a way that the object surface remains in contact with the display hotspot, placed near the screen center, with the local (smoothed) surface plane parallel to the screen, and (optionally) the object's preferential up direction oriented upwards in the real world. In such a way, a user 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. Such an approach, which constrains the object to slide on an anchor point placed near the screen center, nicely handles simple convex surfaces, slightly concave surfaces, and, through the usage of multiresolution models for approximating the surface (see later), jumps across gaps or holes. These kinds of objects, with possibly significant protrusions and cavities but a relatively simple topology, correspond well to the typical cultural heritage models (e.g., statues) targeted by our application.

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 maintain the model oriented according to its preferential up direction. In order to apply the transformation, the surface normal $\mathbf{n}$ 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 re-

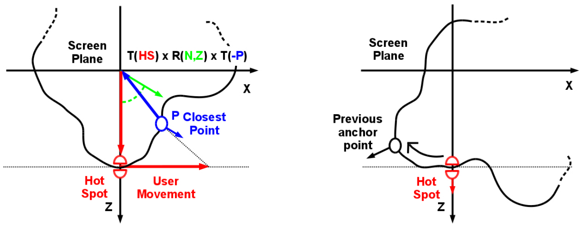

Figure 5: ***Constrained panning and rotation.*** *Left: the red horizontal arrow represents the cursor movement into the plane identified by the hotspot and the front direction.* $\mathbf{p}$ *is the closest point of the model surface to this new cursor position; the model will be translated to the 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.*

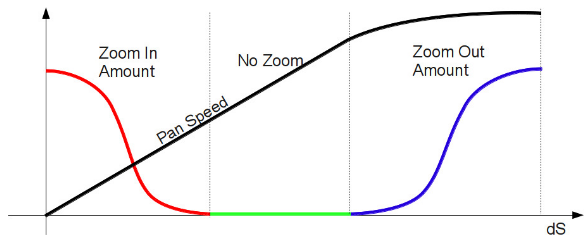

Figure 6: ***Zoom and pan.*** *Zoom and pan as functions of cursor movement: Zoom amount: before the left threshold the zoom-in amount is a decreasing smooth-step of the* dS*, in central part there is no zoom, while after the higher threshold the zoom-out amount is an increasing smooth-step of the* dS*.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

sulting 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 $\mathbf{s_c}$ the screen center. The new modeling matrix is then computed by $\overline{\mathbf{M}}_{\mathbf{i+1}} = \overline{\delta} \times \overline{\mathbf{M}}_{\mathbf{i}}$.

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 idea behind this approach is that when the user starts its input, but remains steady, he is focusing on something and he wants to see more details about it, thus we first keep the model unchanged, and then slowly start to scale up the object, increasing zoom speed with time. When the user, instead, is slowly moving, it means 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.

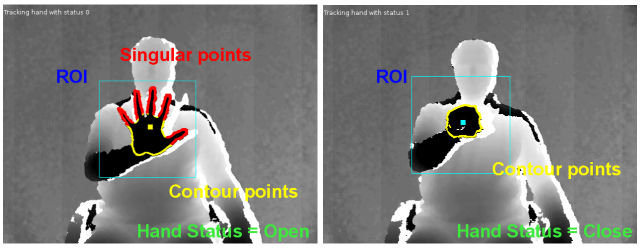

Figure 7: ***Hand tracking using depth sensors.*** *Open and closed hand are detected by estimating contour curvature of the hand, as to individuate singular points.*

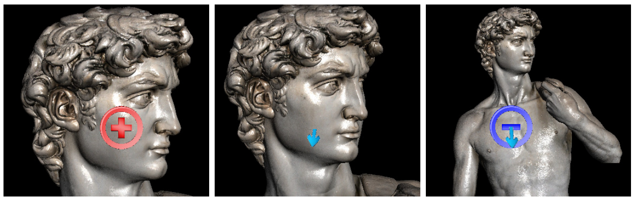

Figure 8: ***Cursor glyphs.*** *On the left image a red cross inside a circle indicates zoom-in. On the central image an arrow shows the direction of movement, while on the right image there is the symbol for zoom-out with also the panning arrow.*

4.2 Input mapping

We can use a variety of devices for motion control, since we simply need a 2D vector, and a state (press/released). A 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.

In order to support fully unencumbered manipulation, we have implemented a tracker using a Kinect depth sensor to recognize hand movements. In each moment, the hand point cloud in world coordinates is used to compute the cursor position, and the hand state. 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 points has been demonstrated in the past [Argyros and Lourakis 2006] to be a robust way to detect fingertips. In our case, we apply the method to depth images 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. the points whose curvature values exceed a given threshold are then used to identify the hand fingers. Even if the method is robust enough to track fingers motion, we are just interested in recognizing only a single status (hand close or open), and it is obtained by simply thresholding the number of singular points (see figure 7).

4.3 Cursor glyphs

Quality of experience is improved by providing visual indication of the current interaction modes through cursor glyphs (see Fig. 8).

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. An arrow representing the movement direction is possibly superimposed to zoom glyphs when pan is present. All glyph sizes are proportional to the norm of the represented quantities.

4.4 Dealing with massive models

Our system must ensure interactive rates with extremely massive models. This requires adaptive algorithms and structures for supporting rendering as well the multi-scale geometric queries required by FOX.

In order to be able to interactively render massive models, we consider a multiresolution structure based on a modification of the Adaptive Tetra Puzzles [Cignoni et al. 2004] method, which allows us to select nearest neighbor points at different levels of resolution in addition to supporting adaptive rendering. The underlying idea of the Adaptive Tetra Puzzles method is to depart from current point or triangle-based multiresolution models and adopt a patch-based data structure, 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 multiresolution 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 order to also 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 the method is efficient enough to ensure interactive and high quality rendering, but too coarse for the spatial queries. For this reason, we introduce a fine grained BSP structure, which is maintained within each node in order to spatially index individual triangles, helping and speeding up the spatial search. This BSP structure is constructed on-the-fly at patch loading time using a fast recursive split procedure. 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 defines a balanced spatial bounding box tree on the patch mesh, such that only the two bounding boxes must be stored. At run-time, the tree can be used for search queries implemented with top-down descents. The model closest point $\mathbf{p}$ and its normal $\mathbf{n}$ are computed by performing a k-nearest neighbor search over the model surface and extracting a certain number of points (64 in our current implementation), which are blended together with Gaussian weights, which fall out with the distance from the search point, with a standard deviation equal to the median of the found distances. The combining of the precomputed geometric simplification of the multiresolution model with the blending of multiple points allows us to smoothly handle non-trivial models.

Our multiresolution structure is integrated in a parallel system, which uses two adaptively refined versions of the model: one for the rendering, and one for interaction support. 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 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 from a specific projector parameters, all back-ends converge to the same representation without the need to exchange information, and the overall image is fully continuous.

5 Implementation and results

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. Each back-end PC has thus to generate $4\times800\times600$ 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 $David0.25mm$ 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 can be inspected at a variety of scales. For instance, viewing the model full-figure 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 performances using a hand-held video camera freely moving in the camera and display workspace. Representative video frames are shown in Fig. 9. Please refer to the accompanying video for further results.

Performance Our multiresolution system is capable to maintain interactive performance using an accuracy of 1 triangle/pixel for the rendering back-ends and 1triangle/cm on the hot-spot for knn search. 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.

Users evaluation In order to assess the FOX manipulation metaphor, we performed an user evaluation, involving quantitative 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. To this end, we benchmarked FOX and compared it with a standard 5-DOF object-in-hand manipulation technique [Ware 1990], both implemented with a precise inertial ultra-sonic 6-DOF tracking device (Intersense IS-900 3D Mouse). The experiments consisted in letting users try the two different manipulation metaphors (FOX and 5-DOF), in the context of a guided target reaching task, where participants were asked to manipulate the model until reaching a given specified position with a specified zoom level. Furthermore, we evaluated whether a practical free hand implementation of our metaphor is reasonable, and how the reduced tracking accuracy impacts on user experience. Thus, we compared the FOX performances of a free hand Kinect implementation to the performance obtained considering the Intersense 3D mouse. The evaluation procedure involved 33 participants. Our preliminary results indicate 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 reduced when tasks were performed by trained users, since the 5-DOF metaphor offers more control, and subjects tend to avoid uncomfortable and blurred model configurations. For both novice and trained users, FOX was generally preferred as control metaphor. We plan to report a detailed quantitative and qualitative analysis of our user evaluation in an expanded version of this work.

6 Conclusions and future work

We have presented an interactive system for natural immersive exploration of extremely detailed surface models, which appear floating in space to multiple freely moving naked-eye viewers in a room-sized workspace.

The navigation interface appears to be reasonably intuitive to use even for casual users, which quickly understand how to manipulate the object after a very short trial and error period. A more reliable markerless hand tracking for interaction is an important area for future research. Finally, it is important to note that, in all situations, the object is explored while being maintained in good viewing conditions, because of the combination of assisted navigation with light-field specific visual effects.

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 the light field display hardware. The resulting virtual environment, which combines ease of use with high representation fidelity, appears well suited for creative installations at exhibition centers. Our current work is concentrating on improving our proof-of-concept vision-based tracking system to handling multiple users while providing more reliable input. We also plan to provide a detailed qualitative and quantitative analysis of our user evaluation.

Acknowledgments. 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

AGOCS, T., BALOGH, T., FORGACS, T., BETTIO, F., GOBBETTI, E., AND ZANETTI, G. 2006. A large scale interactive holographic display. In *Proc. IEEE VR Workshop on Emerging Display Technologies*.

AGUS, M., GOBBETTI, E., GUITIÁN, J. A. I., MARTON, F., AND PINTORE, G. 2008. GPU accelerated direct volume rendering on an interactive light field display. *Computer Graphics Forum 27*, 2, 231–240.

ARGYROS, A. A., AND LOURAKIS, M. I. A. 2006. Vision-based interpretation of hand gestures for remote control of a computer mouse. In *In Computer Vision in Human-Computer Interaction*, Springer-Verlag, 40–51.

BETTIO, F., GOBBETTI, E., MARTON, F., AND PINTORE, G. 2008. Scalable rendering of massive triangle meshes on light field displays. *Computers & Graphics 32*, 1, 55–64.

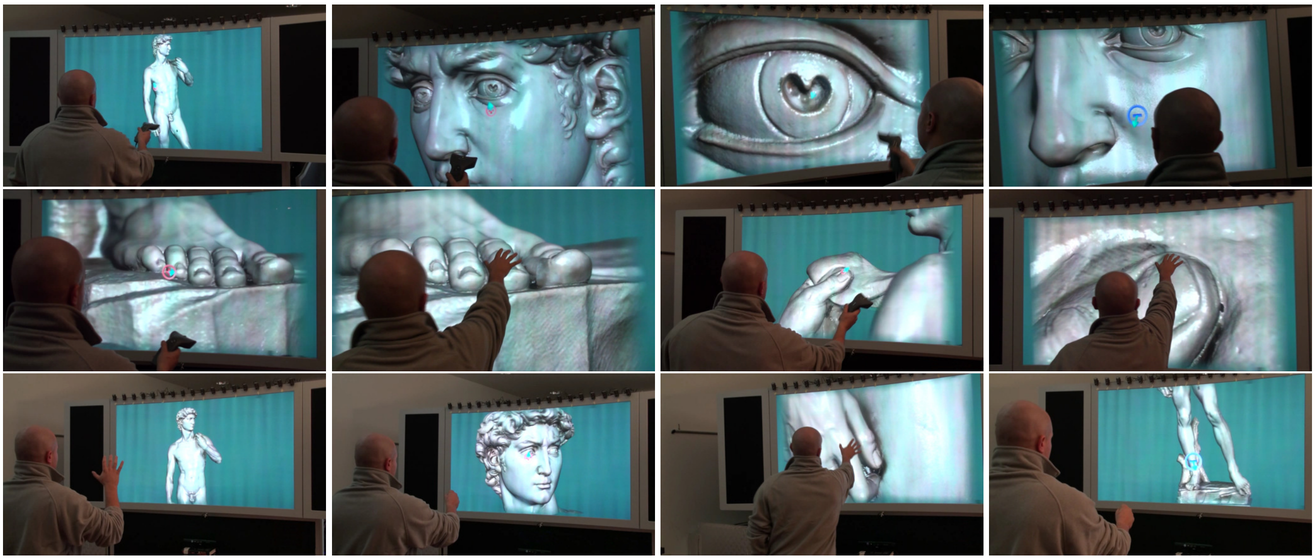

Figure 9: ***Live capture.*** *Representative frames recorded using a hand-held video camera freely moving in the camera and display workspace.*

BURTNYK, N., KHAN, A., FITZMAURICE, G., BALAKRISHNAN, R., AND KURTENBACH, G. 2002. Stylecam: interactive stylized 3d navigation using integrated spatial & temporal controls. In *Proc. UIST*, 101–110.

BURTNYK, N., KHAN, A., FITZMAURICE, G., AND KURTENBACH, G. 2006. ShowMotion: camera motion based 3D design review. In *Proc. I3D*, 167–174.

CHRISTIE, M., AND OLIVIER, P. 2009. Camera control in computer graphics: models, techniques and applications. In *ACM SIGGRAPH ASIA Courses*, 3:1–3:197.

CIGNONI, P., GANOVELLI, F., GOBBETTI, E., MARTON, F., PONCHIO, F., AND SCOPIGNO, R. 2004. Adaptive tetrapuzzles: efficient out-of-core construction and visualization of gigantic multiresolution polygonal models. *ACM Trans. Graph. 23*, 3, 796–803.

CIGNONI, P., GANOVELLI, F., GOBBETTI, E., MARTON, F., PONCHIO, F., AND SCOPIGNO, R. 2005. Batched multi triangulation. In *Proc. IEEE Visualization*, 207–214.

IGARASHI, T., AND HINCKLEY, K. 2000. Speed-dependent automatic zooming for browsing large documents. In *Proc. UIST*, 139–148.

JONES, A., MCDOWALL, I., YAMADA, H., BOLAS, M. T., AND DEBEVEC, P. E. 2007. Rendering for an interactive 360 degree light field display. *ACM Trans. Graph 26*, 3, 40.

KHAN, A., KOMALO, B., STAM, J., FITZMAURICE, G., AND KURTENBACH, G. 2005. Hovercam: interactive 3d navigation for proximal object inspection. In *Proc. I3D*, 73–80.

LAUTERBACH, C., YOON, S.-E., AND MANOCHA, D. 2007. Ray-strips: A compact mesh representation for interactive ray tracing. In *IEEE/EG Symposium on Interactive Ray Tracing*, 19–26.

MCCRAE, J., MORDATCH, I., GLUECK, M., AND KHAN, A. 2009. Multiscale 3D navigation. In *Proc. I3D*, 7–14.

WARE, C. 1990. Using hand position for virtual object placement. *Vis. Comput. 6* (November), 245–253.

WEISS, K., AND DE FLORIANI, L. 2010. Simplex and diamond hierarchies: Models and applications. In *Eurographics State of the Art Reports*, 113–136.

YOON, S.-E., SALOMON, B., GAYLE, R., AND MANOCHA, D. 2004. Quick-VDR: Interactive View-Dependent Rendering of Massive Models. In *Proceedings of IEEE Visualization 2004*, 131–138.

ZHOU, T., CHEN, J., AND PULLEN, M. 2007. Accurate depth of field simulation in real time. *Computer Graphics Forum 26*, 1, 15–23.

ZWICKER, M., VETRO, A., YEA, S., MATUSIK, W., PFISTER, H., AND DURAND, F. 2007. Resampling, antialiasing, and compression in multiview 3-D displays. *IEEE Signal Processing Magazine 24*, 6, 88–96.