Machine Vision and Applications (2009) 21:69–83
DOI 10.1007/s00138-008-0142-2

ORIGINAL PAPER

Integrating perceptual level of detail with head-pose estimation and its uncertainty

Javier E. Martinez · Ali Erol · George Bebis · Richard Boyle · Xander Twombly

Received: 5 September 2007 / Revised: 11 February 2008 / Accepted: 17 April 2008 / Published online: 22 July 2008

Abstract Immersive virtual environments with life-like interaction capabilities can provide a high fidelity view of the virtual world and seamless interaction methods to the user. High demanding requirements, however, raise many challenges in the development of sensing technologies and display systems. The focus of this study is on improving the performance of human–computer interaction by rendering optimizations guided by head pose estimates and their uncertainties. This work is part of a larger study currently being under investigation at NASA Ames, called "Virtual GloveboX" (VGX). VGX is a virtual simulator that aims to provide advanced training and simulation capabilities for astronauts to perform precise biological experiments in a glovebox aboard the International Space Station (ISS). Our objective is to enhance the virtual experience by incorporating information about the user's viewing direction into the rendering process. In our system, viewing direction is approximated by estimating head orientation using markers placed on a pair of polarized eye-glasses. Using eye-glasses does not pose any constraints in our operational environment since they are an integral part of a stereo display used in VGX. During rendering, perceptual level of detail methods are coupled with head-pose estimation to improve the visual experience. A key contribution of our work is incorporating head pose estimation uncertainties into the level of detail computations to account for head pose estimation errors. Subject tests designed to quantify user satisfaction under different modes of operation indicate that incorporating uncertainty information during rendering improves the visual experience of the user.

Keywords Virtual environments · Perceptual level of detail · Head-pose estimation · Uncertainty estimation

1 Introduction

Virtual environments (VEs) have for a long time been of great interest to researchers and the general public. Creating virtual worlds where the laws of nature can be replicated opens the door to many interesting applications such as training systems, surgical simulations, machinery tele-operation (e.g. in hazardous situations), diagnosis and therapy in neuroscience, and visualization of large datasets among others. The key element of these computing environments is an immersion effect provided by a realistic view of the virtual world and seamless interaction methods. The high complexity required in creating such immersion effects has motivated significant research and development in display, rendering and sensing technologies. Direct sense of the hand, eye-gaze, head and even the whole human body is required to capture natural input, while new display technologies accompanied by fast rendering algorithms are needed to convey high quality visual information in real-time.

Display subsystems in VEs can show marginally different characteristics than conventional ones. Many applications

J. E. Martinez · A. Erol · G. Bebis (✉)
Computer Vision Laboratory, University of Nevada,
Reno, NV 89557, USA
e-mail: bebis@cse.unr.edu

J. E. Martinez
e-mail: javier@cse.unr.edu

A. Erol
e-mail: aerol@cse.unr.edu

R. Boyle · X. Twombly
BioVis Laboratory, NASA Ames Research Center,
Moffett Field, CA 94035, USA
e-mail: rboyle@mail.arc.nasa.gov

X. Twombly
e-mail: xtwombly@mail.arc.nasa.gov

require simulating our natural way of viewing the world, which is not as simple as displaying an image sequence on a static display screen. In most cases, head position and eye-gaze determine what we see; therefore, tracking viewing direction and integrating this information with rendering is usually necessary to provide a natural view of the VE. This type of view control is especially important for head mounted displays (HMDs) and flat stereo displays.

Navigating through a VE can be naturally performed using some information about the user's viewing direction. This information can be used to compensate for the user's constantly changing position and orientation relative to a flat display. When the location and orientation of the screen relative to the user's viewing direction deviates significantly from the standard perspective view, a series of modifications to the viewport mapping can be made to generate a visual image that is consistently registered with the physical space. Tracking viewing direction can also be useful is rendering optimizations. Once the viewing direction is known, it becomes possible to employ adaptive level of detail (LOD) algorithms [1] to improve the visual experience perceived by the users without major increases in computational load. In a sense, systems employing LOD simulate the multi-resolution characteristics of the human visual system to avoid rendering everything at the highest possible detail.

Choosing an effective motion tracking method to implement advanced display systems represents an important design decision that can have considerable effects on user satisfaction. An important criterion to be considered is the level of intrusiveness introduced by the method. Computer vision has a distinctive role as a direct sensing method because of its non-intrusive, non-contact nature. However, computer vision faces several other challenges including generality, precision, robustness and processing speed requirements. Even in the case of head tracking, a 6 degree-of-freedom (DOF) rigid object tracking problem, various issues could have an adverse effect on the robustness and precision of pose estimates (e.g., varying illumination conditions, varying background, feature extraction, modeling errors). Overcoming these difficulties requires more research.

Currently, precise high frequency motion tracking can be accomplished using electro-mechanical or magnetic sensing devices [2]. These methods, however, are invasive. The sensors require contact with the body, hindering the naturalness of interaction. As the number of sensors increases, the workspace gets more and more tethered, while calibration of measured DOFs gets more and more time consuming. This study is part of a larger study and aims to improve the fidelity of display subsystems in an immersive virtual environment, called VGX (see Fig. 1). The goal of VGX, which is currently under investigation at NASA Ames, is to assist in training astronauts to conduct technically challenging life-science experiments in a glovebox aboard the ISS. VGX integrates high-fidelity graphics, force-feedback devices, and real-time computer simulation engines to achieve an immersive training environment [3,4].

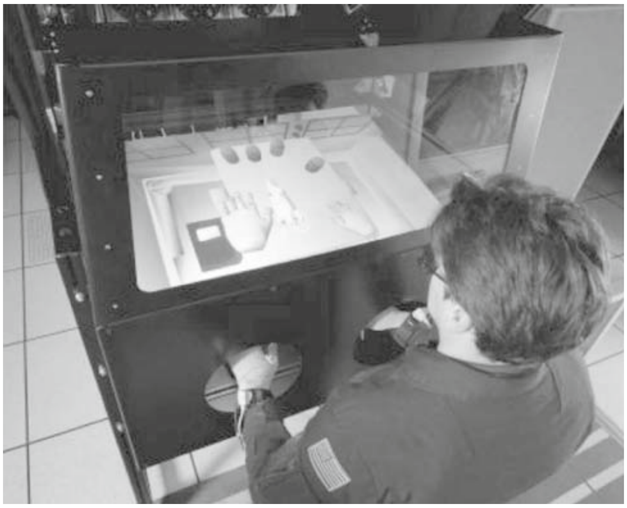

Fig. 1 VGX: A stereoscopic display station provides a high-resolution immersive environment corresponding to a glovebox facility

To support interaction with virtual objects, the current interface of VGX uses off-the-shelf tracking and haptic feedback devices, which contain cumbersome elements such as wired gloves, tethered magnetic trackers, and haptic armatures inside the workspace. The visualization system consists of a custom stereo rendering system using LCD projectors. The only invasive element of the display system is a pair of polarized eye-glasses that have to be worn by the user to get a 3D view of the inside of the glovebox. Our objective is to enhance the display subsystem of VGX by incorporating information about the user's head pose into the rendering process. For head tracking, a computer-vision-based solution has been adopted due to its non-invasiveness.

In the current implementation of our system, viewing direction is approximated by head pose using several markers placed on the frame of a pair of polarized eye-glasses. The use of markers does not introduce any restrictions in our operational environment since the eye-glasses represent an integral part of the stereo display system used in VGX. Using markers greatly improves the robustness, precision and speed of head-pose estimation. During rendering, perceptual level of detail (PLOD) algorithms (i.e., LOD algorithms using criteria derived from physiological aspects of human vision) are employed. A key contribution of our work is the incorporation of head-pose estimation uncertainties in PLOD computations to account for errors in head-pose estimation. As our experimental results illustrate, ignoring errors in head pose estimation decreases user satisfaction. An earlier version of this work has appeared in [5].

The rest of the paper is organized as follows: Sect. 2 provides background information and a brief review of PLOD methods. Section 3 describes our prototype system, operational environment and hardware components. The head pose estimation algorithm, and the implementation details of the PLOD system are presented in Sects. 4 and 5, respectively. Our experimental results, including user satisfaction experiments, are reported in Sect. 6. Finally, Sect. 7 concludes our study and provides directions for future research.

2 Background and previous work

The LOD is a rendering approach where a scene is rendered by adaptively changing the amount of detail across it. Typically, areas more visible to the user are rendered with higher detail. LOD facilitates various tasks, such as image compression and bandwidth reduction [6], simulation of visual defects [7], or rendering optimization [1,8–10]. Depending on the data used as input, LOD methods can be categorized as image or geometry-based. Changes in the detail of objects in a 3D scene can be used to further categorize geometry-based approaches into discrete and continuous. In discrete LOD methods, a small, finite number of LODs are created off line for a given object. Then, depending on the distance of the object from the viewer, a single LOD for the whole object is rendered. Details of the object are controlled continuously all over the scene depending on various criteria based on perceptual aspects of the human vision system. The work presented here falls in the category of continuous LOD.

At a higher level, a PLOD system can be described as the interaction of three elements as shown in Fig. 2: (1) criteria, (2) mechanism, and (3) error measure. Criteria encompass those variables that affect the selection of an LOD. The mechanism selects a way of manipulating the geometry so that it has the desired LOD. Finally, the error measure is used to control and measure the model's deviation from the original model.

Two of the variables used as part of the criteria are the relative size and distance of the object from the camera or point of view of the user. Research work in experimental psychology has introduced additional variables including:

- *Contrast sensitivity* the LOD is modulated depending on the contrast that determines the maximum and minimum spatial frequencies visible to humans. This relation is captured by a *Contrast Sensitivity Function* (CSF) [11].
- *Velocity* the LOD is modulated proportionally to the relative velocity of the eyes across the visual field. The visual system has reduced sensitivity to the details of moving objects.
- *Eccentricity* the LOD is modulated proportionally to the angular distance of the object to the viewpoint. The visual sensitivity falls off at the visual periphery.
- *Depth of field* the LOD is modulated proportionally to the distance to the Panum's fusional area [9]. This is used only in connection with stereo-vision.

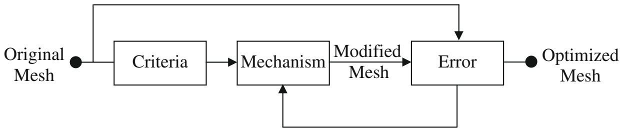

Fig. 2 Interaction between main LOD elements

There are various methods for changing object geometry in order to achieve the desired LOD determined by evaluating the criteria. In continuous LOD, polygonal simplification algorithms are utilized. These algorithms can be grouped into four main categories [12]:

- *Sampling* a dense mesh surface is sampled to obtain a low resolution version.
- *Adaptive subdivision* a low resolution mesh is created from the original one, which is then recursively subdivided to create higher resolution meshes.
- *Decimation* vertices and faces are iteratively removed from the mesh with resulting holes being triangulated.
- *Vertex merging* vertices are collapsed into a single vertex.

Measuring the error between a high resolution mesh and a lower resolution one involves various factors. The most obvious one is the geometric error which depends on the number and location of the vertices, however, small geometric errors may not always translate to small visible errors on the screen. What is really needed is a perceptually-consistent error measure, an issue which is not well understood yet. Some less obvious sources of error include color, normal, and even texture [12]. Common error measures in the literature include vertex–vertex, vertex–plane, vertex–surface, and surface–surface distances.

The perceptual criteria described above depend on estimating eye-gaze or making certain assumptions. There exist several systems making use of eye-gaze to guide perceptually motivated simplifications including [1,13–15]. Both [1] and [15] use an eye tracker to estimate eye-gaze. Computer vision provides a non-contact method for estimating eye-gaze; however, a non-intrusive solution that can compensate for large head motions is not available yet. In [1], the user's head was placed in a chin rest to avoid calculating the position of the eyes. Only [15] tracks the head and the eyes simultaneously using IR cameras and sensors attached to a HMD.

3 System design

The design of a VE system is closely related to the intended application, both for performance and economical reasons.

Our aim in this study is to build a system capable of displaying images to a user while the user interacts with the system. The display being used within VGX is a stereo display where images for each eye are alternatively projected on a screen. The user wears a pair of polarized glasses to make each eye view only the images intended for it. This is minimally invasive compared to the use of HMDs .

Ideally, we would like to estimate the point of interest (POI) by estimating eye-gaze exactly, in order to guide the optimizations. Full eye-gaze estimation requires solving two separate problems: (1) estimating the orientation of the head and (2) estimating the orientation of the eyes within their sockets. However, the use of polarized glasses in the VGX environment (see Fig. 3) blocks direct view of the eyes. Therefore, our current implementation uses head orientation for approximating eye-gaze (e.g., [16]).

Several techniques have been proposed for head tracking and pose estimation in human-computer interaction including [17–19]. Since the emphasis of this work is on the integration of head pose uncertainties in the LOD computations, we opted for a simpler and more practical approach in order to estimate the 3D position and orientation of the head; however, more sophisticated approaches might be more appropriate in this context. Specifically, head orientation and location was estimated by tracking several markers placed on the frame of the eye-glasses in a specific configuration. The use of markers does not introduce any restrictions in our operational environment since the eye-glasses represent an integral part of the stereo display system used in VGX. Using markers greatly improves the robustness, precision and speed of head pose estimation. Since good illumination is always important in marker-based tracking, our solution must ensure that lighting does not distract the user. To deal with these issues, we are using IR illumination in conjunction with IR reflective markers.

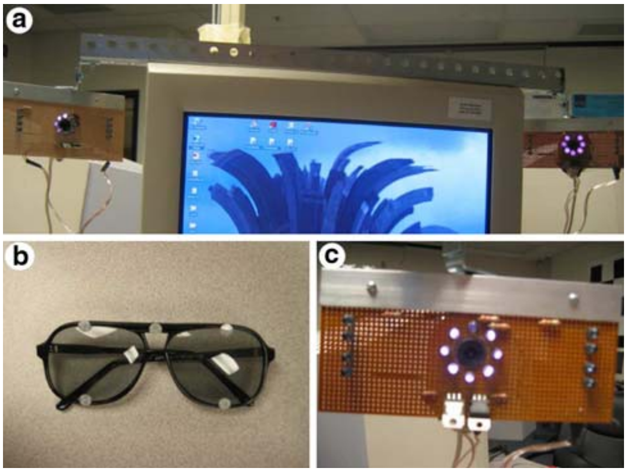

Fig. 3 System setup: **a** camera setup; **b** polarized eye-glasses with markers; **c** camera close-up with LEDs on

IR illumination is invisible to the human eye while common off-the-shelf CCD cameras can sense IR by simply removing the IR filter on the lens. Our system uses two Phillips TUCam webcams with their IR filters removed. An array of IR diodes, placed around the lens of each camera, generate IR light evenly at 800nm (see Fig. 3c). To filter out the visible light, a high-pass filter was placed behind the lens of each camera. In particular, the filter used was a Kodac Wratten 89c filter with a cutoff limit of 800nm. At this wavelength, only the red sensors in the camera can pick up any signal. The result is a grayscale image in the red channel. A sample image captured by the system is shown in Fig. 6(a). This configuration provides high quality images containing only the markers on a dark background, facilitating the segmentation of the markers.

In our prototype system, the cameras are fixed to the sides of a monitor as shown in Fig. 3(a). Both cameras continuously capture images at 30fps. The use of a stereo setting allows for recovering the 3D positions of the markers using triangulation. The cameras have been calibrated using a planar calibration pattern [20]. Additionally, they have been synchronized so that each pair of images is captured at the same instant of time. We have implemented image acquisition and camera synchronization using a custom application in DirectShow.

Our system contains three main modules as shown in Fig. 4: Head Pose Estimation (HPE), PLOD, and rendering. The HPE module is in charge of grabbing the images from the cameras, processing them, and estimating the position and orientation of the head, as well as the uncertainty associated with the estimates. This information is then passed on to the PLOD module which determines the minimal level of detail required to render the primitives in the scene. Using LOD information, the rendering module draws the scene on the screen in a way consistent with the user's point of view. The renderer used in our system is fairly standard and employs back face culling, frustum culling, and clipping to reduce computational load.

4 HPE module

This module takes as input a pair of images and processes them to estimate head-pose. This is performed by estimating head position and orientation vectors from the recovered 3D locations of the markers $\mathbf{P}_0$ to $\mathbf{P}_4$ as shown in Figs. 5 and 3b. A critical issue in head pose estimation is the accurate

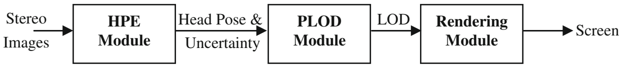

Fig. 4 Main components of our system

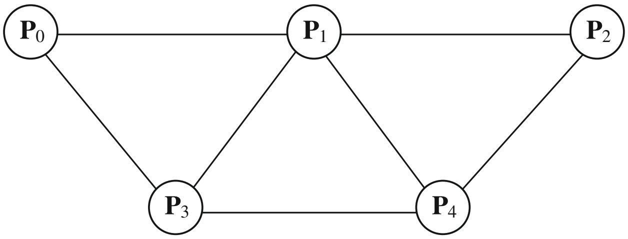

Fig. 5 Marker arrangement on the frame of the polarized eye-glasses

localization of the markers. Estimating the locations of the marker centers precisely is quite difficult since each marker occupies a few pixels only due to the relatively low resolution of the images captured by our system. Slight segmentation errors could also affect correct marker localization. Since errors in marker localization could affect head pose estimation seriously, our system models the uncertainties in head pose estimation and incorporates this information in the LOD computations to minimize error effects.

The HPE module includes three main stages: (1) marker extraction and identification, (2) head pose estimation, and (3) uncertainty estimation. The first stage segments the pixels associated with each marker and identifies each marker uniquely; the second stage uses stereo reconstruction to estimate the 3D locations of the markers and estimates head pose; the last stage computes the uncertainty in head pose.

4.1 Marker extraction and identification

Using IR illumination along with IR reflective markers allows for fast and robust marker detection and extraction. A sample image is shown in Fig. 6(a); as it can be observed, background information has already been suppressed due using a filter that blocks visible light. This allows detecting and extracting the markers using simple thresholding as shown in Fig. 6(b). The threshold value was experimentally set to 100. The thresholded image is then smoothed using a 9 × 9 Gaussian filter to eliminate noise as shown in Fig. 6(c). Sharp blobs with small area are then discarded since they correspond to reflections on the surface of the glasses. The special arrangement of the markers, as shown in Fig. 5, allows for identifying them uniquely in each image. A detailed description of our marker extraction and identification algorithm is provided below:

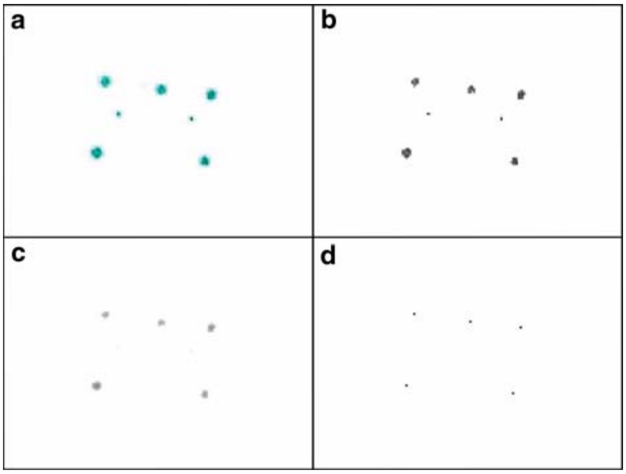

Fig. 6 Image processing pipeline: **a** input image, **b** marker segmentation (threshold value used: 100), **c** smoothed image (9x9 Gaussian filter), and **d** marker centers

(1) Threshold input image to segment markers.
(2) Smooth image and remove small, sharp blobs.
(3) Calculate the center or each marker using a weighted sum of pixel locations and their intensity.
(4) Find the three markers that lie on a line.
(5) The marker in the middle of the trio will be $\mathbf{P}_1$. The other two markers would be $\mathbf{P}_0$ and $\mathbf{P}_2$, however, their order must be determined. Let us denote them by $\mathbf{P}_a$ and $\mathbf{P}_b$.
(6) Calculate the distance of the remaining two markers from $\mathbf{P}_a$. Let us denote the closest and farthest markers as $\mathbf{P}'_a$ and $\mathbf{P}'_b$, respectively.
(7) If the sign of the cross product $(\mathbf{P}_a - \mathbf{P}_1) \times (\mathbf{P}'_a - \mathbf{P}_1)$ is positive, then $\mathbf{P}_a$ is $\mathbf{P}_0$, $\mathbf{P}'_a$ is $\mathbf{P}_3$, $\mathbf{P}_b$ is $\mathbf{P}_2$ and $\mathbf{P}'_b$ is $\mathbf{P}_4$; otherwise, $\mathbf{P}_b$ is $\mathbf{P}_0$, and $\mathbf{P}'_b$ is $\mathbf{P}_3$, $\mathbf{P}_a$ is $\mathbf{P}_2$, $\mathbf{P}'_a$ is $\mathbf{P}_4$.

4.2 Pose estimation

The use of off-the-shelf cameras with IR illumination has some drawbacks. The poor resolution of the webcams results in small marker images, on average totaling about 35 pixels. Apart from the small marker image size, the combination of sensor noise, lighting conditions and image processing all contribute to missing some pixels; particularly around the marker boundary. Traditional methods for pose estimation use the known geometry of the markers to recover their pose, for example, by fitting an ellipse to the marker image [21]. These methods are sensitive to missing pixels, especially around the marker's boundary, giving poor results. Here, we opted for a simpler but more practical approach using the more stable marker image centroid, however, one could consider using more sophisticated approaches.

Once the markers have been extracted and identified in both images, the centers of corresponding markers (i.e. the projections of the same marker in the each image) can be triangulated to compute their 3D coordinates. To estimate the pose of the head, we need to compute its position and orientation in 3D. Head position is taken as the 3D position of marker $\mathbf{P}_1$, that is, the marker in the middle of the eye-glasses (see Figs. 3(b), 5).

To compute head orientation, we associate a 3D coordinate system with it using three vectors as shown in Fig. 7.

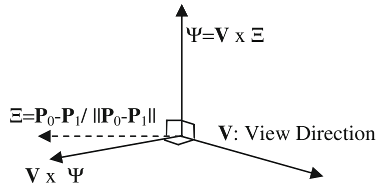

Fig. 7 The coordinate system representing head orientation

The first vector $\mathbf{V}$, which determines the POI on the display, is the average of the normals associated with the three main triangles formed by the markers as shown in Fig. 5 (i.e., (i) $\boldsymbol{P_0}$, $\boldsymbol{P_1}$, $\boldsymbol{P_3}$, (ii) $\boldsymbol{P_1}$, $\boldsymbol{P_2}$, $\boldsymbol{P_4}$, and (iii) $\boldsymbol{P_1}$, $\boldsymbol{P_3}$, $\boldsymbol{P_4}$). The second vector Ξ, is taken to be the unit vector along $\mathbf{P}_0 - \mathbf{P}_1$. The third vector Ψ, is computed as the cross product of the other two. Using this information, it is possible to define an orthogonal frame using $\mathbf{V} \times \Psi$. Currently, only the viewing vector $\mathbf{V}$ is used in the LOD computations. In the future, we plan to combine head orientation with eye orientation to compute eye gaze exactly as mentioned in Sect. 3.

The head-pose estimate obtained using the procedure above is considered to be a *mean estimate* for uncertainty modeling purposes. The main factors affecting the accuracy of the mean estimate include: (1) distance to the camera (i.e., affects the resolution of markers), (2) viewing direction with respect to the camera (i.e., affects the resolution of the markers), and (3) segmentation and camera calibration errors. The purpose of uncertainty estimation is to model the error in head pose estimation and incorporate it in the LOD computations.

4.3 Uncertainty estimation

In general, uncertainty estimation is a useful component of any computer vision system. A common method in modeling uncertainty is by error propagation [22], that is, propagating pixel-wise errors, originating from camera calibration and feature extraction errors, to 3D estimates. For simple tasks, such as local depth or orientation estimation [23], it is possible to derive analytic expressions for uncertainty; however, in our case we are dealing with global estimates that yield complex nonlinear equations. We have devised a simpler approach based on sampling that simulates error propagation in the three main processing stages. First, each marker is represented by a "cloud" of points corresponding to the pixels comprising each marker. Then, each cloud is uniformly sub-sampled to obtain a set of head-pose estimates. The variation of the resulting estimates from the mean estimate (i.e., see Sect. 4.2) is modeled as a multi-variate Gaussian distribution which is used in the LOD computations.

4.3.1 Marker representation

Ideally, the 3D location of each marker can be estimated by triangulating the 2D centers of each marker in the stereo image. Since accurate 2D marker center localization is prone to errors, we have devised a strategy to account for errors in head pose estimation. Specifically, given a pair of corresponding regions in the two images (i.e., projection of the same marker) we represent each marker as a "cloud" of points corresponding to the pixels contained in each region as shown in Fig. 8. Then, we form pairs of points, one from each image, to estimate the 3D coordinates of the marker. Since we do not know the correct correspondences, one could form all possible correspondences, which would yield to a "cloud" of 3D points for each marker. This, however, has high computational requirements. Fortunately, many of the hypothesized corresponding pairs can be ruled out quickly using the epipolar constraint [24] and structural constraints.

To take advantage of the epipolar constraint, we threshold the distance between the viewing rays starting from each camera optical center and passing through the corresponding pixels. This represents the reconstruction error (see Fig. 9). In ideal conditions, the rays should intersect and the distance should be zero; however, this is not the case in practice due to calibration and pixelization errors (see Fig. 9). The threshold was determined empirically by analyzing the reconstruction error histogram as shown in Fig. 10. In particular, the smallest error occurs between corresponding pixels which is

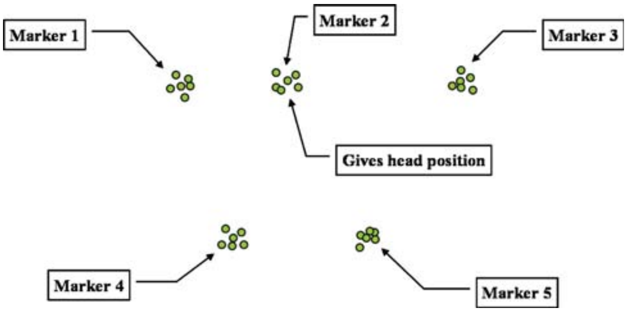

Fig. 8 Representing markers as clouds of points

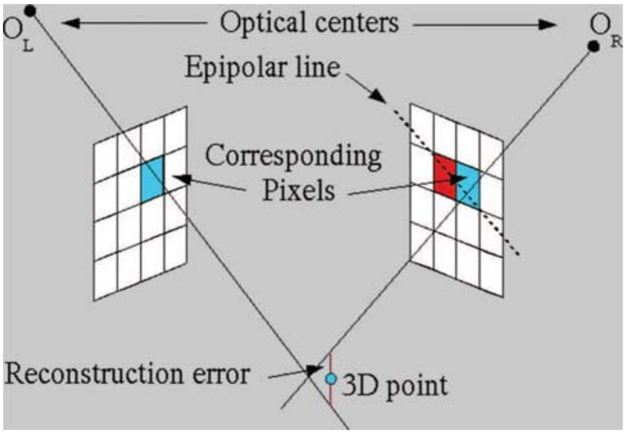

Fig. 9 Illustration of the reconstruction error

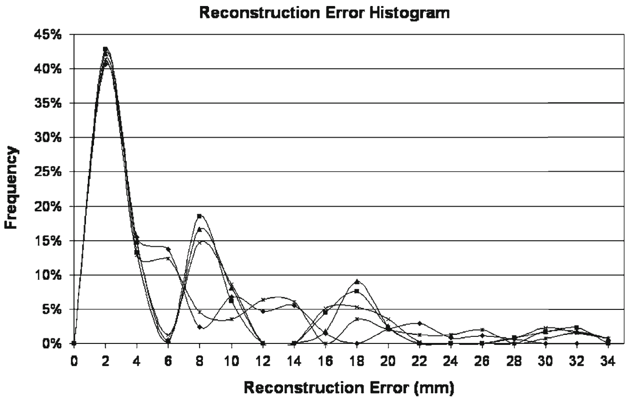

Fig. 10 Several reconstruction error histograms: the first peak of the histogram was used to determine the reconstruction error threshold

represented by the first peak in Fig. 10. For this reason, the reconstruction error threshold was set to 2.0 mm.

Although the epipolar constraint is very effective in ruling out many invalid correspondences, it can not eliminate all of them. To further reduce the number of invalid correspondences we apply structural constraints by eliminating topologically inconsistent pairings. Specifically, since all the markers have the same shape, we do not allow a point close to the boundary of a marker in one image to be paired with a point close to the center of a marker in the other image. To implement this criterion, we compare the distances of the points to marker centers. If the difference of the distances is greater than a threshold, then the pair is discarded. To set the value of this threshold, we used the average difference calculated over several frames.

4.3.2 Error modeling

The purpose of this step is to model the errors in head pose estimation which affect the location of the POI on the screen. Specifically, head pose uncertainty can be expressed in terms of orientation and position errors. Since position depends only one marker (i.e., $\mathbf{P}_1$), it is straightforward to model position uncertainty given a cloud of points. In fact, it is possible to estimate it analytically using error propagation techniques such as the ones reported in [23]. Our initial experiments, however, indicated that errors in orientation estimates are much more important in the context of our application that errors in position estimates. For example, when the user moves farther away from the display, slight orientation changes could cause large shifts in the POI. Therefore, we have ignored the uncertainty in head position and have estimated only the uncertainty in orientation.

Orientation estimates depend on the location of all five markers, each of which are represented by a cloud of points as described in Sect. 4.3.1. Any combination of 5 pixels, one from each cloud, produces an orientation estimate. A straightforward way to model orientation uncertainty is by considering all possible combinations; however, this would increase computational load. To keep computational requirements low, we uniformly sub-sample each cloud of points and form all possible combinations among the samples, yielding a "cloud" of orientation estimates. The reason for using uniform sampling is because, mathematically, each point in the cloud can yield a valid pixel in the marker images. The resulting orientation estimates are then coded as unit vectors in the head orientation frame using spherical coordinates as shown in Fig. 11. In spherical coordinates, two angles, (ϕ, θ), are sufficient to represent each orientation sample since all the vectors are taken to be unit vectors. Orientation uncertainty is modeled by fitting a Gaussian distribution centered at the viewing direction. Assuming that ϕ and θ are independent of each other, the sample covariance of the points on the (ϕ, θ) plane represents a maximum likelihood estimate of the parameters of the Gaussian distribution [25].

4.4 HPE validation using a magnetic tracker

To test the accuracy of head pose estimation, the output of our system was compared to the readings of a magnetic tracker. Specifically, we used a Flock of Birds (FoB) magnetic tracker by Ascension Technology Corporation to obtain ground truth data. FoB is capable of handling up to 144 measurements per second with a static accuracy of 1.8mm in position and ±0.5° in orientation. To use the magnetic tracker for validation, we first calibrated it with our vision-based system in order to find the transformation between the cameras' coordinate system and the one corresponding to the magnetic tracker. This involved placing a marker on the FoB and taking about 200 samples using the stereo rig and the FoB. We used the registration method reported in [26] to find the transformation matrix. It is worth noting that, some errors still exist in registering the two coordinate systems since physical

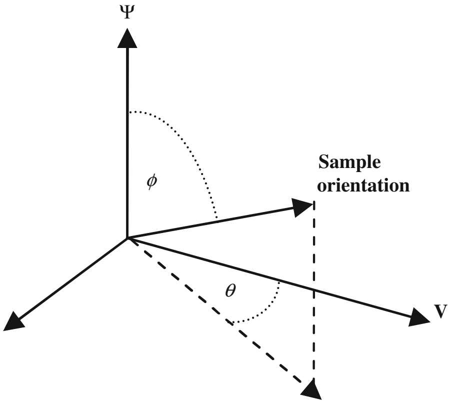

Fig. 11 Orientation uncertainty with respect to the mean expressed in terms of the angles θ and ϕ

constraints prevent us from placing the marker at the exact measurement point of the FoB magnetic tracker. This introduces a slight offset (i.e., about 4.89mm) between the data taken from the camera and those reported by the FoB. After the offset had been removed, the root mean-square error was measured to be 3.65, 2.65 and 3.39 mm in X, Y and Z axes, respectively, which is very close to the tracker's accuracy of 1.8 mm. Similar results were obtained in the other two axes (see Fig. 12). The root mean-square orientation error in (x, y, z) was $\theta_{\mathrm{err}} = (8.85°, 16.31°, 6.88°)$. Orientation errors were due to the large area covered when collecting the data sets. This caused the face to move far from the cameras from time to time which affected the marker images.

5 PLOD module

As shown in Fig. 2, a PLOD system contains three main elements: criteria, mechanism, and error measure. In our prototype system, we employed an adaptive subdivision algorithm as the mechanism to generate an optimized mesh. Specifically, a *base model*, corresponding to the lowest possible resolution, is constructed off-line. During rendering, the polygonal primitives in the model are recursively subdivided as needed. We used the algorithm reported in Junkins et. al. [10]; the main difference between that implementation and our implementation is that we incorporate perceptual information to the culling functions. Subdivision has some limitations in terms of its ability to capture the details of the original shape, especially at sharp corners and creases, but it is also fast, easy to implement, and memory efficient. Also, it eliminates the need for an error loop and allows for separating system performance from mechanism performance.

The basic processing step in the PLOD module is determining the desired LOD level for selected triangles in the base model. To gain performance, the rendering and PLOD modules were not separated as shown in Fig. 4; instead, they were highly coupled. When drawing the scene, the rendering

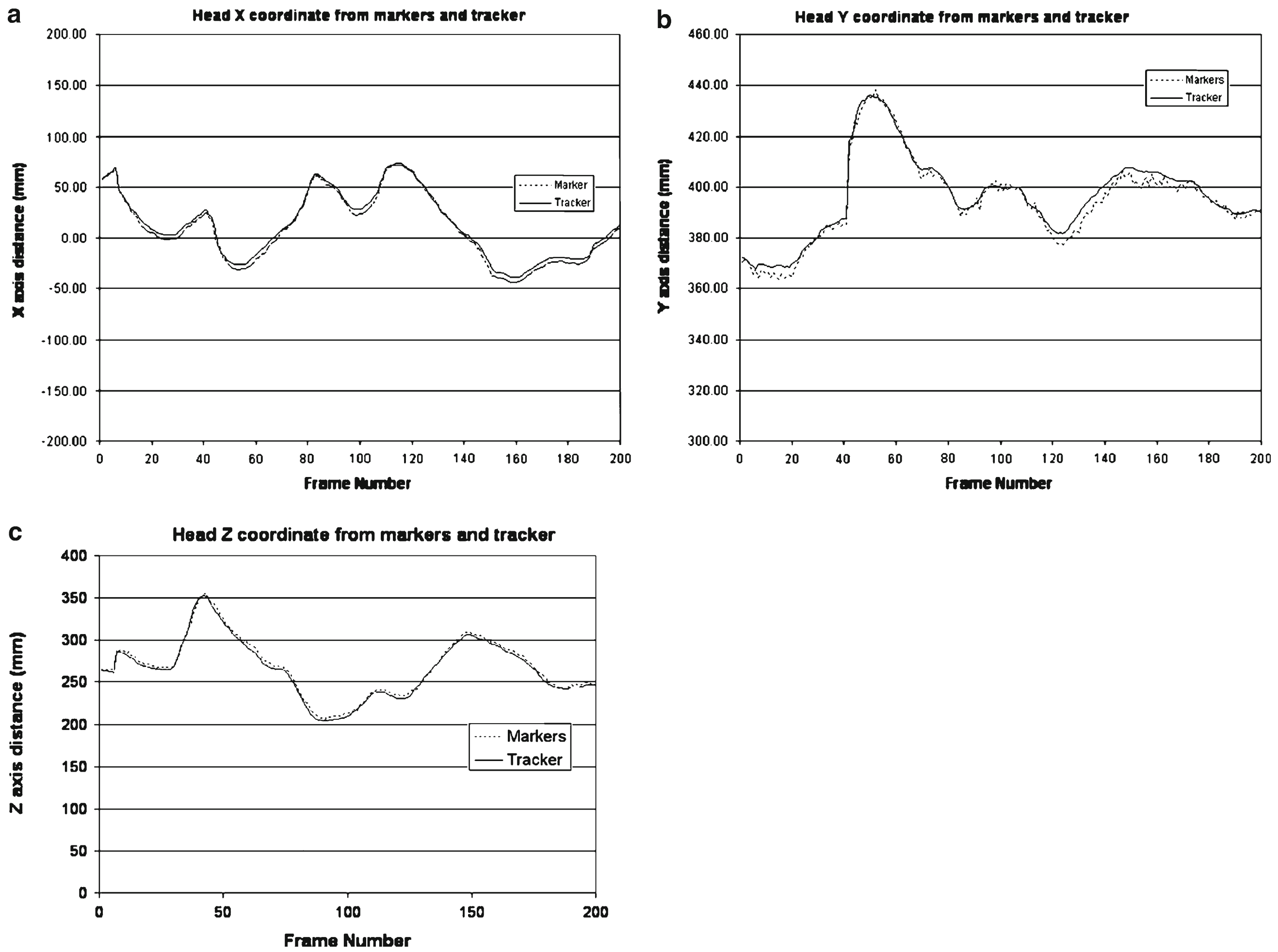

Fig. 12 Comparison of head's location obtained using vision-based and magnetic-based head tracking **a** x-coordinate; **b** y-coordinate; **c** z-coordinate

module discards as many triangles as possible by determining their visibility. Only those triangles that need to be drawn are passed to the PLOD module in order to determine the level of detail which corresponds to the number of subdivisions needed. The computation of the desired LOD depends on several factors including contrast, eccentricity, velocity, and spatial frequency.

Depth of field, mentioned in Sect. 2, was not accounted for because we are not using a stereoscopic display to render the images and we are not estimating eye-gaze vectors in our current implementation. Since all the features depend on the current view of the scene, they were computed on the fly at each frame. Once the features were calculated, a contrast sensitivity model [11] was employed to determine the desired LOD. An important processing step, specific to this study, is the incorporation of uncertainty head orientation estimates in the LOD computations. Although it is possible to make use of uncertainty information with respect to any feature, we determined that the most effective and critical one was eccentricity.

In the next section, we present the uncertainty-based eccentricity computations. Then, we summarize the rest of the processing steps including computation of other features and LOD determination.

5.1 Eccentricity and uncertainty

A triangle's eccentricity is defined to be the angle in degrees of the arc between the user's head position, the triangle's geometric center and the user's POI in the screen (see Fig. 13). To calculate it, the triangle's centroid is projected to the near plane and transformed to screen plane coordinates. The screen plane should not be confused with screen coordinates, which are measured in pixels; it is a 3D plane that represents the physical display and contains the POI. The POI corresponds to the intersection of the screen plane with the ray emanating from the head position, along the viewing direction. In head coordinates, as shown in Fig. 7, the eccentricity corresponds to the angle between the **V** axis (i.e. the viewing direction) and the vector between the head position and the triangle's projection. LOD is inversely proportional to eccentricity, that is, eccentricity increases as the LOD decreases.

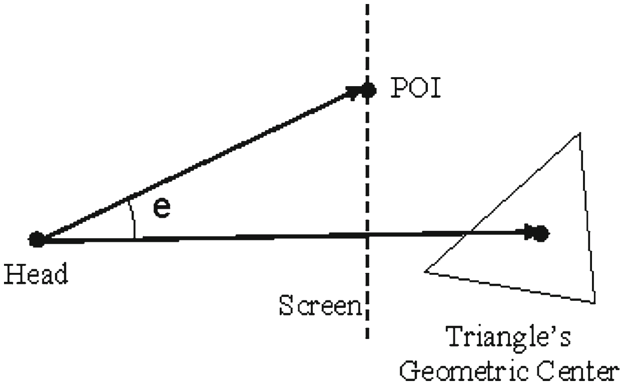

Fig. 13 Visual interpretation of a triangle's eccentricity

In the presence of uncertainty, the location of the POI is described in terms of a 2D Gaussian defined on the (ϕ, θ) plane as discussed in Sect. 4.3.2. To make use of uncertainty information, a triangle's projection is shifted towards the origin (i.e., along the viewing direction), proportionally to the probability that the projection itself is the POI. This "pseudo-shift" causes a triangle to have higher level of detail by *moving* it closer to the POI proportionally to uncertainty. Formally, let Σ denote the covariance matrix of the Gaussian, and $\mathbf{P}$ denote the (ϕ, θ) coordinates of the triangle's projection on the screen plane. Then, the uncertainty-corrected coordinate $\mathbf{P}'$ is obtained as follows:

$$\mathbf{P}' = \left(1 - e^{\mathbf{P}^T \Sigma^{-1} \mathbf{P}}\right) \mathbf{P} \quad (1)$$

5.2 Contrast, velocity and spatial frequency

Velocity is defined as the relative velocity of the triangle with respect to the head. Velocity information can be computed by keeping a short history of head pose estimates. To compute spatial frequency and contrast, the projection of the triangle has to be computed. Spatial frequency is given by the inverse of the diameter of the triangles' projection. Contrast is defined as follows:

$$C = \frac{L_{\max} - L_{\min}}{L_{\max} + L_{\min}} \quad (2)$$

where $L_{\min}$ and $L_{\max}$ denote the relative minimum and maximum luminance levels inside the projection of the triangle [11]. To calculate the luminance levels, the triangle is rendered and the pixels are read back from the frame buffer. The advantage of this approach is that it takes lighting into account. If a triangle is under a shadow, the LOD will be lower than the case of a triangle being under a bright light. The disadvantage is that reading data from the frame buffer is an expensive operation, especially when a large number of triangles must be processed. To mitigate this problem, only a 10 × 10 pixel region around the triangle's centroid was used for contrast calculations.

5.3 A Triangle's level of detail

A triangle's LOD was set to the highest level that a user can perceive to avoid unnecessary details and guarantee fidelity. The limit on perception comes from CSF (see Sect. 2). This function takes several variables that affect perception and produces the contrast levels at which the user stops perceiving. Figure 14 shows the visibility limit as a function of spatial frequency and contrast. The contrast threshold is the minimum contrast needed for a feature to become visible to the

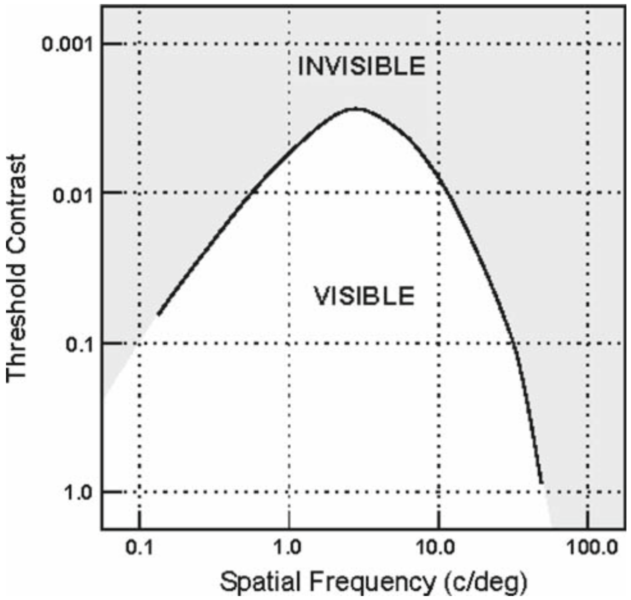

Fig. 14 Contrast sensitivity function showing the relationship between contrast and spatial frequency with human perception

human eye. Increasing the spatial frequency (i.e., by making triangles smaller) or reducing the contrast, can make features disappear. Eccentricity and velocity change the shape of the CSF curve and these relations are captured by the following equation [11]:

$$C(\alpha_{\max}, \nu, e) = \frac{(250.1 + 299.3|\log_{10}(\nu/3)|^3)\nu\alpha_{\max}^2 10^{-5.5\alpha_{\max}(\nu+2)/45.9}}{1 + 0.29e} \tag{3}$$

where ν, e, $\alpha_{\max}$ denote the velocity, eccentricity, and maximum visible spatial frequency respectively. Using the CSF equation, it becomes possible to predict CSF under given eccentricity and velocity values. Once the contrast, velocity, and eccentricity of a triangle are known, inversion of the equation leads to the maximum spatial frequency required to draw the triangle. Given the maximum frequency value $\alpha_{\max}$ and the measured spatial frequency α of the triangle, the desired LOD level, or equivalently the number of subdivisions that needs to be applied on the triangle, can be calculated as follows:

$$LOD = \left\lceil log_2\left(\frac{\alpha_{\max}}{\alpha}\right)\right\rceil \tag{4}$$

The base of the logarithm in Equation 4 is related to the subdivision algorithm employed [10] with each subdivision approximately doubling the spatial frequency.

5.4 Summary of steps

The operation of the PLOD system can be summarized by the following processing steps which are applied on all the visible triangles in the base model:

(1) locate triangle's centroid;
(2) project and transform the centroid to the screen plane;
(3) calculate angular offset with respect to the POI;
(4) calculate corrected eccentricity using Eq. (1);
(5) calculate spatial frequency by projecting the vertices of the triangle;
(6) render triangle;
(7) read back a 10 × 10 region around the triangle's centroid to determine the contrast using Eq. (2);
(8) compute velocity;
(9) use contrast, velocity and eccentricity and Eq. 3 to find the maximum spatial frequency;
(10) using maximum frequency, find the triangle's LOD using Eq (4).

6 Subject tests

Our testing procedure aims to measure the benefits of using uncertainty corrections in a perceptually optimized display system. In our tests, users were shown a terrain (see Fig. 15) where the rendering optimizations were applied. To control the POI on the screen, the user moves his/her head around. At the same time, the user tries to qualitatively assess the fidelity of the simulation by classifying perceived satisfaction in one of the following three categories: low, medium, or high. At the end of the experiment, we counted the number of users that perceived an increase, decrease or no change in satisfaction and expressed the result as a percentage.

We used three test cases corresponding to different views of the same terrain (i.e., Fig. 15a–c). The test cases differ from each other in the angle and distance to the model in order to see different sections of it. The idea is to find consistent results with varying geometry. The first test case (i.e., Fig. 15a) shows an overview of the terrain from afar and covering the whole image. The second test case (i.e., Fig. 15b) offers a close view of the terrain from the left with a mountainous landscape dominating the picture. The last test case (i.e., Fig. 15c) contains a mix of plains (foreground) and mountains (background).

Each case had different characteristics such as number of triangles, display resolution, and terrain roughness. Each case was displayed under three scenarios, with different optimizations enabled. The first scenario does not involve uncertainty information as part of LOD computations. The second scenario uses a fixed covariance matrix, corresponding to the maximum variance measured in our experiments. The result

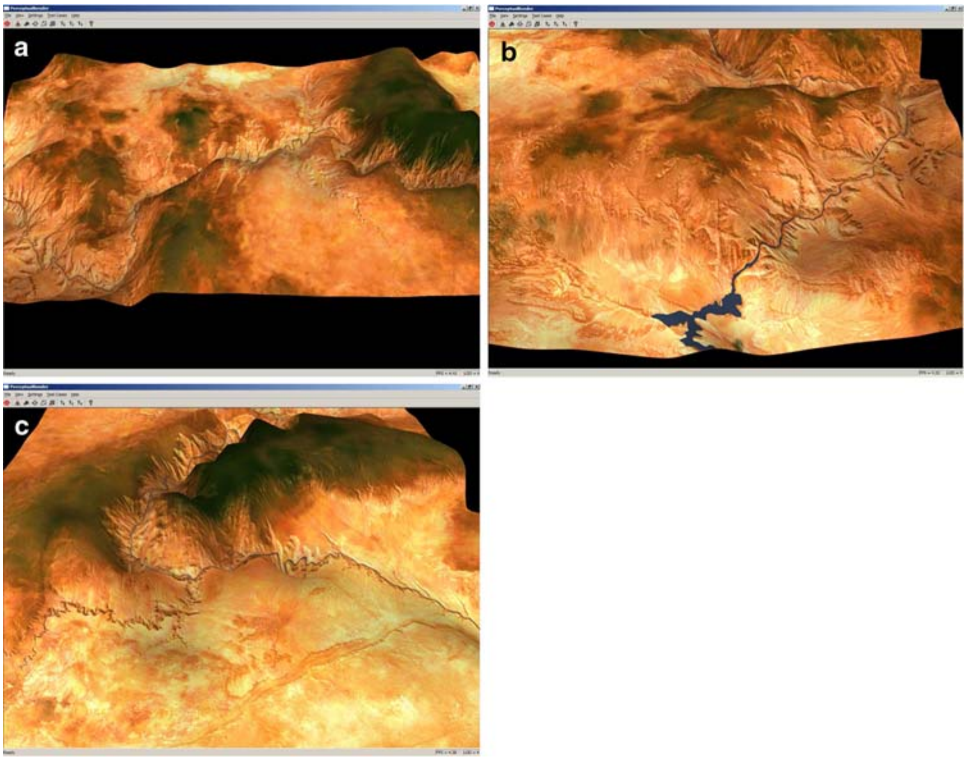

Fig. 15 Terrain view used for testing: **a** test case 1; **b** test case 2; **c** test case 3

is an expanded high definition area in the image. By expanding this area, the user is less likely to notice triangles changing resolution in the periphery. The last scenario uses an adaptive covariance matrix, updated at each frame.

The image shown in Fig. 16 is a view of the terrain from Fig. 15(b), rendered in wireframe mode, with perceptual optimizations on and an adaptive covariance matrix. It should be noted that the triangles closer to the POI, marked with a crosshair, are rendered with higher LOD. Also, the LOD regions do not follow an elliptical pattern because they are affected by the contrast level of the textured triangles and the relative velocity to the head.

Our experiments were performed using 19 subjects. Comparisons between different scenarios were performed and the increase/decrease in user satisfaction between test scenarios was recorded. The results are shown in Tables 1, 2, 3, 4. Table 1 shows the averages over all test cases, while Tables 2, 3, 4 report specific results for each test case separately.

Table 1 Satisfaction comparison between test scenarios across all test cases

All cases	Increase (%)	No change (%)	Decrease (%)	Total (%)
Fixed vs. none	63.16	29.82	7.02	100.00
Variable vs. none	26.32	54.39	19.30	100.00
Variable vs. fixed	8.77	33.33	57.89	100.00

Table 2 Satisfaction comparison between test scenarios for test case 1

Case 1	Increase (%)	No change (%)	Decrease (%)	Total (%)
Fixed vs. none	52.63	47.37	0.00	100.00
Variable vs. none	21.05	52.63	26.32	100.00
Variable vs. fixed	10.53	21.05	68.48	100.00

Table 3 Satisfaction comparison between test scenarios for test case 2

Case 2	Increase (%)	No change (%)	Decrease (%)	Total (%)
Fixed vs. none	63.16	31.58	5.26	100.00
Variable vs. none	21.05	63.16	15.79	100.00
Variable vs. fixed	0.00	42.11	57.89	100.00

Table 4 Satisfaction comparison between test scenarios for test case 3

Case 3	Increase (%)	No change (%)	Decrease (%)	Total (%)
Fixed vs. none	73.68	10.53	15.79	100.00
Variable vs. none	36.84	47.37	15.79	100.00
Variable vs. fixed	15.79	36.84	47.37	100.00

Table 1 shows that using a fixed covariance matrix to represent uncertainty has a positive impact on performance with 63% increase in user satisfaction. Only 7% of the time

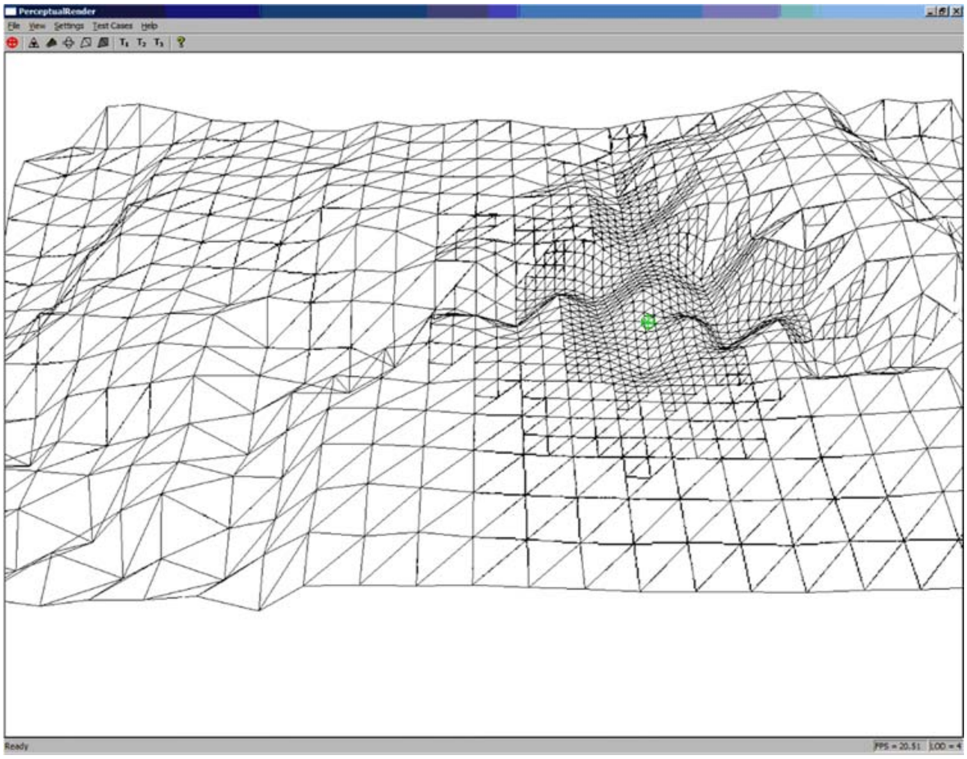

Fig. 16 The terrain, in wireframe mode (test case 2), rendered with perception optimizations and adaptive uncertainty

subjects perceived worse performance compared to not having uncertainty optimizations enabled. The results using adaptive covariance matrices were a bit different; the majority of subjects experienced an improvement or no change while almost 20% reported worse performance compared to using no uncertainty. Comparing adaptive with fixed uncertainty, the former performs up to 57% worse. A similar trend was observed in all three cases as shown in Tables 2, 3, 4).

A closer inspection of our experimental results revealed that the main reason for the under-performance of the adaptive uncertainty approach was the jitter in the estimation of the covariance matrix. This translated into LOD oscillations for some of the triangles at the visual periphery, where these oscillations are more likely to be noticed. We think that the cause for the jitter in the calculation of the covariance matrices was the random sampling strategy used to sub-sample the clouds of points (see Sect. 4.3.2).

The sub-sampling rate is an important parameter in this processing step. In our experiments, it was determined experimentally as the minimum fraction of points for which the covariance matrix was relatively constant. Figure 17 shows the variance of orientation angles as a function of the sub-sampling rate. As the sub-sampling rate increases, the standard deviation of the orientation estimates saturates to a constant value at a range of less than 0.2°. We used a sampling ratio of 0.3 in our experiments; however, very small perturbations in the correlation matrix entries result in noticeable oscillations in LOD.

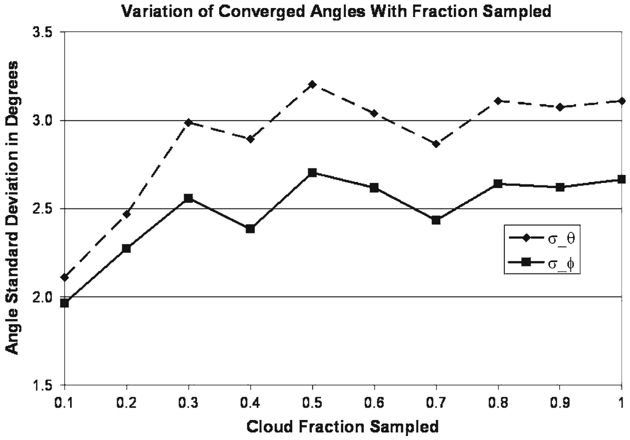

Fig. 17 Horizontal (Θ) and vertical (Φ) angle variances as a function of the sub-sampling rate

Ideally, we would need to use a fraction much closer to 1, however, such a high value would severely degrade user satisfaction due to lags in frame updating. A straightforward solution to this problem would be increasing processing power to minimize lag. Moreover, replacing the uniform distribution assumption during sub-sampling with a more realistic

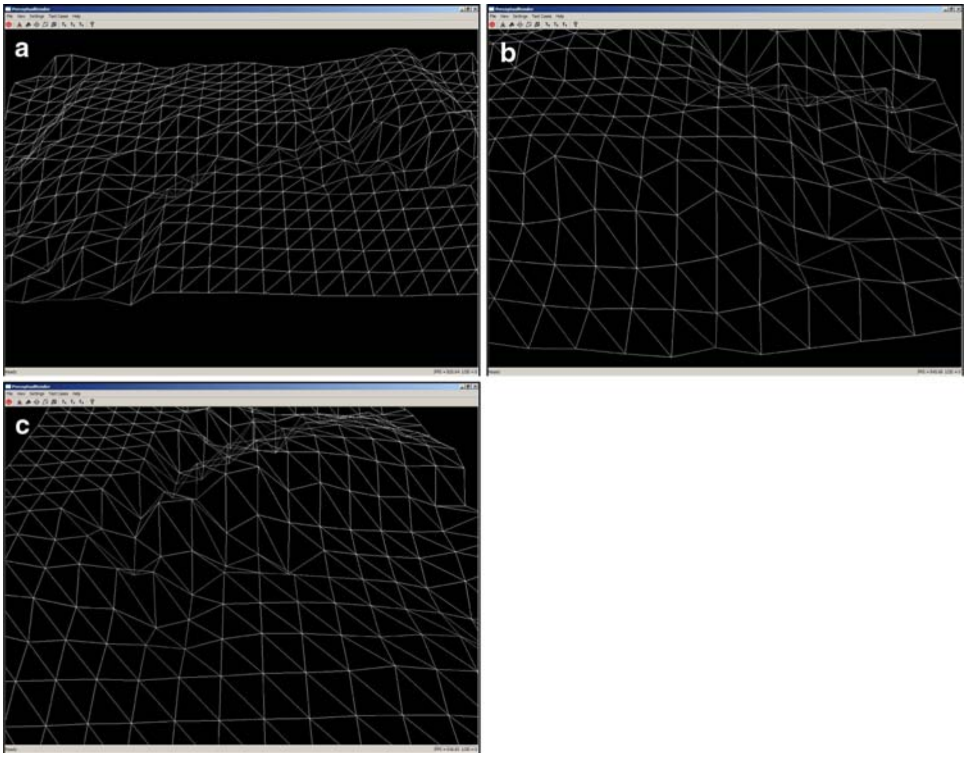

Fig. 18 Wireframe mode of terrain rendered at level 0: **a** test case 1, **b** test case 2, **c** test case 3

distribution that depends on the geometry of the markers might help in minimize jitter effects without major increases in computational load.

Although it is not evident from the above trends, the effect of dynamically changing resolutions was accentuated in test case 2. If we look at the responses received, test case 2 had the worst perceived results compared to the other two tests. The reason is the combination of the LOD and the high relief of the terrain section displayed. A terrain section with high relief will present abrupt transitions at the lower levels which are precisely the ones selected by the perceptual algorithm in test case 2. Figure 18(b) corresponds to test case 2 and should be compared to the other two test cases in Fig. 18(a), (c). The high relief and relative size of the triangles should be noted in this case.

As expected, using PLOD increased rendering speed considerably compared to rendering the scene at the highest LOD. Specifically, the frame rate increased from 5 fps to 15 fps on a Pentium 4 2.56 MHz processor with 1 GB of RAM. The savings from rendering extra triangles were used to compensate for the computational cost of implementing the PLOD system by increasing the frame rate. In general, higher frame rates have a positive impact on user quality perception. This did not have an effect on the experiments because the test cases were explicitly designed with static content to avoid distracting the user and making him shift his POI.

7 Conclusions

We have presented a PLOD-based display system for VEs using modest off-the-shelf components. The system was built using a modular design that combines a computer-vision-based head-pose estimation module, a PLOD module, and a rendering module. A key contribution of this study was the incorporation of head pose estimation uncertainties in the LOD computations. Subject tests performed to quantify the impact of uncertainty in perceptual optimizations indicated improvements in user satisfaction. A significant frame rate increase was also observed compared to rendering the scenes with no optimizations. For future research, we plan to improve our system by considering more sophisticated approaches for head tracking and pose estimation as well as estimating eye-gaze more accurately, that is, combining the orientation of the head with the orientation of the eyes in their sockets. For this, we plan to attach tiny, IR-illuminated, cameras on the frame of the eye-glasses, looking at the reflection of the eyes on the inside of the eye-glass. Also, we plan to eliminate the undesired jitter effect in the computation of the covariance matrix by investigating different sampling strategies and smoothing algorithms. Finally, we plan to investigate the issue of uncertainty estimation in more detail, for example, by considering non-uniform sampling schemes for error estimation as well as the effect of non-random errors such as calibration errors.

Acknowledgements This work was supported by NASA under grant # NCC5-583.

References

1. Luebke, D., Hallen, B., Newfield, D., Watson, B.: Perceptually driven simplification using gaze-directed rendering. Tech. Rep. CS-2000-04, University of Virginia, 2000
2. Foxlin, E.: Motion tracking requirements and technologies. In: Stanney, K.M. (ed.) Handbook of Virtual Environments: Design, Implementation, and Applications. Lawrence Erlbaum Associates, New Jersey (2002)
3. Smith, J., Gore, B., Dalal, M., Boyle, R.: Optimizing biology research tasks in space using human performance modeling and virtual reality simulation systems here on earth. In: 32nd International Conference on Environmental Systems, (2002)
4. Twombly, A., Smith, J., Montgomery, K., Boyle, R.: The virtual glovebox (vgx): a semi-immersive virtual environment for training astronauts in life science experiments. J. Syst. Cybern. Inf. **2**(3), 30–34 (2006)
5. Martinez, J., Erol, A., Bebis, G., Boyle, R., Towmbly, X.: Rendering optimizations guided by head-pose estimates and their uncertainty. In: International Symposium on Visual Computing (ISVC05), (LNCS, vol 3804), Lake Tahoe, NV, December 2005
6. Duchowski, A.: Acuity-matching resolution degradation through wavelet coefficient scaling. IEEE Trans. Image Process. **9**(8), 1437–1440 (2000)
7. Perry, J., Geisler, W.: Gaze-contingent real-time simulation of arbitrary visual fields. SPIE: Hum. Vis. Electron. Imaging (2002)
8. Clark, J.: Hierarchical geometric models for visible surface algorithms. Commun. ACM **19**(10), 547–554 (1976)
9. Ohshima, T., Yamamoto, H., Tamura, H.: Gaze-directed adaptive rendering for interacting with virtual space. In: VRAIS '96: Proceedings of the 1996 Virtual Reality Annual International Symposium (VRAIS 96), p. 103. IEEE Computer Society, (Washington, DC, 1996)
10. Junkins, S., Hux, A.: Subdividing reality. Intel Architecture Labs White Paper (2000)
11. Luebke, D., Watson, B., Cohen, J., Reddy, M., Varshney, A.: Level of Detail for 3D Graphics. Elsevier Science Inc., New York (2002)
12. Luebke, D.: A developer's survey of polygonal simplification algorithms. IEEE Comput. Graph. Appl. **21**(3), 24–35 (2001)
13. Reddy, M.: Perceptually modulated level of detail for virtual environments. CST-134-97. PhD thesis, University of Edinburgh (1997)
14. Williams, N., Luebke, D., Cohen, J., Kelley, M., Schubert, B.: Perceptually guided simplification of lit, textured meshes. SI3D '03: Proceedings of the 2003 symposium on Interactive 3D graphics, pp. 113–121, ACM Press, New York (2003)
15. Murphy, H., Duchowski, A.: Gaze-contingent level of detail rendering. In: EuroGraphics Conference, (Las Vegas), September 2001
16. Gee, A., Cipolla, R.: Determining the gaze of faces in images. Image Vis. Comput. **30**, 639–647 (1994)
17. Fu, Y., Huang, T.: hmouse: head tracking driven virtual computer mouse. IEEE Workshop Appl. Comput. Vis., 2007.
18. Hu, Y., Chen, L., Zhou, Y., Zhang, H.J.: Estimating face pose by facial asymmetry and geometry. IEEE Int. Conference on Automatic Face and Gesture Recognition, (2004)
19. Tao, H., Huang, T.: Explanation-based facial motion tracking using a piecewise bezier volume deformation model. Comput. Vis. Pattern Recognit. **1**, 611–617 (1999)
20. Bouguet, J.: Camera calibration toolbox for matlab
21. Ma, S.D.: Conics-based stereo, motion estimation and pose determination. Int. J. Comput. Vis. **10**(1), 7–25 (1993)
22. Ji, Q., Haralick, R.: Error propagation for computer vision performance characterization. In: International Conference on Imaging Science, Systems, and Technology, Las Vegas, (1999)
23. Murray, D., Little, J.: Patchlets: Representing stereo vision data with surface elements. In: Workshop on the Applications of Computer Vision (WACV), pp. 192–199, 2005
24. Trucco, E., Verri, A.: Introductory Techniques for 3-D Computer Vision. Prentice Hall, 1998
25. Duda, Hart, and Stork, *Pattern Classification*. John-Wiley, (2001)
26. Arun, K., Huang, T., Blostein, S.: Least-squares fitting of two 3-d point sets. IEEE Trans. Pattern Anal. Mach. Intell. **9**(5), 698–700 (1987)

Author biographies

Javier E. Martinez received his B.S. degree in chemical engineering from the Universidad Simón Bolívar, Venezuela in 1999. He then came to the United States to pursue a M.S. degree in Chemical Engineering and Computer Science at the University of Nevada, Reno. He currently works as a Software Engineer at Intel. His research interest interest include Rendering optimizations, Image processing, Human–computer interaction and system simulation. In 2003, he received a NASA EPSCoR award for his research in color image segmentation.

Ali Erol received the B.S degree in Electrical and Eloctronics Engineering from Bilkent University, Turkey in 1991 and the M.Sc. and Ph.D. degrees in Electrical and Eloctronics Engineering from Middle East Technical Univeristy, Turkey in 1995 and 2001, respectively. He worked as a post-doctoral fellow in the Computer Vision Laboratory (CVL) of the Department of Computer Science and Engineering at the University of Nevada, Reno (UNR). Currently, he is working as a research scientist in Ocali Software, Turkey for the development of an image-based 3D reconstruction software. His research interests include Computer vision, Image processing and Pattern recognition.

George Bebis received the B.S. degree in Mathematics and M.S. degree in Computer Science from the University of Crete, Greece in 1987 and 1991, respectively, and the Ph.D. degree in Electrical and Computer Engineering from the University of Central Florida, Orlando, in 1996. Currently, he is an Associate Professor with the Department of Computer Science and Engineering at the University of

Nevada, Reno (UNR) and Director of the UNR Computer Vision Laboratory (CVL). His research interests include Computer vision, Image processing, Pattern recognition, Machine learning, and Evolutionary computing. His research is currently funded by NSF, NASA, ONR, and Ford Motor Company. Dr. Bebis is an Associate Editor of the Machine Vision and Applications Journal, and serves on the Editorial Board of the Pattern Recognition Journal and the International Journal on Artificial Intelligence Tools. He has served on the program committees of various national and international conferences, and has organized and chaired several conference sessions. In 2002, he received the Lemelson Award for Innovation and Entrepreneurship. He is a member of the IEEE and the IAPR Educational Committee.

Richard Boyle received a B.A. in (Bio) Psychology from the University of Colorado, Boulder, an M.Sc. in Physiology from McGill University, Montreal, Canada, and a Ph.D. in Biological Sciences from the Scuola Normale Superiore, Pisa, Italy. Currently, he is Director of the BioVIS Technology Center at NASA Ames. BioVis does biology research and develops advanced visualization, imaging, simulation and computer know-how to support the goals of NASA's life sciences and space biosciences programs. In the past, he has been on the faculty of the Department of Otolaryngology/Head–Neck Surgery at the Oregon Health Sciences University (OHSU) in Portland, the Departments of Physiology and Pharmacology and Neurosciences Graduate Program at OHSU, and adjunct scientist at the R. S. Dow Neurological Sciences Institute (formally of the Good Samaritan Hospital, Legacy Health System, and now OHSU). Since 1984, he has regularly conducted hair cell studies at the Marine Biological Laboratory, Woods Hole, Massachusetts. Dr. Boyle has presented and published more than 140 papers and abstracts on neural mechanisms of cardiovascular regulation; Spinal cord physiology; Cerebellar mechanisms controlling head posture, Optokinetic and smooth pursuit ocular pursuit in behaving animals; Peripheral and central vestibular morphophysiology in fish and primates; Neural regeneration; and Biophysical mechanisms of hair cell function. He has been an invited speaker at over 50 institutions in the United States, Europe, Russia, and Japan.

Xander Twombly received the B.A. degree in Physics from Reed College, Portland, in 1987 and the Ph.D. degree in Biophysics and Computational Neuroscience from The Johns Hopkins University, Baltimore, in 1997. Currently, he is a Staff Scientist with the Research Institute for Advanced Computer Science (RIACS) at the NASA Ames Research Center, Moffett Field, CA and technical area lead in Discovery and Systems Health (DaSH). His research interests multi-source data fusion, Bayesian modeling, Pattern recognition, Image processing and Adaptive control systems. Recent research has focused on physics-based modeling of wiring insulation failure in aging aircraft and Bayesian inversion techniques to localize and classify fault types.