NuNav3D: A Touch-less, Body-driven Interface for 3D Navigation

Charilaos Papadopoulos∗ Daniel Sugarman† Arie Kaufman‡

Computer Science
Stony Brook University

ABSTRACT

We introduce NuNav3D, a body-driven 3D navigation interface for large displays and immersive scenarios. While 3D navigation is a core component of VR applications, certain situations, like remote displays in public or large visualization environments, do not allow for using a navigation controller or prop. NuNav3D maps hand motions, obtained from a pose recognition framework which is driven by a depth sensor, to a virtual camera manipulator, allowing for direct control of 4 DOFs of navigation. We present the NuNav3D navigation scheme and our preliminary user study results under two scenarios, a path-following case with tight geometrical constraints and an open space exploration case, while comparing our method against a traditional joystick controller.


1 INTRODUCTION & RELATED WORK

Traditionally, navigation interfaces for VR and Immersive scenarios require some sort of tracked navigation device, prop or marker-based tracking system. World-in-Miniature [9] requires a projection either inside the VR environment or on a surface and must be manipulated by touch or props. The concept of gesture based navigation for VR was discussed in [5] and [7]. Head-directed navigation [3] maps head orientations to navigation operations, however it does not permit looking at a direction different to that of the movement vector. Hand-tracking has been used to allow the user to grasp two virtual objects and either orbit around them or pull himself towards them [8] and pointing towards a destination in the virtual space[6]. Locomotion modalities such as Walking-in-Place [10] also allow navigation on a 2D plane while providing for a greater sense of presence than alternatives. LaViola et al. [4] have proposed a merging of foot gestures with rotational tracking of the torso for hands-free navigation in VR. The Pengufly [11] paradigm utilizes 2D projections of the tracked positions of the user’s hands and head to define a navigation vector.

As already stated, all methods described above require some degree of marker-based tracking of either a prop or the user himself to be implemented, which might be infeasible under certain situations. Additionally, most techniques limit the number of DOFs exposed and potentially constrain the user on navigating on a 2D plane. Our proposed technique, NuNav3D, utilizes body pose recognition driven from a single depth camera [1] [2], and directly maps rotations and translations to simple hand movements, for simultaneous manipulation of a total of 4-6 DOFs without requiring any tracking markers or props.

∗ e-mail: cpapadopoulos@cs.stonybrook.edu
† e-mail: dsugarman@cs.stonybrook.edu
‡ e-mail: ari@cs.stonybrook.edu

2 THE NU NAV3D PARADIGM

NuNav3D operates on poses. We have developed a simple pose recognition scheme, based on the concept of pose uniformization. A uniform pose $P_{\text{Uni}}$ results from a pose $P$ by placing the skeleton at the center of the coordinate system, compensating for torso rotation by applying the inverse rotation to all skeletal joints and then normalizing all bone segments to unit length. Pose recognition is performed by comparing two poses in uniform space using the metric $s_{P_1, P_2} = \frac{1}{N} \sum_{i=0}^{N} \{ |P_{\text{Uni}}^i - P_{\text{Uni}}^j| \} \leq s_{\text{Thres}}$ (where $s_{\text{Thres}}$ being the pose recognition threshold scalar) and utilizing a simple voting scheme over a sliding window of poses. The above method appears to be relatively robust against variance in body geometry.

We define a navigation pose as illustrated in Figure 1. Once the navigation pose is detected, the system stores the user’s position (defined as $P_{\text{Nav}}$), relaxes $s_{\text{Thres}}$ and on each follow-up update calculates $\vec{v}_{\text{Offset}} = \vec{P}_{\text{Nav-Current}} - \vec{P}_{\text{Nav-Base}}$ and $\vec{v}_{\text{RightHand}} = \vec{P}_{\text{Nav-Current}} - \vec{P}_{\text{Nav-Base}}$. (which are also scaled based on the relaxed pose recognition threshold to achieve roughly unit length).

Figure 1: An example NuNav3D navigation pose, with the user’s arms parallel to the torso and the forearms perpendicular to the arms. The system stores a snapshot of the user’s pose $P_{\text{Base}}$ when the navigation pose is detected and calculates offset vectors for the user’s left and right hands which are mapped to translations and rotations respectively.

On each update, $\vec{v}_{\text{LeftOffset}}$ and $\vec{v}_{\text{RightOffset}}$ are used to manipulate a virtual camera. Assuming a virtual camera with position $C_{\text{Nav}}$ and basis vectors $\hat{C}_x, \hat{C}_y, \hat{C}_z$, for time-steps $t$ and $t + 1$, we define a translation vector $T_{t+1} = (\vec{v}_{\text{LeftOffset}} \cdot \hat{C}_z) \cdot \kappa_{\text{Translate}}$ and $\vec{v}_{\text{RightOffset}} \cdot \hat{C}_z$ with $\kappa_{\text{Translate}}$ being a ramp exponent. The position of the virtual camera $C$ is defined as $C_{t+1} = C_t + T_{t+1} \cdot \hat{C}_z$ with $\kappa_{\text{Translate}}$ being a scalar scaling factor. Effectively this maps the left hand’s vertical hand movements to translations along the camera’s view direction and horizontal movements to strafing. Rotations are handled by defining rotation transformations $R_y$ and $R_x$ along the camera’s y and x axis (for yaw and pitch respectively). The angle value for $R_y$ is equal to $-(\vec{v}_{\text{RightOffset}} \cdot \hat{C}_x)$ (similarly for the $x$ axis). The quaternions are composed to $R_{t+1}$ and used to rotate $\hat{C}_x$ and $\hat{C}_y$. Effectively, this maps horizontal movements of the right hand to cam-
era yaw and vertical movements to camera pitch. For both translation and rotation, if the offset vector magnitude is smaller than an experimentally defined threshold, no transformation occurs (thus filtering out minor, naturally occurring, hand movements). Figure 2 graphically demonstrates the above mapping.

Figure 2: An indicative application of offset vectors, manipulating a virtual camera. The top subfigure illustrates the calculation of the per-hand offset vectors, while the bottom subfigure shows a third-person view of a virtual camera at time step $t$ (before applying the offset vectors) and $t+1$, after applying the corresponding transformations. The resulting rotation, denoted by the blue shaded area, is the difference in orientation for the $C_x$ basis vector between $t$ and $t+1$.

3 Evaluation

We conducted a user study, comparing NuNav3D against a conventional joystick controller under a general, open-space exploration task (discovery of hidden objects) and a path-following task with strict geometrical constraints (which was based on simulated virtual colonoscopy data). Figure 3 shows two indicative views of the trial datasets. Our quantitative metric was elapsed time (ET). Subjects were asked to rate their familiarity with 3D navigation on a scale of 1 to 5 (5 being the most experienced). The study was conducted on a total of 12 individuals, with a mean age of 24.1 (SD 1.89). The mean 3D navigation experience level was 3.25 (SD 1.65).

Figure 3: Path-following data set, inspired by virtual colonoscopy (top), and open space exploration dataset (bottom, inset denotes hidden object positions and starting camera position).

Over the entirety of study subjects, NuNav3D presented lower performance than the joystick. Specifically, for the path-following task, NuNav3D was on average 79.2% (SD 24.1%) slower in ET. In contrast, users in the 3-5 experience bracket were 100.9% (SD 83.9%) slower in the path following task and 45% (SD 82.2%) slower in the exploration task.

The subjects were also asked post-study qualitative evaluation questions, including choice of interface for usage in the future and which interface they found to be less intrusive with the experience. 66.6% commented that NuNav3D was less intrusive and 50% stated that they would choose to use NuNav3D in the future.

4 Discussion and Future Work

While the sample size of the user study was not large enough to draw conclusions, it provides interesting indications for directing future work. For non-expert users, who were unfamiliar with the joystick controller as well as NuNav3D, performance was comparable in the open-space exploration task. Going forward we plan on investigating solutions for the over- translating issue observed in the path-following task, including ways to visually represent the navigation data to the user, as well as options in applying the navigation offset vectors to the virtual camera. Additionally, we plan on conducting a much more thorough experimental evaluation on the system, including reviewing user fatigue over time and exploring other metrics such as simultaneous DOF utilization and precision in movement. Finally we wish to explore a multi-modal interaction scheme that includes NuNav3D for navigation and other body-driven modalities for tasks such as picking, etc.

Acknowledgements

We wish to thank AMD for providing the FirePro GPU and AVNet for the tiled-display array used in our experimental setup. This work was supported by NSF grants IIS0916235 and CNS0959979 and NIH grant R01EB7530.

References