Practical chirocentric 3DUI platform for immersive environments

C. Papadopoulos, *Student Member, IEEE*[*] H. Choi[†] J. Sinha [‡] K. Yun[§]
A. E. Kaufman, *Fellow, IEEE*[¶] D. Samaras, *Member, IEEE*[‖] B. Laha, *Member, IEEE*[**]

Department of Computer Science, Stony Brook University

Abstract

Chirocentric 3D user interfaces are sometimes hailed as the “holy grail” of human-computer interaction. However, implementations of these UIs can require cumbersome devices (such as tethered wearable datagloves), be limited in terms of functionality or obscure the algorithms used for hand pose and gesture recognition. These limitations inhibit designing, deploying and formally evaluating such interfaces. To ameliorate this situation, we describe the implementation of a practical chirocentric UI platform, targeted at immersive virtual environments with infrared tracking systems. Our main contributions are two machine learning techniques for the recognition of hand gestures (trajectories of the user’s hands over time) and hand poses (configurations of the user’s fingers) based on marker clouds and rigid body data. We report on the preliminary use of our system for the implementation of a bimanual 3DUI for a large immersive tiled display. We conclude with plans on using our system as a platform for the design and evaluation of bimanual chirocentric UIs, based on the Framework for Interaction Fidelity Analysis (FIFA).

Keywords: User Interface, Immersive Virtual Environment, Chirocentric, Natural User Interfaces.

1 Introduction

Interaction with immersive visualization systems (CAVEs, HMDs, etc) is generally conducted via dedicated devices, such as tracked wands. These devices provide tactile buttons that can be used to either trigger specific actions or enter and exit various manipulation states. In the majority of recent systems, tracking of the interaction props is provided via an infrared (IR) tracking system, through a commercial rigid-body solver that translates fixed arrangements of IR markers to positions and orientations within the tracking space.

These dedicated devices can instead be replaced with hand gestures and hand poses, creating effectively a *chirocentric* user interaction experience. Chirocentric UIs have received some exposure in scientific and popular venues. Existing systems however present a number of drawbacks (limited exposition on their implementation, narrow scope and/or need for additional active devices such as datagloves). These drawbacks impact the formal evaluation of interface designs based on these chirocentric technologies, due to their effect on ergonomics, breadth of supported interaction modalities or lack of reproducibility.

In this technote, we introduce two algorithms that enable the implementation of such a practical chirocentric user interface, within the constraints imposed by a commercial IR tracking system. The first algorithm is targeted at the recognition of unimanual or bimanual hand gestures. Our second algorithm tackles the problem of hand pose recognition from sparse point clouds provided by the IR tracking system, generated by low-cost gloves with attached retroreflective markers. Utilizing the platform defined by these two techniques, we have developed a prototype chirocentric user interface for the exploration of 2D and 3D data within immersive environments. Our system exposes unimanual and bimanual manipulative interactions. We discuss the implementation details of our system and present various insights gained through its development and deployment within a large tiled display. We conclude with an outline of future plans for utilizing our platform for the formal evaluation of bimanual chirocentric 3DUIs under the Framework for Interaction Analysis (FIFA) [11].

2 Related Work

Unencumbered, hand-driven (or chirocentric) user interfaces are, in some ways, the holy grail of UI research. In the early 1980s, Bolt incorporated gestural input in his *”put that there”* experiment [5]. Later, Baudel and Beaudouin-Lafon [3] described a prototype system that used a wired DataGlove in order to expose a set of gestural commands to the user for controlling a presentation. Importantly, they also outlined one of the earlier models for defining gestural commands in chirocentric interfaces. Overall, such interfaces, especially ones that support bi-manual interactions, have been shown to positively affect user performance in spatial tasks. For instance, Hinckley et al. [9] showed that bimanual manipulations are superior to single-handed implementations both in terms of time-saved and also because they improve performance at the cognitive level (allowing one hand to define a frame of reference for the other and affording better separation of subtasks to individual appendages). Later work from Balakrishnan et al. [1] evaluated bimanual interfaces in the context of the relation between the interaction reference frame and the visual feedback space. Papadopoulos et al. [14] described a bimanual UI for navigating 3D scenes, dirven by a depth-sensor.

The above work has focused on relatively simple components of spatial natural user interface design. In fact, most implementations of such systems described in the literature are generally limited in scope or exposition of the technical details. Grossman et al. [7] described a simple chirocentric interface for gestural interactions with a volumetric display, which used an optical tracking system. Hackenberg et al. [8], in addition to describing a finger tracking pipeline from depth sensor data, also used their system as a backbone for a direct-manipulative 3D user interface. However, their approach is targetted at commercial depth sensors and assumes a frontal view of the user. In the commercial realm, Oblong Industries (`www.oblong.com`) has developed a chirocentric UI platform termed *“g-speak”*, which uses a high end IR tracking system. However, to our knowledge there exists no published work detailing the inner workings of the system. Based on g-speak, Zigelbaum et al. [16] implemented a user interface for the exploration of a data set of animated videos. Banerjee et al. [2] designed the *WaveForm* interface, aimed at Video Jockey-ing. They also offer little exposition in terms of the algorithm used for hand pose detection. More recently, Bogdan et al. [4] described and evaluated *HybridSpace*, a dual-modality interface which integrates 3D freehand input and 2D mouse manipulations. Still, the chirocentric aspect of HybridSpace was still relatively simple, limited to a pinch gesture. Levesque et al. [10] described a bimanual 3DUI for immersive virtual environments implemented via tracked data gloves.

[*] e-mail:cpapadopoulo@cs.stonybrook.edu
[†] e-mail:ggxfan@cs.stonybrook.edu
[‡] e-mail:joy.seawolf@gmail.com
[§] e-mail:kyun@cs.stonybrook.edu
[¶] e-mail:ari@cs.stonybrook.edu
[‖] e-mail:samaras@cs.stonybrook.edu
[**] e-mail:blaha@cs.stonybrook.edu

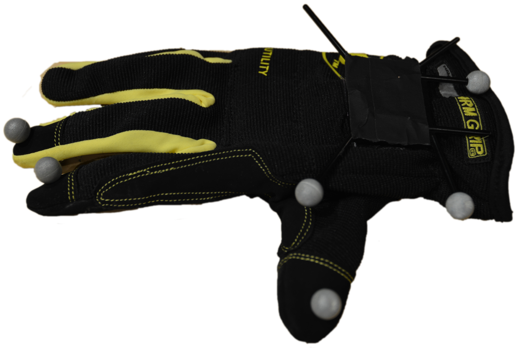

Figure 1: One of the low-cost, passively tracked gloves that we used with our interface. It is constructed by attaching a standard tracking system rigid body to the back of a soft glove, using wires running through the fabric to preserve its elasticity and ensure a deformation-resistant connection. Retroreflective markers are attached on the tips of the thumb, index and middle fingers. The total cost of materials for one glove is under $25.

3 Algorithmic Framework

Our chirocentric UI platform is driven by two algorithms that utilize prior knowledge for the recognition of hand poses and gestures based on input data from a tracking system. In the description of these algorithms below, rigid body positions for the head and hands are denoted $\mathbf{P_i}, \mathbf{i} \in \mathbf{H}, \mathbf{LH}, \mathbf{RH}$ and marker clouds are annotated as $\mathbf{M_i}$ (for the i-th marker of the cloud).

3.1 Hand Pose Recognition

For our hand pose recognition algorithm, we assume that the tracking system provides us with $\mathbf{P_H}$ which is the position of the hand (obtained by a rigid body mounted on the back of a simple glove). Additionally, we are provided with a marker cloud $Markers = \{\mathbf{M_0} \cdots \mathbf{M_n}\}$, which contains all markers reconstructed by the tracking system. This cloud includes markers that are mounted on the tips of the thumb, middle and index finger of each hand of the user. Our low-cost tracked glove is shown in Figure 1. In total, we are interested in the markers that correspond to the hand's rigid body (3 in our case), and the 3 finger tip markers. We construct a subset of $Markers$ termed $FilteredMarkers$ for each hand by filtering the totality of the market cloud based on proximity to $\mathbf{P_H}$. Following that, if the filtered vector contains fewer than 6 markers (due to occlusions), we append placeholder markers at the position $\mathbf{P_H}$. Finally, all markers are sorted based on their distance from $\mathbf{P_H}$ and the closest six markers are returned as $FilteredMarkers$. Given $FilteredMarkers$ we can then proceed to the feature calculation for a particular hand pose.

For a particular hand pose we construct feature vector $\mathbf{F^h}$ from markers $\mathbf{M_i} \in FilteredMarkers$ as $\mathbf{F^h}(\mathbf{i},\mathbf{j}) = \|\mathbf{M_i} - \mathbf{M_j}\|, \forall \mathbf{M_i}, \mathbf{M_j} \in FilteredMarkers$. Effectively, our feature vector is defined as the pairwise Euclidean distance between all markers. To ameliorate the lack of between-frame consistency in marker tracking, this feature vector is sorted prior to use. Its dimensionality is 36 (assuming 3 rigid body markers and 3 finger tip markers).

With the feature calculation defined, training and classification of hand poses are quite straightforward. We use a Support Vector Machine (implemented via the libSVM library) using a Radial Basis Function kernel. Our classification algorithm runs in real time (approximately 4 milliseconds per incoming frame of tracking data).

3.2 Gesture Recognition

We assume that $\mathbf{P_i}$ is the position of the i-th rigid body (or *joint*) in 3D space as reported by the tracking system (practically, we leverage 3 rigid bodies, for the head, left and right hands of the user but our feature calculation generalizes to an arbitrary number). In the rare occasion that a rigid body is not tracked during a particular frame, we replace it with a placeholder at the center of the coordinate system.

Our feature is a combination of the distance between joints and their motion, aggregated over a window of time. In contrast to our hand pose feature, which identifies static poses, without a progression component, this feature calculation allows us to capture the dynamics of a particular gesture as it advances through time. The algorithm described below is based on work by Yun et al. [15] in the field of activity recognition.

We augment the earlier notation by letting $\mathbf{P_{i,t}} \in \Re^3$ be the 3D location of joint i of the subject at time t. Let T be the set of all the frames within the size of a frame window, W. The feature of each such frame window is a single vector, defined as the concatenation of all computed features $\mathbf{F}(\cdot;\mathbf{t})$, where $t \in T$. In particular, we compute two sub features, one based on the pair-wise distance of joints for the current frame and the second based on the pair-wise distance of all pairs of joints in consecutive frames.

The *joint distance* feature $\mathbf{F^{jd}}$ is defined as the pairwise Euclidean distance between all the joints of a persons at time t. It is defined as $\mathbf{F^{jd}}(\mathbf{i},\mathbf{j};\mathbf{t}) = \|\mathbf{P_{i,t}} - \mathbf{P_{j,t}}\|$, where i and j are any joints of the user and $t \in T$.

The *joint motion* feature $\mathbf{F^{jm}}$ is defined as the Euclidean distance between all pairs of joints of a person at time t_1 and at time t_2. It captures dynamic motions between joints and formulated as $\mathbf{F^{jm}}(\mathbf{i},\mathbf{j};\mathbf{t_1},\mathbf{t_2}) = \|\mathbf{P_{i,t_1}} - \mathbf{P_{j,t_2}}\|$, where i and j are any joints of the user, $t_1, t_2 \in T$, $t_1 \neq t_2$.

The window W spans a total of 13 frames. In order to ensure that the between-timestep differences are substantial enough (since our tracking system delivers data at 120hz), we sample every 3rd frame of this 13 frame window. This value was chosen to balance the algorithm's performance, response time and the dimensionality of the feature vector. For each of the 5 sampled frames, we calculate the aforementioned joint distance feature, resulting in a total of 3 distances per frame (or 15 for the entire frame window). Additionally, for every combination of the 5 frames sampled from the window, we pick 10 pairs of frames (5 choose 2) as sources for our joint motion feature calculation. For every pair, we determined the euclidean distance between joint i of the 1^{st} element in the pair and all the 3 joints of the second element of the pair and hence we obtain a 9 dimensional vector for each pair and in totality we have a 90-dimensional joint motion vector extracted from a window of 13 frames. The dimensionality of the combined feature vector is $15 + 90 = 105$. The classifier is trained on a collection of gestures, including swipes, zooming, pointing, etc.

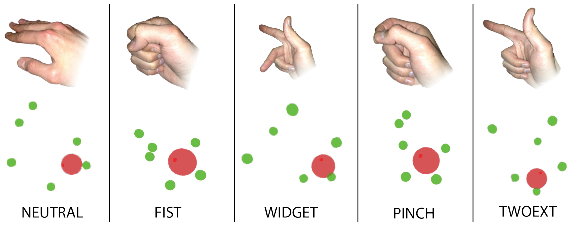

Figure 2: Samples of the various hand poses supported by our system. Top Row: Photographs of the poses performed by a user (without wearing the tracked glove). Bottom row: Equivalent tracking system data. The green spheres are markers and the single red sphere reports the rigid body position. Also visible are shorthand notations for the poses used through the paper.

4 Chirocentric 3DUI Prototype

We demonstrate our chirocentric user interface platform by implementing a bimanual 3DUI for control of visualizations on the Reality Deck [13]. Fig. 3 shows a user utilizing our prototype for interaction with a GIS application within our facility. We also invite readers to view the companion video to this technote as it demonstrates several of the supported interactions that are also outlined below.

4.1 Visualization platform and applications

The Reality Deck is a large immersive tiled display comprised out of 4 walls with a total of 416 monitors and an aggregate resolution of more than 1.5 gigapixels. It is utilized for various forms of scientific visualization, such as complex 3D proteomics structures

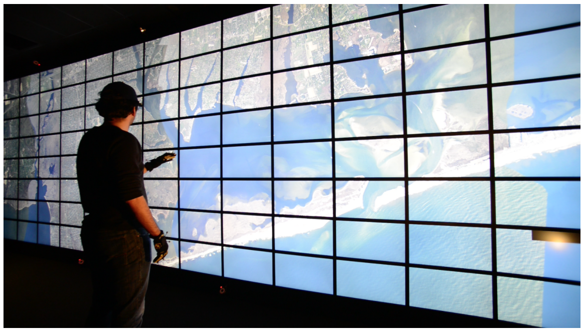

Figure 3: A user is leveraging our chirocentric user interface to explore a 2D GIS dataset on large immersive tiled display wall. The user has just performed a constrained direct-manipulative translation on the 2D map plane.

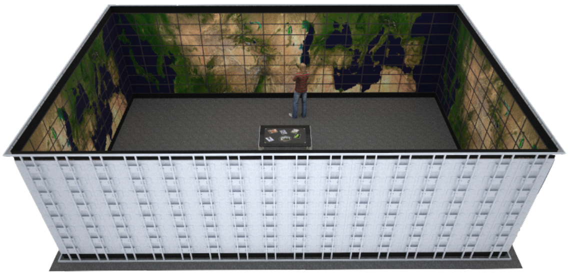

Figure 4: To-scale render of the Reality Deck, the immersive gigapixel resolution display, used as a visualization test-bed for our chirocentric 3D user interface.

and large scale geospatial information exploration. A view of our immersive gigapixel resolution display can be seen in Figure 4.

2D and 3D visualization manipulations are traditionally mapped to the translation of a physical input device (such as a mouse), with buttons acting as modifiers and affecting the active axes of manipulation. With our system, these physical modifiers are instead replaced by the user's hand poses, which place the system into a particular interaction state. The supported hand-poses, along with the short-hand notations used below, are summarized in Figure 2. Within that state, the user's hand motions are either directly applied to the virtual camera, or act as relative transforms that continuously affect the visualization until the interaction modality is terminated by the hands returning to their resting state. Other interactions are traditionally triggered via a selection on a graphical user interface (either on top of the visualization or on a second screen) or dedicated button presses. In our 3DUI prototype, such actions are activated by performing a gesture.

4.2 Supported Interactions

Our prototype supports a number of uni- and bi-manual 2D and 3D navigation interactions. In describing these interactions, we use the notation $\mathbf{P}^{\mathbf{t}}_{\mathbf{R}}$ and $\mathbf{P}^{\mathbf{t}}_{\mathbf{L}}$ for the positions of the user's right and left hand at time t as reported by our tracking system.

Unimanual 3 DoF Continuous Translation The system enters this mode when it detects the **WIDGET** pose on the primary hand. It sets $\mathbf{P}_{\mathbf{Center}} = \mathbf{P}^{\mathbf{t_0}}_{\mathbf{R}}$. In subsequent frames t', it calculates $\vec{\mathbf{v}} = \mathbf{P}^{\mathbf{t}'}_{\mathbf{R}} - \mathbf{P}_{\mathbf{Center}}$. $\vec{\mathbf{v}}$ is then applied as a translation vector to the virtual camera, translating it in 3D space. For example, if the hand is offset upwards from $\mathbf{P}_{\mathbf{Center}}$, the camera is continuously translated along its vertical axis. A small amount of tresholding is applied (approximately $2cm$) to ensure that natural motion while the user is at rest does not trigger an unintended interaction. This interaction mode continues until a pose other than **WIDGET** is detected on the primary hand.

Bimanual 4 DoF Continuous Translation and Rotation The system enters this mode when it detects the **WIDGET** pose on both hands. A continuous translation is mapped to the right hand, in the same way as described above. Additionally, the system sets $\mathbf{P}^{\mathbf{L}}_{\mathbf{Center}} = \mathbf{P}^{\mathbf{t_0}}_{\mathbf{L}}$ upon entering the state. For following frames, the system calculates $\vec{\mathbf{v}} = \mathbf{P}^{\mathbf{t}'}_{\mathbf{L}} - \mathbf{P}^{\mathbf{L}}_{\mathbf{Center}}$. The x (horizontal) component of $\vec{\mathbf{v}}$ is used to determine a rotation, which is continuously applied to the camera. This allows control of the camera's yaw, using the secondary hand. The system remains in this state until a pose other than **WIDGET** is detected on either hand. If **WIDGET** is maintained on the right hand while the left hand assumes **NEUTRAL**, then the system reverts to the above modality without resetting $\mathbf{P}^{\mathbf{R}}_{\mathbf{Center}}$.

Unimanual Continuous Flythrough This mode is triggered once **TWOEXT** is detected on the user's right hand. From then on, at every frame t, $\mathbf{P}^{\mathbf{t}}_{\mathbf{R}}$ is used along with the hand's orientation information to define a pointing direction $\vec{\mathbf{p}}_{\mathbf{physical}}$ within the virtual environment (this is one of a variety of ways to determine a user's intended pointing direction within the visualization space). $\vec{\mathbf{p}}_{\mathbf{physical}}$ is then transformed to the 3D scene's coordinate system, yielding $\vec{\mathbf{p}}_{\mathbf{virtual}}$ which is then applied as a per-frame translation to the virtual camera's position.

For our 2D GIS application, we expose the following functionality:

Unimanual Directly Manipulative Translation When the user's right hand is in the **FIST** pose, this mode is entered. Upon entry at time t, the system stores $\mathbf{P}^{\mathbf{prev}}_{\mathbf{R}} = \mathbf{P}^{\mathbf{t}}_{\mathbf{R}}$. At each subsequent frame t', the system calculates $\vec{\mathbf{v}} = \mathbf{P}^{\mathbf{t}'}_{\mathbf{R}} - \mathbf{P}^{\mathbf{prev}}_{\mathbf{R}}$. The x and y components of $\vec{\mathbf{v}}$ are then applied as a translation to the virtual camera, manipulating its position on the 2D plane (the z component is unused). Following the manipulation, the system updates $\mathbf{P}^{\mathbf{prev}}_{\mathbf{R}} = \mathbf{P}^{\mathbf{t}'}_{\mathbf{R}}$. This process continues for as long as **FIST** is maintained.

Bimanual Rotate-Scale-Translate This mode (which has also been referred to as "air multitouch" by some of our users) in triggered when both hands are in the **FIST** pose. At each incoming frame t, we calculate $\vec{\mathbf{diff}}^{\mathbf{t}} = \mathbf{P}^{\mathbf{t}}_{\mathbf{R}} - \mathbf{P}^{\mathbf{t}}_{\mathbf{L}}$, $\mathbf{M}^{\mathbf{t}} = \vec{\mathbf{diff}}^{\mathbf{t}}/2$ and the between-hand distance $d^t = \|\vec{\mathbf{diff}}^{\mathbf{t}}\|$. Based on these values and the previous frame's data, we can then define a translation vector $\vec{\mathbf{t}} = \mathbf{M}^{\mathbf{t}} - \mathbf{M}^{\mathbf{t-1}}$ which is used to translate the virtual camera. Additionally, we calculate a scale factor $z = d^t/d^{t-1}$ which is applied to the current zoom factor. Finally, a rotation value ϕ is applied to the camera based on the angle between $\vec{\mathbf{diff}}^{\mathbf{t}}$ and $\vec{\mathbf{diff}}^{\mathbf{t-1}}$. Effectively, our system mirrors traditional multitouch functionality. This manipulation continues until either hand exits the **FIST** pose. If the primary hand remains in **FIST**, the system transitions to the unimanual directly manipulative translation mode instead.

Unimanual Continuous Translation and Zoom This interaction mode is similar to the Unimanual 3 DoF Continuous Translation for 3D scenes that we described earlier. However, instead of directly applying $\vec{\mathbf{v}}$ to translate the camera position, only its x and y components are used to translate the camera along the 2D plane, while the z component is scaled and applied as an offset to the current zoom factor. Effectively, forwards/backwards offsets of the user's hand result in zooming in and out respectively.

Additionally, we correlate gestures to certain application-specific actions. For example, in a GIS application, left and right swipe gestures are used to sequentially cycle between a list of predetermined points of interest. Zoom-in and zoom-out gestures allow the user to instantly increment or decrement the current zoom level by one unit. The same zoom gestures performed along the vertical axis minimize and maximize the scale factor of the map.

4.3 Preliminary Observations

We report a number of anecdotal findings that arose from the utilization of our UI by the authors of this paper, as well as a small number of external users.

In 2D exploration scenarios, users are able to accurately navigate, using both the unimanual and bimanual directly manipulative modalities. In a way, these modalities are direct mappings of traditional single and multitouch interactions on modern tablets, making users more likely to be familiar with their operation. However, we received commentary that, for long exploration sessions (or when the traversal of a large amount of virtual space is required), these two modalities can impose additional user fatigue, as they demand multiple repeating arm motions. We reached the same conclusion early in the design process, which was one of the drivers for the addition of the unimanual continuous translation and zoom modality. Here users can just determine the direction and speed of translation by offsetting their hand, and the camera manipulation continues until they return to the **NEUTRAL** pose. Effectively, there exists a precision-versus-comfort tradeoff between these two modes of manipulation. Arguably, the comfort level for the **FIST** based manipulations can also be improved by implementing support for inertial camera manipulations in our visualization system.

For 3D navigation, our system exposes a powerful tool in the form of the bimanual translation and rotation feature. Effectively, it provides a total of 4 navigational DoFs, without a dedicated controller device. More experienced users were able to perform complex maneuvers within and around 3D structures with ease. For non-experts, this type of manipulation proved somewhat unwieldy, but we hypothesize that this may be related to a general lack of familiarity with 3D navigation in general. Originally, we attempted exposing additional degrees of rotation (camera pitch and roll) through this modality, but they proved to be overwhelming for almost all users. The continuous flythrough modality was found very intuitive to use, but it is naturally somewhat constrained in its functionality (particularly if the virtual environment is not fully immersive).

A point of contention is the selection of support gestures and hand-poses that can be recognized by the system. In our current implementation, various interactions were assigned to hand-poses and gestures somewhat arbitrarily. While some of these assignments make sense (for example the **FIST** pose, similar to a "grabbing" movement, triggering a direct manipulation) others may not (a vertical "zoom-out" gesture minimizing the zoom scale). Nielsen et al. [12] and several other scholars can provide guidance on this front when developing further UI prototypes. Additionally, the notion of frames of reference is extremely important, particularly in an immersive setting. Our existing implementation assumes that the user is aligned to the front wall of the facility. Consequently, interactions along the axes of the virtual camera map to hand motions along the width, depth and height of the physical space. In a practical setting, this assumption may not hold, since users can physically navigate and interact with the display from any point and with any body and head orientation. Consequently, the mapping between the physical interaction space, the visual feedback space and any manipulations is not well-defined for some modalities.

4.4 Plans for Formal Evaluation

Our primary motivation for the development of this chirocentric UI platform is the enablement of formal evaluation of various unencumbered bimanual interface designs. Our goal is to better inform the appropriateness of chirocentric UIs of different fidelity levels for various immersive applications. The Framework for Interaction Fidelity Analysis [11] provides strong guidance on this front by outlining three criteria areas for interaction fidelity (biomechanical symmetry, control symmetry and system appropriateness). Using our platform, we plan to implement numerous UI prototypes that satisfy these criteria to different extends and compare them in an experimental setting. Early work on FIFA has shown that higher levels of control symmetry positively affect user performance. But is this truly the case for bimanual chirocentric UIs? From early anecdotes, we have observed that a direct manipulative translation (which has high degrees of biomechanical and control symmetries versus the real-world interaction equivalent) can be more tiring for a longer user session than the continuous translation alternative. Such conundrums can be observed for a number of interaction modalities and gestures that are used as triggers for actions within the virtual world. Our goal is to explore whether natural user interaction is the ultimate interface modality [6].

5 Conclusion

In this paper, we described the development of a practical chirocentric UI platform, to be used for the development of bimanual 3DUIs. Our platform is based on two algorithms for the recognition of hand poses and hand gestures, the two main pillars of a chirocentric user experience. They are targeted at data provided by commercial IR tracking systems, which are usually found in immersive environments such as CAVEs and tiled displays. Using this platform, we developed a prototype 3DUI that exposes a number of uni- and bimanual modalities for the navigation of 2D and 3D data and a set of hand gestures as triggers for various effects on the visualization. We summarized some of our observations from the early usage of our system within a tiled display and outlined plans for the future formal evaluation of chirocentric UIs built on top of our platform.

6 Acknowledgements

The GIS data utilized in our figures and companion video is provided by Mapquest Open (`http://developer.mapquest.com/`). The Dubrovnik Sponza 3D model used in the companion video was created by Marko Dabrovic and improved by Frank Meinl of Crytek (`www.crytek.com`). This work has been supported by NSF grants CNS-0959979, IIP-1069147, IIS-0916235 and the New York Resiliency Institute for Storms and Emergencies.

References

[1] R. Balakrishnan and K. Hinckley. The role of kinesthetic reference frames in two-handed input performance. *Proceedings of the 12th annual ACM symposium on User interface software and technology*, pages 171–178, 1999.

[2] A. Banerjee, J. Burstyn, A. Girouard, and R. Vertegaal. Waveform: remote video blending for vjs using in-air multitouch gestures. *Human Factors in Computing Systems (Extended Abstracts)*, pages 1807–1812, 2011.

[3] T. Baudel and M. Beaudouin-Lafon. Charade: remote control of objects using free-hand gestures. *Communications of the ACM*, 36(7):28–35, 1993.

[4] N. Bogdan, T. Grossman, and G. Fitzmaurice. Hybridspace: Integrating 3d freehand input and stereo viewing into traditional desktop applications. *3D User Interfaces (3DUI), 2014 IEEE Symposium on*, pages 51–58, 2014.

[5] R. A. Bolt. *Put-that-there: Voice and gesture at the graphics interface*, volume 14. ACM, 1980.

[6] D. A. Bowman, R. P. McMahan, and E. D. Ragan. Questioning naturalism in 3d user interfaces. *Communications of the ACM*, 55(9):78–88, 2012.

[7] T. Grossman, D. Wigdor, and R. Balakrishnan. Multi-finger gestural interaction with 3d volumetric displays. *ACM symposium on User interface software and technology*, pages 61–70, 2004.

[8] G. Hackenberg, R. McCall, and W. Broll. Lightweight palm and finger tracking for real-time 3d gesture control. *IEEE Virtual Reality Conference*, pages 19–26, 2011.

[9] K. Hinckley, R. Pausch, and D. Proffitt. Attention and visual feedback: the bimanual frame of reference. *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 121–ff., 1997.

[10] J.-C. Lvesque, D. Laurendeau, and M. Mokhtari. *An Asymmetric Bimanual Gestural Interface for Immersive Virtual Environments*, pages 192–201. Springer, 2013.

[11] R. P. McMahan. *Exploring the Effects of Higher-Fidelity Display and Interaction for Virtual Reality Games*. Thesis, 2011.

[12] M. Nielsen, M. Strring, T. B. Moeslund, and E. Granum. *A procedure for developing intuitive and ergonomic gesture interfaces for HCI*, pages 409–420. Springer, 2004.

[13] C. Papadopoulos, K. Petkov, A. E. Kaufman, and K. Mueller. Reality Deck - Immersive Gigapixel Display. *IEEE Computer Graphics and Applications*, 35(1), 2014.

[14] C. Papadopoulos, D. Sugarman, and A. E. Kaufman. Nunav3d: A touch-less, body-driven interface for 3d navigation. *IEEE Virtual Reality 2012 Poster Session*, 0:67–68, 2012.

[15] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, and D. Samaras. Two-person interaction detection using body-pose features and multiple instance learning. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 28–35, 2012.

[16] J. Zigelbaum, A. Browning, D. Leithinger, O. Bau, and H. Ishii. g-stalt: a chirocentric, spatiotemporal, and telekinetic gestural interface. *Fourth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 261–264, 2010.