A Low-Cost Approach to Fish Tank Virtual Reality with Semi-Automatic Calibration Support

Niko Wißmann*
TH Köln, Computer Graphics Group

Martin Mišiak†
TH Köln, Computer Graphics Group
University Würzburg, HCI Group

Arnulph Fuhrmann‡
TH Köln, Computer Graphics Group

Marc Erich Latoschik§
University Würzburg, HCI Group

Abstract

We describe the components and implementation of a cost-effective fish tank virtual reality system. It is based on commodity hardware and provides accurate view tracking combined with high resolution stereoscopic rendering. The system is calibrated very quickly in a semi-automatic step using computer vision. By avoiding the resolution disadvantages of current VR headsets, our prototype is suitable for a wide range of perceptual VR studies.

Index Terms: Human-centered-computing—Human computer interaction (HCI)—Interaction devices—Displays and imagers; Human-centered-computing—Human computer interaction (HCI)—Interaction paradigms—Virtual reality

1 Introduction

Virtual Reality (VR) can be experienced through a variety of different systems. Advances in display and tracking technology increased the popularity of head-mounted VR systems (HMD´s) in both industry and academia. In contrast to large projection based systems such as CAVE´s or Powerwalls, HMD´s are significantly cheaper and easier to operate. However, they lack the high pixel-per-degree (ppd) resolutions of stationary projection systems.

Another cost effective alternative to a CAVE system is Fish Tank Virtual Reality (FTVR). The term and its characteristics were first described by Ware et al. [7]. FTVR is a monitor based VR system which displays a 3D scene with respect to the user's head position. Due to correct perspective and motion parallax cues, the user perceives the virtual content as seen through a window. In addition, the system can be further enhanced with stereoscopic rendering to improve the performance for various 3D tasks [1, 3, 7].

One important feature of FTVR is the ability to trade spatial resolution against fov by placing the user closer or further away from the display. This can completely eliminate the resolution factor present in current VR headsets, which makes FTVR a very interesting candidate for perceptual studies

For this reason we have developed an easy to calibrate and quickly to use FTVR system. It consists of a stereo capable gaming monitor, NVIDIA's 3D Vision shutter glasses and the HTC Vive tracking system. It enables accurate head motion parallax coupled with high resolution stereo rendering. The whole system is shown in Fig. 1.

Similar FTVR tracking setups can be found in [4], where the authors have created a projective AR setup with stereoscopic projectors and a custom made "tracking helmet" to evaluate visualization techniques and in [5] to evaluate the relationship between motion parallax and stereopsis.

*e-mail: niko.wissmann@gmail.com
†e-mail: martin.misiak@th-koeln.de
‡e-mail: arnulph.fuhrmann@th-koeln.de
§e-mail: marc.latoschik@uni-wuerzburg.de

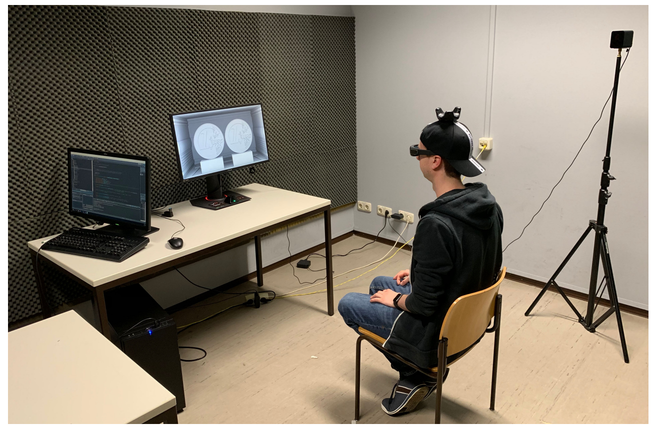

Figure 1: The full FTVR setup consisting of a custom made tracking cap with a fixed mounted HTC Vive tracker, one Vive base station, NVIDIA 3D Vision shutter glasses and a stereo monitor.

2 Technical Description

The tracking system is completely implemented in Java using the LWJGL[1] library. It provides 60 fps per eye, with a resolution of 2560×1440 pixels. The chosen monitor is a 27″ ASUS ROG Swift coupled with NVIDIA's 3D Vision Kit. The positional tracking component of the system relies on a singular Vive tracker mounted on top of a base cap, worn by the user and at least one Vive base station. Standard OpenVR API calls are used to query the necessary tracking data within the application. Using this setup a resolution of 95 ppd is achieved when viewing the monitor at a distance of 1.3m. This is significantly higher than the resolution of a consumer HMD (HTC Vive Pro - 14 ppd), but also above the human resolution limit of around 60 ppd [2].

2.1 Calibration

Prior to use, the system needs to be calibrated. The calibration is split into a one-time calibration and a quick per user calibration. The one-time calibration has to be performed when a new or modified Vive tracking setup is used, or when the position or size of the monitor has changed. To this end, the upper left, upper right and lower left corners of the monitor are located within the tracking coordinate system. Our implementation guides the user through this process by simply placing the Vive tracker on the corresponding locations of the monitor. The stored positions are then used to automatically determine the size and position of the screen, which is required for the calculation of the projection matrix.

[1]https://www.lwjgl.org/

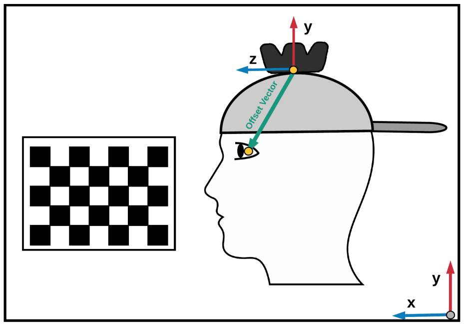

Figure 2: Illustration of the offset vector calibration process showing the relevant coordinate systems, as well as alignment and the two points to be set manually (orange).

The second calibration is done for each new user to adapt the system to the user's individual head shape. This is done by taking a side-facing photo of the user wearing the tracked base cap, in conjunction with a checkerboard reference pattern card held in the body's longitudinal plane. The photo is read into a small program which runs OpenCV checkerboard detection to determine the required offset vector. This is needed to shift the tracker origin to the users actual eye position along the y and z axis.

The program requires two manual inputs: one to mark the tracker origin in the photo and another to mark the position of the eyes as outlined in Fig. 2. Since the checkerboard has a known size and the tracker´s y and z axis coincide with the image plane, the world-space offset can be easily calculated using the supplied image space values. The same technique is also used to quickly determine the user's individual interpupillary distance (IPD) by taking a second front-facing photo with the reference card held against the forehead (see Fig. 3). The user is instructed to look into the camera lens, which is located at a similar distance as the later viewing distance to the monitor, to avoid deviating IPDs due to ocular convergence. The manual selection of both pupil centers results in the IPD. Both measured parameters are then inserted into a configuration file and the system is ready for use.

2.2 Tracking and Rendering

Within the tracking loop of the application, the projection matrices are calculated, which create an asymmetric view frustum for the current user position. The tracker position and rotation is queried via the OpenVR interface and is further shifted by the previous measured offset vectors to the actual eye position. This ensures correct view frustums. The subsequent rendering is done via OpenGL, where the 3D objects are drawn for each eye into the respective left and right back buffer. Switching the buffers invokes the synchronization call for the shutter glasses.

3 Conclusion

We have shown a cost efficient and easy to set up FTVR system with accurate head tracking and stereo rendering. The system is built with three simple components for tracking and stereo vision. Using a base cap with a mounted Vive tracker on top, enables head-coupled perspective rendering with motion parallax, while a high refresh rate gaming monitor with shutter glasses ensures high temporal and spatial resolution for stereoscopic rendering. A quick computer vision based calibration step adapts the system to each individual head shape and IPD. Especially the fast and straightforward calibration of the system makes it easy to use in a greater manner. The presented system has already been successfully evaluated in a perceptual study [6].

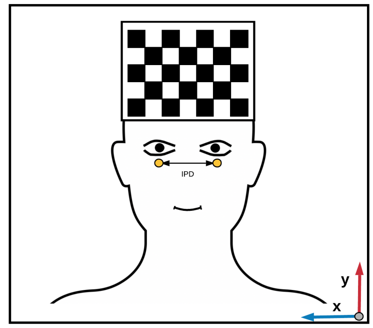

Figure 3: A front-faced picture is used to measure the user's individual IPD using the same reference card.

4 Limitations and Future Work

We have faced a few limitations during the development and study runs. First, the eye positions are only determined by extending the offset vector (Fig. 2) by half of the IPD along the trackers x axis, which does not account for individual eye offsets along the other axis. In addition, eye movements and the associated pupil offset cannot be tracked by our setup. Due to a technical limitation caused by IR synchronization of the shutter glasses, the system only works properly if the glasses are front-facing the monitor. Small head rotations to the left or right are possible, but more extensive rotations may cause shutter synchronization errors.

In the future we would like to use and extend the system for further studies, as well as find novel use cases. Furthermore, we seek to fully automate the calibration step, as well as improve upon the calibration accuracy.

References

[1] K. W. Arthur, K. S. Booth, and C. Ware. Evaluating 3d task performance for fish tank virtual worlds. *ACM Trans. Inf. Syst.*, 11(3):239–265, July 1993. doi: 10.1145/159161.155359

[2] E. Cuervo, K. Chintalapudi, and M. Kotaru. Creating the perfect illusion: What will it take to create life-like virtual reality headsets? In *Proceedings of the 19th International Workshop on Mobile Computing Systems & Applications*, pp. 7–12. ACM, 2018.

[3] D. Fafard, I. Stavness, M. Dechant, R. Mandryk, Q. Zhou, and S. Fels. Ftvr in vr: Evaluation of 3d perception with a simulated volumetric fish-tank virtual reality display. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290605.3300763

[4] F. Heinrich, K. Bornemann, K. Lawonn, and C. Hansen. Depth perception in projective augmented reality: An evaluation of advanced visualization techniques. In *25th ACM Symposium on Virtual Reality Software and Technology*, VRST '19, pp. 26:1–26:11. ACM, New York, NY, USA, 2019. doi: 10.1145/3359996.3364245

[5] S. Kongsilp and M. N. Dailey. Motion parallax from head movement enhances stereoscopic displays by improving presence and decreasing visual fatigue. *Displays*, 49:72 – 79, 2017. doi: 10.1016/j.displa.2017.07.001

[6] M. Misiak, N. Wissmann, A. Fuhrmann, and M. E. Latoschik. The impact of stereo rendering on the perception of normal mapped geometry in virtual reality. In *25th ACM Symposium on Virtual Reality Software and Technology*, VRST '19, pp. 92:1–92:2. ACM, New York, NY, USA, 2019. doi: 10.1145/3359996.3364811

[7] C. Ware, K. Arthur, and K. S. Booth. Fish tank virtual reality. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, p. 37–42. Association for Computing Machinery, New York, NY, USA, 1993. doi: 10.1145/169059.169066