Dual Sensor Filtering for Robust Tracking of Head-Mounted Displays

Nicholas T. Swafford[†], Bastiaan J. Boom[‡], Kartic Subr[‡], David Sinclair[‡], Darren Cosker[†], Kenny Mitchell[‡]

[‡]Disney Research
dred@disneyresearch.com

[†]University of Bath
[n.t.swafford, d.p.cosker]@bath.ac.uk

Abstract

We present a low-cost solution for yaw drift in head-mounted display systems that performs better than current commercial solutions and provides a wide capture area for pose tracking. Our method applies an extended Kalman filter to combine marker tracking data from an overhead camera with onboard head-mounted display accelerometer readings. To achieve low latency, we accelerate marker tracking with color blob localisation and perform this computation on the camera server, which only transmits essential pose data over WiFi for an unencumbered virtual reality system.

Keywords: Fast Feature Tracking, Head-Mounted Display

1 Introduction

Head-Mounted Displays (HMD) are gaining popularity as an interface for Virtual Reality (VR) applications. Systems like the Oculus Rift Development Kit 1 (DK1) provide built-in angular tracking of the user's head. However, the cummulative residual error in accelerometer based angular tracking can be significant, even with calibration and magnetometer-based corrections. Although recent consumer HMDs use front-facing cameras to attenuate the error and provide positional tracking, their positional tracking coverage is very restrictive and does not support multiple users.

The core contribution of this paper is a system that can perform positional and angular tracking of a HMD using real-time color blob marker detection and filters that data with onboard accelerometer data to provide more accurate yaw estimates. Our system uses readily available and inexpensive equipment to enhance existing HMDs.

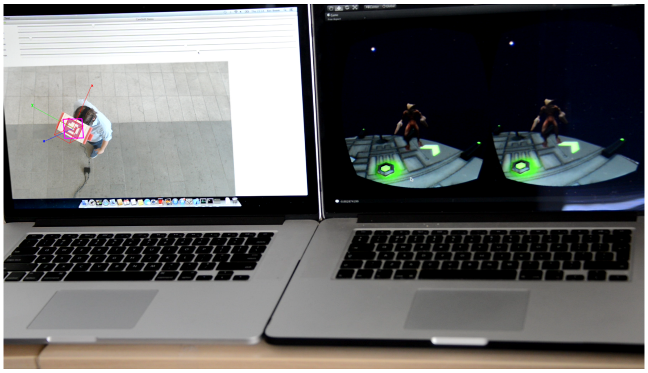

Figure 1: *The server (left) processes camera data and sends it to the client (right) where it is filtered with the accelerometer data.*

2 Related Work

Real-time marker tracking has been accomplished before [Zhang et al. 2002], but VR applications require low latency solutions due to users' heightened sensitivity to anomalies. Alternatively, color blob tracking is much faster but it is much less robust than marker tracking in imperfect lighting conditions [Pérez et al. 2002] .

The use of HMDs for entertainment was achieved more than a decade ago [Pausch et al. 1996], but these systems were prohibitively expensive for the average consumer until very recently. Popular interest in upcoming systems motivates research in low-cost improvements to maintain mass-market pricing.

3 Implementation, Experiments, and Results

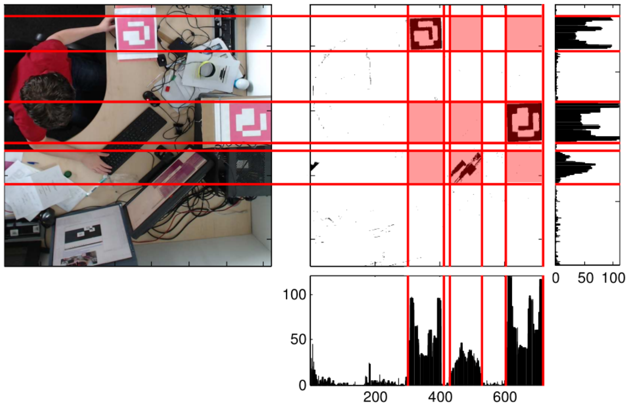

Figure 2: *Visualisation of our color blob tracker. Thresholding the captured image provides us with regions of interest.*

In order to compensate for marker and color tracking shortcomings, we decided to combine both methods to reduce the marker search space and increase the interval of acceptable color to accommodate lighting variation. Our biggest challenge was developing a fast method of obtaining image regions with the target color. To achieve this, we threshold our captured image into a binary image based on a color interval. This allows us to compute a histogram of matching pixels (black in the binary image) over all rows and all columns. We threshold again to remove noise and determine the regions which contain a significant number of black pixels and and run the marker detection on these sub-regions. A visualisation of this process can be seen in Figure 2. The color tracker used in OpenCV [Pérez et al. 2002] is too slow for our purposes, so we use Fast Feature Color Tracking [Spieldenner et al. 2014] instead. The camera is placed overhead to provide a wider tracking area, better coverage for typical head movement, and better estimate of yaw.

In order to not restrict the user's movement, the vision tracking is calculated on a server PC, which then transmits this pose data wirelessly to a client. A laptop (optionally placed in a backpack) connects to the HMD and processes a Kalman filter [Kalman 1960] combining the accelerometer stream with the visual angular data.

If we fail to detect the marker, a fail state is passed on to the client and we only use the accelerometer stream for most recent data. Importantly, the Kalman filter prevents judder during tracking failures because it ensures the use of historical data, smoothing out sudden transitions.

We use a Logitech C920 HD webcam for our overhead camera, the Oculus Rift DK1, and a colored AR marker [Garrido-Jurado et al. 2014] attached to the top of the HMD. Camera data was fed to a server with an Intel Core i7-3770 CPU @ 3.4GHz clockrate for trial data. We also used a MacBook Pro with an Intel Core i7-3720QM CPU and 2.6GHz clockrate as a server for further experiments.

For our experimentations the HMD was fastened securely to a tripod, calibrated to account for any possible magnetic interference from the tripod, and its rest position was marked. Trials were run with magnetometer correction and standard movement, without magnetometer correction, and with magnetometer correction and severe movement (expected in non-seated gameplay). Each individual trial lasted 26 minutes; the HMD would be rotated on each axis to simulate movement during gameplay for an 80 second interval, followed by a 20 second interval for reorientation to the rest position. This provided a periodic moment every 100 seconds where we could reliably observe cummulative drift. Orientation and positional data was polled at 60Hz (the camera's capture rate). In separate trials, we verified that our HMD was near the average yaw drift reported by the manufacturers[1].

The Kalman filter performed well in all three scenarios. This was least noticeable for the standard movement scenario with magnetometer correction, where adjustments during the periodic rest periods were on the order of a fraction of a degree. Without such correction, the HMD data had a yaw drift of approximately 25 degrees compared to the filtered data's 8 degrees. Our system performed exceptionally during the severe motion trials with magnetometer correction. The HMD orientation data had a 20 degree drift by the end of the trail, while the filtered data showed a drift of ˜5 degrees.

HMD VR developers propose maximum input-to-transmission latencies at 20ms, with suggested values between 7 and 15 ms[2]. On our trial PC, marker detection on its own took 9.09ms, whereas our color marker detection took 7.69ms; 4.01ms to locate color blobs plus 3.68ms for marker detection on the reduced search space. On laptop, our performance dropped to 12.21ms (5.54ms + 6.67ms), but a GPU implementation reduced this value to 8.21ms (1.54ms + 6.67ms). Network transmission delay was negligible at 0.54ms.

4 Discussion and Future Work

As an alternative tracking method, we could explore the use of cheap spherical tripoles for tracking [Sỳkora et al. 2008] with our fast-feature tracking [Spieldenner et al. 2014]. We expect that such a system would be less susceptible to failure on steeper orientation angles, operate faster than our combined color and marker tracking method, and be less cumbersome to attach to an HMD. We also believe there is unexplored potential of HMDs in low cost entertainment supporting multiple users as our system does. We would like to run extensive multiple-user trials in shared spaces.

5 Acknowledgements

We gratefully acknowledge funding from EPSRC and the European FI-PPP project, FI Content 2.

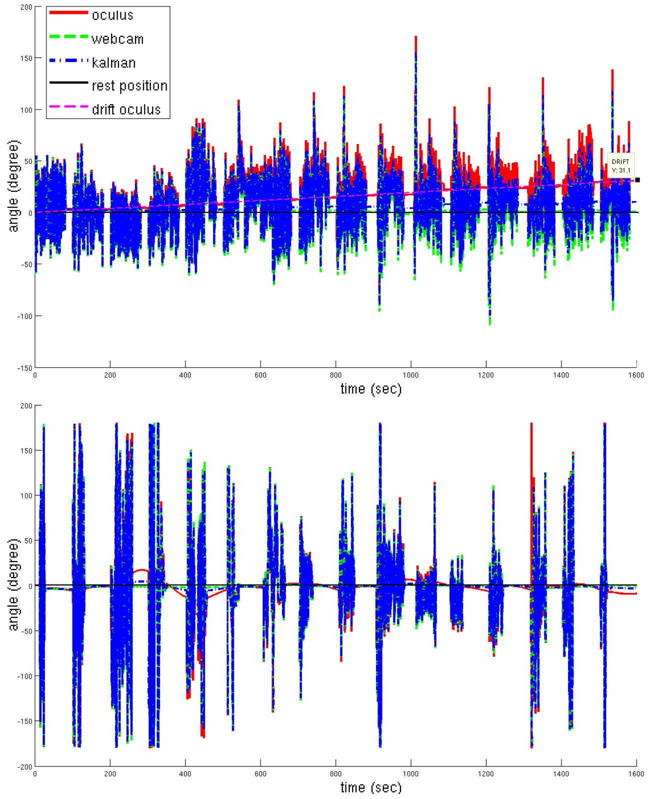

Figure 3: *Angular yaw estimates for each data stream over a regular movement without magnetometer correction trial (above) and a severe movement with magnetometer correction trial (below). There is a noticeable amount of drift in the yaw data throughout.*

References

GARRIDO-JURADO, S., MUOZ-SALINAS, R., MADRID-CUEVAS, F., AND MARN-JIMNEZ, M. 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition 47*, 6, 2280 – 2292.

KALMAN, R. 1960. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering 82*, 1, 35–45.

PAUSCH, R., SNODDY, J., TAYLOR, R., WATSON, S., AND HASELTINE, E. 1996. Disney's aladdin: first steps toward storytelling in virtual reality. In *Proceedings of SIGGRAPH'96*, ACM, 193–203.

PÉREZ, P., HUE, C., VERMAAK, J., AND GANGNET, M. 2002. Color-based probabilistic tracking. In *ECCV'02*, 661–675.

SPIELDENNER, T., SONS, K., LANCELLE, M., ZND, F., AND MITCHELL, K. 2014. An ecosystem for interactive mixed-reality applications on the web. *19th International Conference on 3D Web Technology - Tutorial*.

SỲKORA, D., SEDLÁČEK, D., AND RIEGE, K. 2008. Real-time color ball tracking for augmented reality. In *Proceedings of the 14th Eurographics conference on Virtual Environments*, Eurographics Association, 9–16.

ZHANG, X., FRONZ, S., AND NAVAB, N. 2002. Visual marker detection and decoding in ar systems: A comparative study. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, IEEE Computer Society, 97.

[1] http://www.oculusvr.com/blog/magnetometer/

[2] http://oculusrift-blog.com/john-carmacks-message-of-latency/682/