Fly It Like You Mean It
Mitigating Motion Sickness in First-Person-View Drones

Elias Wu
Stanford University
eliwu@stanford.edu

Abstract

Development of augmented and virtual reality systems for applications such as games or helping with daily life has become an active area of research, facilitated by the capabilities of GPUs as well as high density displays. However, motion sickness as a result of users wearing headsets still presents itself as a problem. A small subset of AR and VR, drone-related applications, is especially prone to producing motion sickness. Interestingly, however, efforts to mitigate these effects in these applications have been few and far between.

In our project, we implement a full system to mitigate a number of motion-sickness inducing factors, first implementing a low-latency video streaming system from the drone to the display system, then implementing a 3-axis gimbal in order to decouple the camera body completely from the drone body, and finally using a low-latency wireless link to mimic the headset's orientation on the 3 axis gimbal, coupling the user's head motions to the camera body.

We show in our limited studies that motion sickness is reduced slightly among the users tested when using this new system versus a standard non-gimballed camera of similar resolution.

1. Introduction

A common problem in the drone racing and drone piloting space is motion sickness - although some pilots can overcome this problem and fly well, a vast majority of people get motion sickness due to several factors. Some of the most important of these factors include the conflict between the user's physical body movement and the movement being displayed to the user [13] and the lag between head movement, sensor reading, recomputation and presentation to the display, a problem studied by Hettinger *et al*. [8] in 1992 as well as more recently by Mirk *et al*. [11]. In this particular use case, other reasons for motion sickness in virtual reality are diminished, such as the "uncanny valley" problem studied by Mori *et al*. [12].

2. Problem

In this project, we tackle several of the problems that induce motion sickness in first-person-view drone flying, namely, nonvestibular/vestibular disconnect, and time delay for both head movement and video transmission.

To this end, first-person-view systems for drones must accomplish several tasks differently from the "traditional" method of a fixed camera transmitting through commercial means such as WiFi; they must transmit a live stereo video feed from the drone to the viewer with minimal lag, as well as present this video feed to the viewer with minimal vestibular/nonvestibular disconnect by using some method to decouple the camera image from the drone body as well as couple the image to the user's head and body.

2.1. Literature review

Due to the relatively unstudied nature of first-person-view drones and motion sickness induced by flying these vehicles, there have only been blog posts and forum posts about the issue; no academic studies have been published regarding this narrow but growing problem. However, several papers attempt to implement ways to improve on mitigating motion sickness, proposing several different interesting methods. For example, a paper found that the more real an environment was, the less apparent motion sickness was (Jerome *et al*. [9]). Another utilized a nose in the middle of the VR display as a way of "anchoring" the user (Whittinghill *et al*. [15]).

Perhaps the most pertinent to our work, however, has been the aforementioned work by Palmisano *et al*. [13], verifying that the disconnect between physical body and head movement vs image movement caused motion sickness.

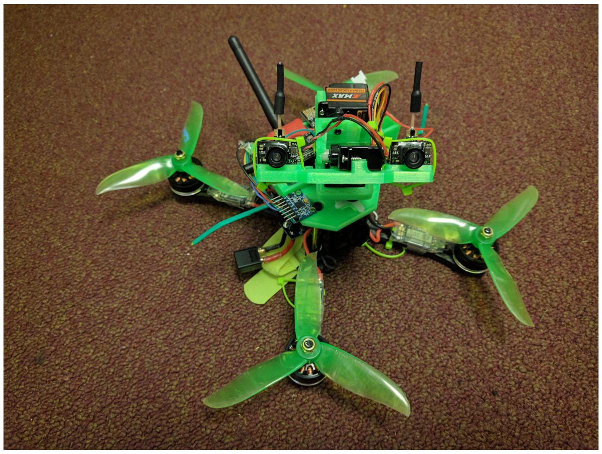

Figure 1. A view of the gimbal system mounted on top of a 210 mm size quadrotor racing-spec drone.

3. Methods

3.1. Implementation

Our embedded systems are implemented in Arduino-flavored C++, using two Teensy 3.2 boards [6]. The video receiving and displaying portion was implemented in javascript using the Three.JS framework [7].

3.2. Physical hardware

We first build a system to decouple the camera body from the drone body, couple the camera body to the VR headset body, and transmit live video between the headset and the drone.

3.2.1 3-axis Gimbal

The 3-axis gimbal is made by connecting 180-degree servos on 3 orthogonal axes, as shown in figure 2.

Additionally, the camera body holder is designed to hold the individual camera-transmitter modules to the approximate average interpupillary distance of 65mm. The arms were designed such that all arms were able to move a total of 180 degrees regardless of other arms' rotations, allowing the user to rotate in almost every direction. Parts were printed in PLA plastic for speed of prototyping and rigidity. Parts were designed in Solidworks 2017.

The base of the gimbal is designed to fit on top of a MultirotorMania Reaper 217 Stretch frame [5], an ideal frame for a gimbal of this size due to its light weight and stiff construction. The base also houses a Teensy 3.2 and an ESP 8266 WiFi chip [2] for communication to and from the headset. An additional two voltage regulators, LM317 [14] are used to power the ESP 8266 chip and the two camera modules, as both draw too much current for the Teensy 3.2's onboard regulator.

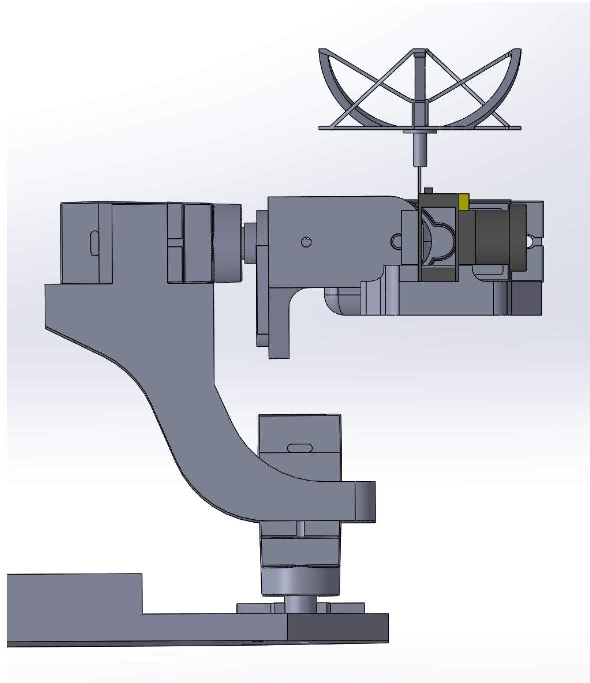

Figure 2. A view of the gimbal, showing the 3 orthogonal axes on which the camera body can move.

Figure 3. The camera stays level while the quad tilts to the right.

¡possibly another figure here for the circuitboard behind gimbal¿

3.2.2 Headset

The headset was built from existing plans from the EE267 Stanford course - a ViewMaster VR Headset Kit, 1080p display, as well as a VRDuino board. More information about these materials can be found at the course website [1].

Additionally, an ESP 8266, as well as an LM317 voltage regulator to power it, are added for communication to the

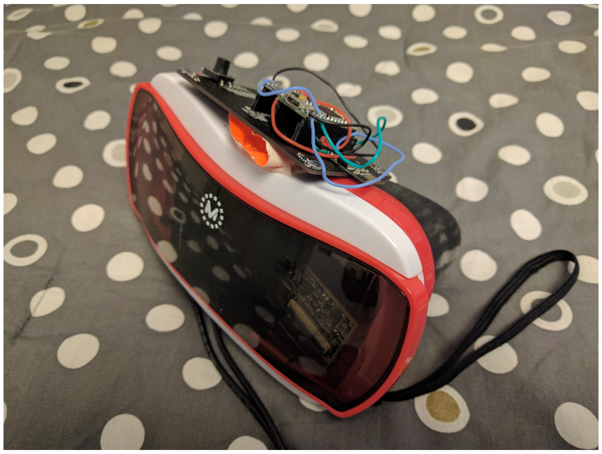

Figure 4. A view of the headset we built to display and track head movement.

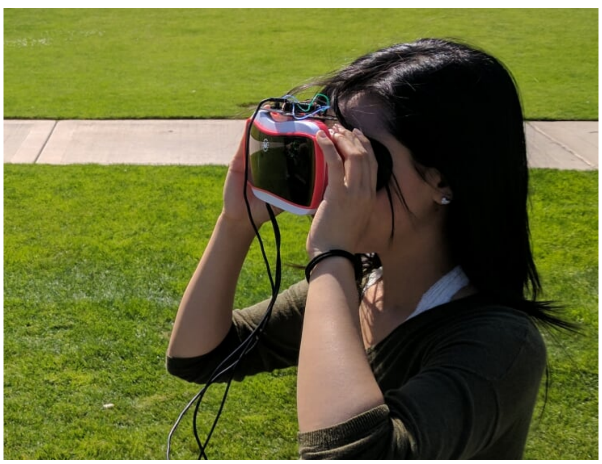

Figure 5. A subject using the headset.

gimbal for head tracking. An example of the headset as it was built is presented in figure 4. An example of the headset in use is presented in figure 5

Unfortunately, this headset is not ideal for the purposes of the project; since it does not have straps to mount to the users' head, users were unable to control the drone as well as look through the camera as originally intended.

Nevertheless, the the head mounted display was able to engage the users and allow them to accurately gauge whether they were experiencing motion sickness.

3.2.3 Video Transmission

Figure 6. A picture of the video receiver along with the camera module that includes a 200mW video transmitter.

To minimize the amount of time spent processing and displaying the video, we use analog video transmission, allowing for extremely low latencies of about 16 milliseconds [3], compared to digital transmission protocols such as wifibroadcast on linux, with 10x as much latency. video broadcasting at the cost of lower resolution and visual artifacts when signal is not strong. We use the 5.8ghz band as this is the easiest frequency to obtain hardware for, and is also not a restricted band.

For stereo vision, we use two different camera/transmitter modules paired to two different video receivers. These video receivers encode the analog video into digital and transmit the video data through USB to the base computer, to which the headset is attached. An example of the video receiver and the camera/transmitter module is presented in figure 6.

3.3. Software

3.3.1 Gimbal stabilization

To decouple the camera body from the drone body, an algorithm to keep the camera body in the same orientation regardless of the orientation of the drone body was needed.

Translation was not considered as this would have considerably increased development time and cost.

To get the quaternion rotation values of the drone body, an MPU-6050 6-axis inertial measurement unit from Invensense [4] was used. This sensor was ideal due to its very low cost and relatively accurate readings. Rather than implementing our own complementary or Kalman filter to get quaternion from the acceleration and rotation rate values, we use the MPU-6050's built-in Digital Motion Processor (DMP). The quaternion values were then converted to Euler angles for use with the gimbal motors according to equations (1), (2), and (3). (Note that it becomes clear that innately, 3-axis gimbals also physically have the problem of gimbal lock.)

$$\phi = atan2(2 * (q_0 * q_1 + q_2 * q_3), \\ 1 - 2 * (q_1^2 + q_2^2)) \quad (1)$$

$$\theta = asin(2 * (q_0 * q_2 - q_3 * q_1)) \quad (2)$$

$$\psi = atan2(2 * (q_0 * q_3 + q_1 * q_2), \\ 1 - 2 * (q_2^2 + q_3^2)) \quad (3)$$

Once these values were obtained, we set the axis motors accordingly; namely, roll and pitch axes on the gimbal switch as the drone body is increasingly yawed. (To see this, imagine the pitch axis of the gimbal looking forward - it tilts up as the drone body tilts down. However, when the gimbal is yawed $90 \deg$ to either left or right, the roll axis is now the new pitch axis). The equations for each of the axis angles are shown in equations (4), (5), and (6).

$$\textbf{roll} = 180 - (\phi) * cos(\psi) - (\theta) * sin(\psi) \quad (4)$$

$$\textbf{pitch} = 180 + (\phi) * sin(\psi) - (\theta) * cos(\psi) \quad (5)$$

$$\textbf{yaw} = \psi \quad (6)$$

3.3.2 Head Tracking

Now decoupled from the drone body, the gimbal can freely track the head motions of the user.

First, the Euler angles are read from the head tracker's MPU-6050 in the same manner as the gimbal's. The headset's ESP8266 is put into AP mode, to which the gimbal's ESP8266 connects to. A UDP connection on port 10002 is initiated, and the head tracker begins to stream Euler angle data to the gimbal. A small addendum is added to the stabilization algorithms to allow it to track the head angles. The updated values are shown in equations 7, 8, and 9.

$$\textbf{roll} = 180 - (\phi) * cos(\psi + yaw_{given}) \\ -(\theta) * sin(\psi + yaw_{given}) - roll_{given} \quad (7)$$

$$\textbf{pitch} = 180 + (\phi) * sin(\psi + yaw_{given}) \\ -(\theta) * cos(\psi + yaw_{given}) - pitch_{given} \quad (8)$$

$$\textbf{yaw} = \psi + yaw_{given} \quad (9)$$

Although the MPU-6050 does do an auto-calibrate for drift at the beginning of its boot cycle, there is eventual drift on the yaw axis - to combat this, "reset" buttons were placed on both the headset as well as the gimbal to reset the yaw axes.

Figure 7. The gimbal on the vehicle tracks the headset of the user.

3.3.3 Video display

Using javascript, particularly the navigator module, the two separate webcam feeds are read in a relatively straightforward manner - after a call to getUserMedia(), the last two video devices are taken and set to two video elements.

These two elements are read by another javascript file, and are turned into Three.JS textures. The textures are then applied to two basic mesh objects, and placed in a 3d scene with two viewports - one for the left and one for the right eye. The mesh objects are placed at $\pm IPD/2$ in order to make sure the images are centered on each pupil. The camera for each eye is set at a distance that allows most of the image to be inside the viewport while filling most of the blank space as well. This maximizes the amount of image the viewer can see, but minimizes the blank space in the screen. An example of the output images is shown in figure 8.

Figure 8. An example of what the user sees inside the headset. Notice the stereo images slightly offset.

4. Experiments and results

In testing the gimbal, it appeared to be robust - even flipping the drone upside down and performing high speed maneuvers did not make the gimbal lose its orientation lock. The only axis which drifted was yaw, which was to be expected as compass readings were not included.

The software angle limits work as expected, not allowing the gimbal to go past the hard stop points, and preventing damage.

Head tracking was also robust - although the ESP8266 chips are not very powerful and are sometimes disturbed by the 2.4ghz control signal to the drone, recovery is very fast and latency for head tracking is not perceivable. If the head tracker goes past the software limits, the gimbal locks to the maximum angle without damaging itself.

Compared to the fixed camera on the drone, the gimballed stereo camera was much more stable and appeared to stay stable regardless of the drone's orientation.

We also tested the gimbal and head tracking system on 6 test subjects - 3 using the fixed camera, and 3 using the gimballed, head tracking stereo camera. Overwhelmingly the results point to the gimballed camera offering a much more stable experience, resulting in less motion sickness. We used the Simulation Sickness Questionnaire [10], a standard simulation sickness questionnaire widely used for recording motion sickness in VR, to record the users' reactions to the experiences. A table of results on subjects who used the gimballed system are presented in table 4.

Symptom	Subj. 1	Subj. 2	Subj. 3
General discomfort	Slight	Slight	None
Fatigue	Slight	None	None
Headache	None	None	None
Eye strain	Slight	Slight	Moderate
Difficulty focusing	None	None	None
Salivation increasing	None	None	None
Sweating	None	None	None
Nausea	Slight	None	None
Difficulty concentrating	None	None	None
Fullness of the Head	Slight	None	None
Blurred vision	None	Slight	None
Dizziness with eyes open	None	Slight	None
Dizziness with eyes closed	None	None	Slight
Vertigo	None	None	None
Stomach awareness	None	None	None
Burping	None	None	None

Table 1. Questionnaire answers for subjects who used the gimballed, head-tracking system. The scale ranges from None, Slight, Moderate, to Severe.

Likewise, a table of results for subjects who used the fixed camera are presented in figure 4.

5. Discussion

Our results, although on a small number of subjects due to time constraints, indicate that our gimballed system does indeed decrease motion sickness, confirming our hypotheses and prior research. Although subjects responded that they still felt slight discomfort while using the stabilized and head-tracked system, responses from the subjects who used the fixed camera tend towards moderate discomfort more than the subjects who used the gimballed system.

5.1. Future work

Future work which would likely immediately lead to improved results fine tuning the angles as well as using servos with higher accuracy, speed, and fidelity. Currently, our servos are analog, metal gear, low-fidelity, and low-cost options in order to prove that the concept could in fact produce good results. Many servos on the market use coreless, brushless, or other exotic drive systems that result in much higher fidelity, smoothness, and accuracy. Additional qualifiers to explore would be digital servos which can respond much faster, as well as ultra-high-speed, a designation for servos with $0°$ to $60°$ times in under 0.05 seconds.

Symptom	Subj. 4	Subj. 5	Subj. 6
General discomfort	Moderate	Moderate	Moderate
Fatigue	Slight	Moderate	None
Headache	None	Slight	None
Eye strain	Moderate	Slight	Moderate
Difficulty focusing	Moderate	Moderate	Slight
Salivation increasing	None	None	None
Sweating	None	None	None
Nausea	Slight	Slight	Slight
Difficulty concentrating	Slight	Slight	Slight
Fullness of the Head	Slight	Slight	None
Blurred vision	None	Slight	None
Dizziness with eyes open	None	Slight	None
Dizziness with eyes closed	None	None	Moderate
Vertigo	Slight	None	Slight
Stomach awareness	None	None	None
Burping	None	None	None

Table 2. Questionnaire answers for subjects who used the gimballed, head-tracking system. The scale ranges from None, Slight, Moderate, to Severe.

Another avenue of improvement is in the video lag - although our use of analog video for one portion of the video transmission decreases latency greatly, our method of using USB-based analog video receivers to pipe video information to a web browser is the largest bottleneck for video transmission. Processing the video data from the analog receivers is relatively slow and results in a perceivable, undesirable lag in the system.

The cameras used are low-cost, all-in-one packages with little care taken in aligning lenses correctly. As a result, tuning the camera angles and making sure the stereo image resolves when users see the image is difficult. The use of higher quality cameras, both higher in resolution and build quality, would solve this problem, and would provide a better experience as images are clearer.

The headtracking system currently uses low-cost ESP8266 as data transmission - not only is this not fully reliable, it is also much lower range than the other wireless radios on the drone itself, making it the bottleneck of the experience. To this end, using a higher-throughput, more reliable networking solution, such as FrSky, Ubiquiti, or others would allow the experience to not only be smoother, but also quicker on startup.

Lastly, the gimbal is not able to rotate past 180 degrees, limiting the user to a small subset of orientations they could be in. Using slip rings and clever structural design, this could be either increased or made continuous—360°—making the experience more immersive.

Acknowledgments

The author would like to thank the teaching staff of EE267 for offering a captivating and organized course. We especially enjoyed the demo sessions and appreciate the feedback we received.

References

[1] Ee 267: Virtual reality, 2017.

[2] Esp8266, 2017.

[3] Fpv video latency comparison: Analog vs. connex prosight vs. fpv.blue vs. wifibroadcast, 2017.

[4] Mpu-6050 six-axis (gyro + accelerometer) mems motion-tracking devices, 2017.

[5] Mrm reaper 217 frame, 2017.

[6] Teensy 3.2 new features, 2017.

[7] three.js, 2017.

[8] L. J. Hettinger and G. E. Riccio. Visually induced motion sickness in virtual environments. *Presence: Teleoper. Virtual Environ.*, 1(3):306–310, July 1992.

[9] C. Jerome, R. Darnell, B. Oakley, and A. Pepe. The effects of presence and time of exposure on simulator sickness. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 49, pages 2258–2262. SAGE Publications Sage CA: Los Angeles, CA, 2005.

[10] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology*, 3(3):203–220, 1993.

[11] D. Mirk and H. Hlavacs. Virtual tourism with drones: Experiments and lag compensation. In *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, pages 45–50. ACM, 2015.

[12] M. Mori, K. F. MacDorman, and N. Kageki. The uncanny valley [from the field]. *IEEE Robotics Automation Magazine*, 19(2):98–100, June 2012.

[13] S. Palmisano, R. Mursic, and J. Kim. Vection and cybersickness generated by head-and-display motion in the oculus rift. *Displays*, 46:1–8, 2017.

[14] Texas Instruments Inc. *3-Terminal Adjustable Regulator*, 2016.

[15] D. Whittinghill, B. Ziegler, J. Moore, and T. Case. 'virtual nose' may reduce simulator sickness in video games, 2015.