Head Tracking for the Oculus Rift

Steven M. LaValle[1]   Anna Yershova[1]   Max Katsev[1]   Michael Antonov

Oculus VR, Inc.
19800 MacArthur Blvd
Irvine, CA 92612 USA

Abstract— **We present methods for efficiently maintaining human head orientation using low-cost MEMS sensors. We particularly address gyroscope integration and compensation of dead reckoning errors using gravity and magnetic fields. Although these problems have been well-studied, our performance criteria are particularly tuned to optimize user experience while tracking head movement in the Oculus Rift Development Kit, which is the most widely used virtual reality headset to date. We also present novel predictive tracking methods that dramatically reduce effective latency (time lag), which further improves the user experience. Experimental results are shown, along with ongoing research on positional tracking.**

I. INTRODUCTION

In June 2012, Palmer Luckey's prototype headset generated widespread enthusiasm and hopes for transformative virtual reality (VR) as John Carmack used it to develop and show a compelling Doom demo at the Electronic Entertainment Expo (E3). This convinced industry leaders that a low-cost, high-fidelity VR experience could be achieved by leveraging MEMS sensing and video display technology from the smart phone industry. One important aspect of the design is a wide viewing angle, which greatly increases the sense of immersion over most prior headsets. Momentum has continued to build since that time, with broad media coverage progress on the potential for consumer VR, the dispersal of 50,000 Oculus Rift Development Kits, and the active cooperation of developers and researchers around the world. Although originally targeted at the game industry, it has been finding application more broadly in art, entertainment, medicine, architecture, military, and robotics. Of particularly high potential in robotics is telepresence, where a portable VR interface to a robot can be maintained to allow VR-based teleconferencing, travel, and even virtual attendance of a conference such as ICRA. Furthermore, roboticists have a long history of borrowing new sensors and devices from other industries and finding exciting new uses; examples include the SICK laser, Kinect, and Wii Remote.
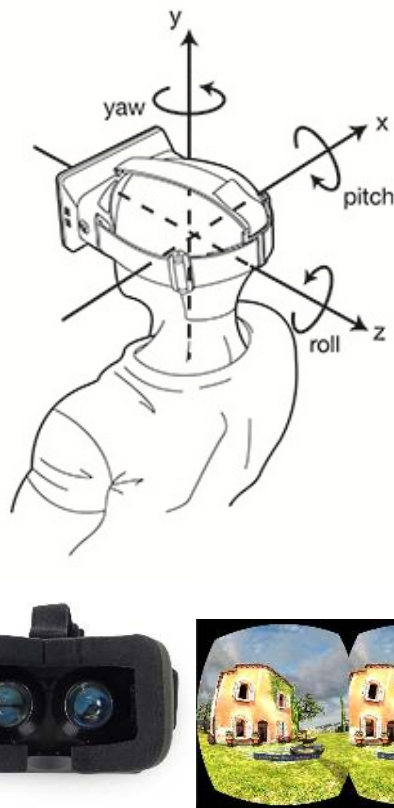

Fig. 1.   The Oculus Rift Development Kit tracks head movement to present the correct virtual-world image to the eyes.

At the same time, a core challenge to making the Rift work is quite familiar to roboticists: Localization [1]. However, the VR version of it has special requirements [2]. Virtual reality works by aligning coordinate frames between physical and virtual worlds. Maintaining camera poses for rendering in the virtual world requires sensing and filtering of the head orientation. Including head position further aligns the physical and virtual worlds to improve the experience. Because VR is fooling the brain, the head must be tracked in a way that minimizes per-

---

[1]Also affiliated with the Department of Computer Science, University of Illinois, Urbana, IL 61801 USA. Corresponding author: Steve LaValle, Principal Scientist, Oculus VR, steve.lavalle@oculusvr.com
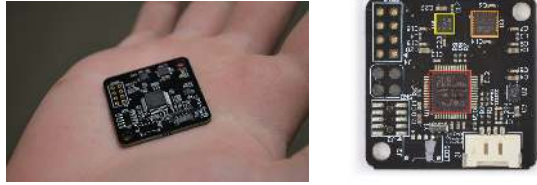
Fig. 2. The custom sensor board inside of the Rift.

ceptual artifacts. In addition to accuracy, it is particularly important to provide stable motions. A further challenge is to reduce *latency*, which is the time between moving the head and producing the correct change on the user's retinas. Up to now, latency has been the key contributor to VR simulator sickness. In our case, these particular challenges had to be overcome with a simple MEMS-based inertial measurement unit (IMU).

This paper presents our head tracking challenges, solutions, experimental data, and highlights some remaining research issues.

II. THE HARDWARE

All sensing is performed by a single circuit board, shown in Figure 2. The main components are:

- STMicroelectronics 32F103C8 ARM Cortex-M3 microcontroller
- Invensense MPU-6000 (gyroscope + accelerometer)
- Honeywell HMC5983 magnetometer.

The microcontroller interfaces between the sensor chips and the PC over USB. Each of the gyroscope (gyro), accelerometer, and magnetometer provide three-axis measurements. Sensor observations are reported at a rate of 1000Hz.[2] We therefore discretize time t into intervals of length $\Delta t = 0.001$s. The kth *stage* corresponds to time $k\Delta t$, at which point the following 3D measurements arrive:

1) Angular velocity: $\tilde{\omega}[k]$ rad/sec
2) Linear acceleration: $\tilde{a}[k]$ m/s^2
3) Magnetic field strength: $\tilde{m}[k]$ Gauss.

Also available are temperature and timing from the microprocessor clock.

III. GYRO INTEGRATION AND DRIFT

From now until Section VIII, we consider the problem of estimating the current head *orientation*, as opposed to position. Each orientation can be expressed as a 3 by 3 rotation matrix. Let $SO(3)$ denote the set of all rotation matrices, which is the space on which our filter operates.

By Euler's Rotation Theorem, any 3D orientation can be produced by a single rotation about one particular axis through the origin. This *axis-angle* representation maps naturally to the space of unit quaternions as $q(v, \theta) =$

$$(\cos(\theta/2), v_x \sin(\theta/2), v_y \sin(\theta/2), v_z \sin(\theta/2)), \quad (1)$$

in which $q(v, \theta)$ denotes a unit-length quaternion that corresponds to a rotation of θ radians about a unit-length axis vector $v = (v_x, v_y, v_z)$. (Note that $q(v, \theta)$ and $-q(v, \theta)$ represent the same rotation, which is carefully handled.)

The quaternion representation allows singularity-free manipulation of rotations with few parameters while correctly preserving algebraic operations. It is also crucial to making a numerically stable dead-reckoning method from gyro measurements. Let $\omega = (\omega_x, \omega_y, \omega_z)$ be the current angular velocity in radians/sec. Let the magnitude of ω be $\ell = \|\omega\|$. Following from the definition of angular velocity:

- The current axis of rotation (unit length) is $\frac{1}{\ell}\omega$.
- The length ℓ is the rate of rotation about that axis.

From the classical relationship of a Lie algebra to its associated Lie group [3], the axis-angle representation of velocity can be integrated to maintain an axis-angle representation of orientation, which is conveniently expressed as a quaternion. The details follow.

Let $q[k]$ be a quaternion that extrinsically represents the Rift (or sensor) orientation at stage k with respect to a fixed, world frame. Let $\tilde{\omega}[k]$ be the gyro reading at stage k. Let $\hat{q}[k]$ represent the *estimated* orientation. Suppose $\hat{q}[0]$ equals the initial, identity quaternion. Let $\ell = \|\tilde{\omega}[k]\|$ and $v = \frac{1}{\ell}\tilde{\omega}[k]$. Because ℓ represents the rate of rotation (radians/sec), we obtain a simple dead reckoning filter by setting $\theta = \ell\Delta t$ and updating with[3]

$$\hat{q}[k+1] = \hat{q}[k] * q(v, \theta), \quad (2)$$

in which $*$ represents standard quaternion multiplication. This is equivalent to simple Euler integration, but extended to $SO(3)$. (This easily extends to trapezoidal integration, Simpson's rule, and more sophisticated numerical integration formulas, but there was no significant performance gain in our application.)

A common gyro error model [4] is:

$$\tilde{\omega} = \omega + S\omega + M\omega + b + n, \quad (3)$$

in which S is the scale-factor error, M is the cross-axis coupling error, b is the zero-rate offset (bias), and n is the zero-mean random noise. Note that this model ignores

---

[2]The magnetometer readings, however, only change at a rate of 220Hz.

[3]Note that $q(v, \theta)$ multiplies from the right because $\tilde{\omega}[k]$ is expressed in the local sensor frame. If converted into the global frame, it would multiply from the left.
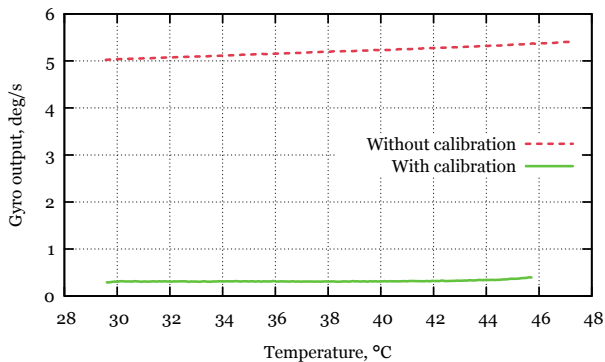
Fig. 3. Stationary gyro output with and without calibration.

less significant error sources, including effects of linear acceleration, board stress, nonlinearity, and quantization.

The scale-factor and cross-axis terms can be combined to produce a simple linear model:

$$\tilde{\omega} = K\omega + b. \tag{4}$$

Values for K and b can be determined by a factory calibration; however, due to imperfections in the calibration procedure, as well as temperature dependence, the corresponding error terms cannot be completely eliminated. Figure 3 demonstrates zero-rate offset with and without calibration for different values of temperature. Note that even after calibration, most of the error is systematic, as opposed to random noise.

Over time, we expect dead-reckoning error to accumulate, which will be called *drift error*:

$$e[k] = \hat{q}[k] - q[k]. \tag{5}$$

Note that this can also account for the difference between initial frames: $\hat{q}[0] - q[0]$.

IV. DRIFT CORRECTION WITH A CONSTANT FIELD

Additional sensing beyond the gyro is needed to drive down the drift error. The methods of Sections V and VI, are based on aligning the orientation with respect to a fixed, constant vector field that permeates the physical space of the sensor. They each have particular issues, but the common principle is first covered here.

Suppose that the Rift is placed into physical workspace, modeled as $\mathbb{R}^3$. In that space, a constant vector field $f : \mathbb{R}^3 \rightarrow \{w\}$ is defined in which $w = (w_x, w_y, w_z)$. At every position $p \in \mathbb{R}^3$, the field yields the same value, $f(p) = w$. The magnitude of this field will have no consequence for drift correction. Therefore, assume without loss of generality that $\|w\| = 1$.

Imagine that a magical sensor is placed in the Rift that perfectly measures the field. If the Rift is at the identity

orientation, then $\tilde{w} = w$ is observed. However, if the Rift is at an arbitrary orientation q, then the sensor observation is transformed using the inverse rotation q^{-1} (this is the conjugate of q). The resulting observation is $\tilde{w} = q * w * q^{-1}$.[4] Using rotation matrices, this would be $\tilde{w} = R^t w$, for rotation matrix R.

The difference between $\tilde{w}$ and w clearly contains powerful information about orientation of the Rift, but what are the limitations? Consider the following sensor mapping:

$$h : SO(3) \rightarrow \mathbb{R}^3, \tag{6}$$

which yields $\tilde{w} = h(q)$ as the observed vector, based on the orientation q. The trouble is the preimage of the mapping [5]:[5]

$$h^{-1}(\tilde{w}) = \{q \in SO(3) \mid h(q) = \tilde{w}\}. \tag{7}$$

In other words: Consider the set of all orientations that produce the same sensor observation. Each preimage is a generally a one-dimensional set that is generated for each $\tilde{w}$ by applying a rotation about an axis parallel to w. This should not be surprising because the set of all possible directions for w is two-dimensional whereas $SO(3)$ is three-dimensional.

In practice, we do not have the magical field sensor. The actual sensors have limitations due to calibration errors, precision, noise. Furthermore, they unfortunately measure multiple, superimposed fields, making it difficult to extract the correct information. The next two sections handle these problems for the gravity-based and magnetic field-based drift correction methods.

V. TILT CORRECTION

This section presents our approach to correcting *tilt error*, which is the component of all drift error except rotation about the vertical axis in the physical world, and it results in the perceived horizontal plane not being level. Correction is accomplished by using gravity as a constant vector field, in the way described in Section IV. The preimages (7) in this case correspond to rotations around axis parallel to the gravity vector.

In an ideal world, we would love to have a *perfect gravitational field sensor*. It would always provide a vector of magnitude 9.81m/s^2, with the direction indicating the tilt. In reality, gravitational field of Earth deviates slightly in both the magnitude and direction of gravity across its surface. These deviations are minor, though, and we currently ignore them. A much worse problem is that an accelerometer measures the vector sum of *all* of the

---

[4]For quaternion-vector multiplication, we assume the vector is converted to a quaternion as $(0, w_x, w_y, w_z)$.

[5]To be more precise, we should write $R \in SO(3)$ in which R is the rotation matrix to which q corresponds.
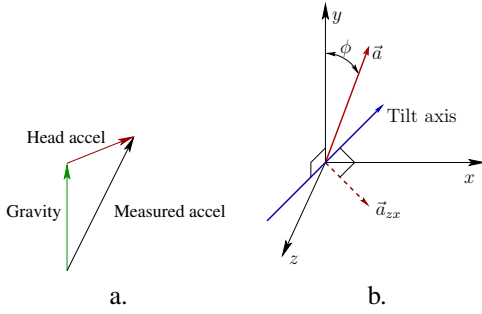
Fig. 4. (a) Accelerometers necessarily measure the vector sum of acceleration due to gravity and linear acceleration of the sensor with respect to Earth. (b) To determine tilt error angle ϕ, the tilt axis is calculated, which lies in the horizontal, XZ plane.

Fig. 5. Tilt correction performance. Ground truth data was collected using OptiTrack motion capture system.

contributing accelerations (see Figure 4(a)): $a = a_g + a_l$, in which a_g is the acceleration due to gravity and a_l is linear acceleration of the head relative to the Earth.

Suppose that there is no linear acceleration. In this case, $\tilde{a}$ is a direct estimate of the gravity vector; however, this is measured in the local coordinate frame of the sensor. The transformation

$$\hat{a} = q^{-1} * \tilde{a} * q \tag{8}$$

brings it back into the global frame. Note that this applies the inverse of the head orientation. Every tilt error can be described as a rotation about an axis that lies in the horizontal XZ plane; see Figure 4(b). To calculate the axis, project $\hat{a}$ into the XZ plane to obtain $(\hat{a}_x, 0, \hat{a}_z)$. The tilt axis is orthogonal: $t = (\hat{a}_z, 0, -\hat{a}_x)$. The tilt error ϕ is the angle between $\hat{a}$ and the vector $(0, 1, 0)$.

In the presence of movement, we cannot trust accelerometer data in the short term. However, averaged over a long period of time, accelerometer output (in the global frame) produces a good estimate for the direction of gravity. Indeed, for n samples,

$$\left| \frac{1}{n} \sum_{k=1}^{n} \hat{a}[k] - a_g \right| = \left| \frac{1}{n} \sum_{k=1}^{n} (a_g + a_l[k]) - a_g \right|$$
$$= \frac{1}{n} \left| \sum_{k=1}^{n} a_l[k] \right| = \frac{1}{n} |v[n] - v[1]| = O(1/n), \tag{9}$$

in which $v[k]$ is velocity of the headset at stage k and is bounded due to the physiological constraints of human body. To further improve performance, the data are preprocessed by removing samples that differ significantly from the previous measurements, since sudden changes in the combined acceleration are more likely to happen due to movement rather than drift.

The gyro is reliable in the short term, but suffers from drift in the long term. To combine short-term accuracy gyro data with long-term accuracy of accelerometer data
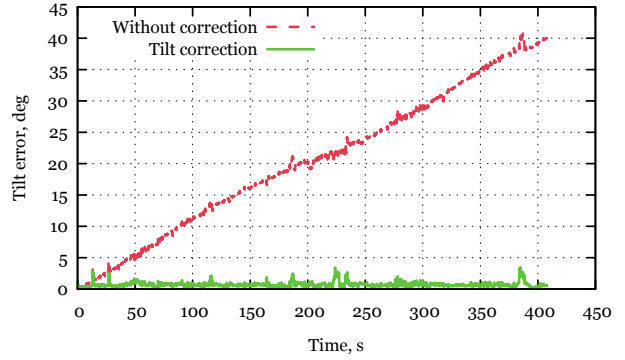
we apply a complementary filter [6], [7], [8]. This particular choice is motivated by simplicity of implementation and adjustment based on perceptual experiments. Let $\hat{q}[k]$ be the estimated orientation obtained by gyro integration (2); the output of the complementary filter with gain $\alpha \ll 1$ is

$$\hat{q}'[k] = q(t, -\alpha\phi) * \hat{q}[k], \tag{10}$$

in which t is the tilt axis. The parameter α should be large enough to correct all drift, but small enough so that the corrections are imperceptible to the user. Figure 5 shows performance during typical use.

VI. YAW CORRECTION

This section addresses *yaw error*, which corresponds to rotation about the vertical axis (parallel to the gravity vector). To accomplish this task, we rely on measurement of the magnetic field using the magnetometer mentioned in Section II.

It is temping to think of a magnetometer as a compass, which measures a vector that always points North. The situation, however, is considerably more complicated. First, note that the observation $\tilde{m} = (\tilde{m}_x, \tilde{m}_y, \tilde{m}_z)$ is three-dimensional. Imagine using the sensor to measure the Earth's magnetic field. Unfortunately, this field is not constant, nor do the vectors point North. The magnitude of the field varies over the surface of the Earth from 0.25 to 0.65 Gauss. The difference between North and the field direction, projected into the horizontal plane is called a *declination angle*. This could be as large as 25 degrees in inhabited areas. To further complicate matters, the field vector points up or down at an *inclination angle*, which varies up to 90 degrees. These cause two problems: 1) The direction of true North is not measurable without knowing the position on the Earth (no GPS system is in use), and 2) if the inclination angle is close to 90 degrees, then the magnetic field is useless for yaw correction because it is
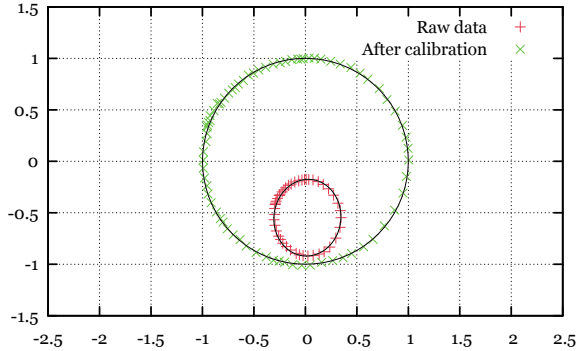
Fig. 6. A 2D slice of magnetometer output before and after calibration.

almost parallel to gravity. Fortunately, this only happens in the most sparsely populated areas, such as northern Siberia and the Antarctic coast.

However, it is even more complicated than this. Recall from Section V that the accelerometer only measures the vector sum of two sources. The magnetometer also measures the sum of several sources. Circuit boards usually contain ferrous materials that interfere with magnetic fields. There may be both *hard iron bias* and *soft iron bias* [9]. In the case of hard iron bias, nearby materials produce their own field, which is observed as a constant offset in the frame of the sensor. In the case of soft iron bias, the materials distort existing fields that pass through them. Most indoor environments also contain enough materials that create magnetic fields. Therefore, a magnetometer measures the vector sum of three kinds of fields:

1) A field that exists in the *local* coordinate frame of the sensor.
2) An indoor field that exists in the *global*, fixed world frame.
3) The Earth's magnetic field, which is also in the global frame.

All of these could be distorted by soft iron bias. Finally, additional anomalies may exist due to the proximity of electrical circuits or moving ferrous materials (someone walks by with a magnet).

Figure 6 shows a plot of magnetometer data obtained by rotating the sensor in one location around a single axis. An ellipse is roughly obtained, where eccentricity is caused by soft iron bias. The center of the ellipse is offset due to the local field. The size of the ellipse is due to the global field, which includes the Earth's field. One problem for correction using a magnetometer is that the vector sum of the Earth's field and the indoor field may mostly cancel, leaving only a small magnitude.

Our approach is based on the following principles: 1) True North is irrelevant because we are not trying to navi-

gate across the Earth; 2) the vector sum of the indoor field and the Earth's field provides the useful measurement; 3) an in-use calibration procedure can eliminate most of the offset due to the local field, and it can furthermore help with soft iron bias; 4) never assume that the calibrated values are completely accurate; 5) the particular field magnitude is irrelevant, except for the requirement of a minimum value above the noise level.

Based on these considerations, we developed a method that first *calibrates* the magnetometer and then assigns *reference points* to detect yaw drift. For calibration, we could ask the user to spin the Rift in many directions to obtain enough data to perform a least-squares fit of an ellipsoid to the raw data. The raw values are then corrected using an affine linear transformation based on the ellipsoid parameters. An alternative is to grab only four well-separated points and calculate the unique sphere through them. In this case, the raw samples are corrected by applying the estimated offset due to the local field. The four points can even be obtained automatically during live use. This often works well, assuming the user looks in enough directions and the eccentricity due to soft iron bias is small.

Suppose that $\tilde{m}$ is now a measurement that has been already corrected by calibration. Let $\tilde{m}_{ref}$ be a value observed at an early time (before much drift accumulates). Let $\hat{q}_{ref}$ be the corresponding orientation estimate when $\tilde{m}_{ref}$ was observed. Now suppose that a value $\tilde{m}$ is later observed and has associated value $\hat{q}$. The following transformations bring both readings into the estimated global frame:

$$\tilde{m}' = \hat{q}^{-1} * \tilde{m} * \hat{q} \text{ and } \tilde{m}'_{ref} = \hat{q}_{ref}^{-1} * \tilde{m}_{ref} * \hat{q}_{ref}. \quad (11)$$

If the magnetometer were perfectly calibrated and $\tilde{m}' \approx \tilde{m}'_{ref}$, then there would be no significant drift. We project $\tilde{m}'$ and $\tilde{m}'_{ref}$ into the horizontal XZ plane and calculate their angular difference. More explicitly, $\theta = \text{atan2}(\tilde{m}'_x, \tilde{m}'_z)$ is compared to $\theta_r = \text{atan2}(\tilde{m}'_{ref,x}, \tilde{m}'_{ref,z})$. Once the error is detected, the methods are the same as in Section V, except that the correction is a rotation about the Y axis only. The complementary filter for yaw correction is

$$\hat{q}'[k] = q((0,1,0), -\alpha_2(\theta - \theta_r)) * \hat{q}[k], \quad (12)$$

in which α_2 is a small gain constant, similar to the case of tilt correction.

To account for poor calibration (for example, ignoring eccentricity), we require that $\hat{q}$ be close to $\hat{q}_{ref}$. Otherwise, the error due to calibration would dominate the yaw drift error. This implies that yaw drift can only be detected while "close" to $\hat{q}_{ref}$. This limitation is resolved by introducing multiple reference points, scattered around $SO(3)$.
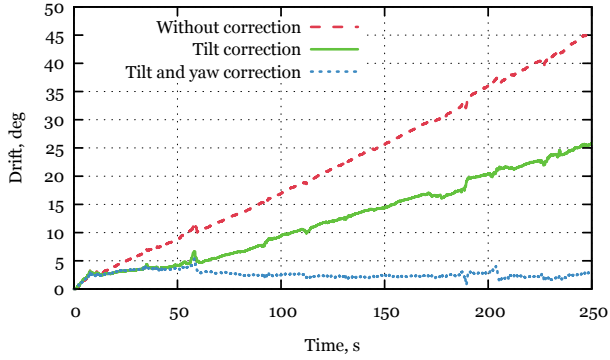
Fig. 7. Effect of correction methods on the overall drift.

VII. PREDICTIVE TRACKING

For our system, it is not enough for the filtering system to determine $q[k]$ at precisely time $t = k\Delta t$. It must reliably predict the future orientation at the time the user observes the rendering output. The *latency interval* is the time from head movement to the appearance of the corresponding image on the retinas. The pipeline includes time for sensing, data arrival over USB, sensor fusion, game simulation, rendering, and video output. Other factors, such as LCD pixel switching times, game complexity, and unintentional frame buffering may lengthen it further.

Latency is widely considered to cause simulator sickness in VR and augmented reality (AR) systems. A commonly cited upper bound on latency is 60ms to have an acceptable VR experience; however, it should ideally be below 20ms to be imperceptible [10]. Carmack's *time warping* method [11] provides one approach to reducing the effective latency.

We alternatively present a predictive filtering approach. Predictive tracking methods have been developed over decades of VR and AR research [12], [13], [14], but they are usually thought to give mixed results because of difficult motion modeling challenges, obtaining dense, accurate data for the filter, and the length of the latency interval. Although historically, latency intervals have been as long as 100ms, in a modern setting it is in the 30 to 50ms range due to improvements in sensing and graphics hardware. Furthermore, MEMS gyros provide accurate measurements of angular velocity at 1000Hz. We have found that predictive filters need only a few milliseconds of data and yield good performance up to 50ms.

We developed and compared three methods (see Figure 9):

1) No prediction: Just present the updated quaternion to the renderer.
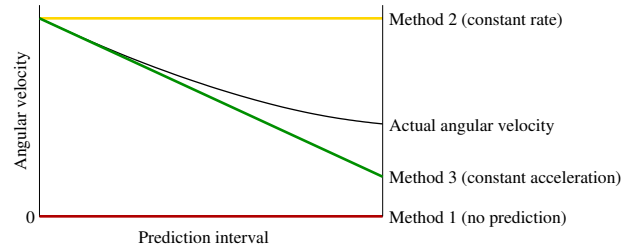2) Constant rate: Assume the currently measured angular velocity will remain constant over the latency interval.
3) Constant acceleration: Estimate angular acceleration and adjust angular velocity accordingly over the latency interval.

The first method assumes the head is stationary during the latency interval. The second method replaces $\Delta t = 0.001$ with $\Delta t = 0.021$ in (2). The third method allows the angular velocity to change at a linear rate when looking into the future. The angular acceleration is estimated from the change in gyro data (this is a difficult problem in general [15], but works well in our setting).

Errors in estimating the current angular velocity tend to be amplified when making predictions over a long time interval. Vibrations derived from noise are particularly noticeable when the head is not rotating quickly. Therefore, we use simple smoothing filters in the estimation of current angular velocity (Methods 2 and 3) and current angular acceleration (Method 3), such as Savitzky-Golay filters, but many other methods should work just as well. We also shorten the prediction interval for slower rotations.

A simple way to evaluate performance is to record predicted values and compare them to the current estimated value after the prediction interval has passed. Note that this does not compare to actual ground truth, but it is very close because the drift error rate from gyro integration is very small over the prediction interval. We compared the performance of several methods with prediction intervals ranging from 20ms to 100ms. Figure 9 shows error in terms of degrees, for a prediction interval of 20ms, using our sensors over a three-second interval.

Numerically, the angular errors for predicting 20ms are:

Method	Avg Error	Worst Error
1	1.46302°	4.77900°
2	0.19395°	0.71637°
3	0.07596°	0.35879°

.

A user was wearing the Rift and turning their head back and forth, with a peak rate of about 240 deg/sec,
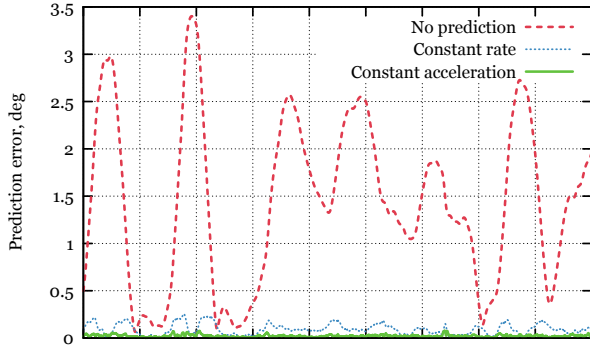

Fig. 8. A depiction of the three prediction methods in angular velocity space (represented as one axis).

Fig. 9. The prediction error in degrees for a 20ms interval for the three methods.

Fig. 10. We enforce kinematic constraints to estimate position.

which is fairly fast. This is close to reported peak velocities in published VR studies [13], [14].

During these motions, the acceleration peaked at around 850 deg/sec^2. Typical, slower motions, which are common in typical usage, yield around 60 deg/sec in velocity and 500 deg/sec^2 in peak accelerations. For both slow and fast motions with a 20ms prediction interval, Method 3 is generally superior over the others. In another experiment, even with an interval of 40ms, the average case is 0.17 and worst case is 0.51 degrees.

VIII. POSITIONAL TRACKING

Tracking the head position in addition to orientation provides a greater sense of immersion in VR. Providing accurate position estimates using only sensors described in Section II is extremely challenging. Using more sensors, the problems are greatly alleviated, but the cost increases. Solutions include computer vision systems and a body suit of IMUs. In our case, we have tried to push position estimation as far as possible using one IMU. We report here on our limited success and remaining difficulties. One of the greatest obstructions to progress is that the sensor mapping for all of our sensors contain constant linear velocity in their preimages. In other words, traveling at constant linear velocity is unobservable by any of our sensors (unless the magnetic field is non-constant over small displacements).

The coordinate frame origin for positional tracking is located at the midpoint between the retinas. Call this the *retinal center frame* (RCF). Without position information, this point remains fixed in the virtual world before the left and right eye views are extruded to form a stereo pair. That is, the position of the RCF is always $p = 0$. One general strategy is to design a mapping $p = f(q)$ in which q represents the head orientation and p is the position of the RCF. In other words, the most likely position p
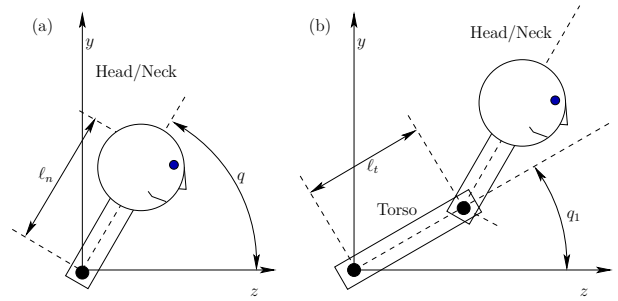
is specified for each given orientation q. Perhaps this relationship could be learned from large data sets.

A simple example, which is currently in use in the Oculus Rift Software Development Kit (SDK), is to move the rotation center to the base of the neck, rather than at the RCF. This improves the experience by accounting for positional offset of the eyes as the head rotates (see Figure 10(a)). If the distance from the RCF to the base of the neck is ℓ_n, then the corresponding mapping is

$$p = f(q) = q * (0, \ell_n, 0) * q^{-1}. \tag{13}$$

If the VR experience requires large movements of the human body, then the above solution is insufficient. It is tempting to use the accelerometer to determine position. Recall Figure 4(a), in which $a = a_g + a_l$. If the estimated gravity vector $\tilde{a}_g$ can be subtracted from the observed vector sum $\tilde{a}$, then the remainder corresponds to measured linear acceleration $\tilde{a}_l$. From basic calculus, double integration of the acceleration produces the position. The main trouble is that offset error and extreme sensitivity to vibration cause fast drift error. A fixed offset in acceleration corresponds to quadratic error growth after double integration (imagine attaching fictitious thrusters to your head!).

One promising possibility is to use kinematic constraints to narrow down the potential motions. For example, if the user is known to be sitting in a chair, then their motions can be kinematically approximated as a two-link spatial mechanism, as shown in Figure 10. The bar at the origin represents a torso with length ℓ_1 and orientation q_1. The second bar represents a head (and stiff neck). Note that the head orientation is still q, and is computed using methods described in the previous sections, independently of q_1. The RCF position is computed as

$$p = f(q) = q_1 * (0, \ell_t, 0) * q_1^{-1} + q * (0, \ell_n, 0) * q^{-1}. \tag{14}$$

Let r_1 and r denote the first and second terms above, respectively, to obtain $p = r_1 + r$.

Next we estimate the angular velocity $\hat{\omega}_1$ of the torso from the measured linear acceleration $\tilde{a}_l$ (in the global
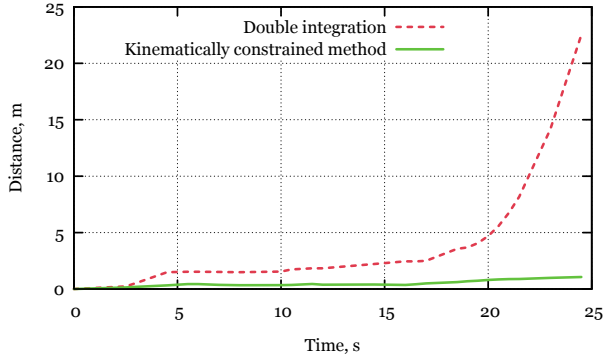
Fig. 11. Local kinematic constraints yield lower position error.

frame) of the head. Once $\hat{\omega}_1$ is computed, $\hat{q}_1$ can be found using methods from Section III. The method iteratively computes and refines an estimate of the linear velocity $\hat{v}_1$ of the head. First, $\hat{v}_1$ is computed by integration of $\tilde{a}_l$. The resulting value may drift and therefore needs to be updated to fit the kinematic model. Observe that $\hat{v}_1$ is the vector sum of velocities of the two bars:

$$\hat{v}_1 = \hat{\omega}_1 \times r_1 + \hat{\omega} \times r. \tag{15}$$

This equation can be rewritten as an underdetermined system of linear equations $A\tilde{\omega}_1 = b$ with a singular matrix A. By projecting the right hand side b on the span of A we can solve the resulting system $A\hat{\omega}_1 = b_A$, and update the current linear velocity estimate using

$$\hat{v}_1 = b_A + \hat{\omega} \times r. \tag{16}$$

We compared the performance of this method to straightforward double integration of $\tilde{a}_l$. The position error is shown in Figure 11. Using kinematic constraints significantly reduces position drift. After one minute of running time, the position error of the double integration exceeds 500m, whereas the kinematic-based method keeps the error within 1.1m.

This positional tracking method is only effective over a few seconds. It may be useful as a complement to other sensing systems; however, in its present form, it is insufficient as a standalone technique. Possibilities for future research include using more accurate accelerometers and experimenting with stronger kinematic constraints and limited motions.

IX. CONCLUSIONS

We presented an approach to head tracking that uses low-cost MEMS sensors and is deployed in tens of thousands of virtual reality headsets. Crucial aspects of the system are quick response to head movement, accurate prediction, drift correction, and basic position

information. The greatest challenge ahead is to obtain reliable position estimates over longer time intervals using MEMS sensors or a comparable low-cost technology. In our most recent prototypes, we have achieved this the straightforward way by combining the techniques in this paper with position data from a camera that observes infrared LEDs on the Rift surface.

The full C++ code of our methods is available at:

`https://developer.oculusvr.com/`

REFERENCES

[1] S. A. Zekavat and R. M. Buehrer, *Handbook of Position Location*. IEEE Press and Wiley, 2012.

[2] G. Welch and E. Foxlin, "Motion tracking: no silver bullet, but a respectable arsenal," *IEEE Computer Graphics and Applications*, vol. 22, no. 6, pp. 24–38, 2002.

[3] S. Sastry, *Nonlinear Systems: Analysis, Stability, and Control*. Berlin: Springer-Verlag, 1999.

[4] D. Titterton and J. Weston, *Strapdown Inertial Navigation Technology*, 2nd ed. Institution of Engineering and Technology, Oct. 2004.

[5] S. M. LaValle, *Sensing and Filtering: A Fresh Perspective Based on Preimages and Information Spaces*, ser. Foundations and Trends in Robotics Series. Delft, The Netherlands: Now Publishers, 2012, vol. 1: 4.

[6] A.-J. Baerveldt and R. Klang, "A low-cost and low-weight attitude estimation system for an autonomous helicopter," in *IEEE International Conference on Intelligent Engineering Systems*, 1997, pp. 391–395.

[7] J. Favre, B. Jolles, O. Siegrist, and K. Aminian, "Quaternion-based fusion of gyroscopes and accelerometers to improve 3D angle measurement," *Electronics Letters*, vol. 42, no. 11, pp. 612–614, 2006.

[8] W. Higgins, "A comparison of complementary and Kalman filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-11, no. 3, pp. 321–325, 1975.

[9] D. G.-E. amd G. H. Elkaim, J. D. Powell, and B. W. Parkinson, "Calibration of strapdown magnetometers in magnetic field domain," *Journal of Aerospace Engineering*, vol. 19, no. 2, pp. 87–102, 2006.

[10] M. Abrash, "Latency: the sine qua non of AR and VR," Dec. 2012. [Online]. Available: http://blogs.valvesoftware.com/abrash/latency-the-sine-qua-non-of-ar-and-vr/

[11] J. Carmack, "Latency mitigation strategies," Feb. 2013. [Online]. Available: http://www.altdevblogaday.com/2013/02/22/latency-mitigation-strategies/

[12] R. T. Azuma, "Predictive tracking for augmented reality," Ph.D. dissertation, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.

[13] U. H. List, "Nonlinear prediction of head movements for helmet-mounted displays," Operations Training Division Air Force Human Resources Laboratory, Tech. Rep., Dec. 1983.

[14] B. R. Smith Jr., "Digital head tracking and position prediction for helmet mounted visual display systems," in *AIAA 22nd Aerospace Sciences Meeting*, Reno, NV, 1984.

[15] S. Ovaska and S. Valiviita, "Angular acceleration measurement: a review," *IEEE Transactions on Instrumentation and Measurement*, vol. 47, no. 5, pp. 1211–1217, 1998.