# Implementation and Evaluation of a 50 kHz, 28μs Motion-to-Pose Latency Head Tracking Instrument

Alex Blate, Mary Whitton, *Life Member IEEE*, Montek Singh, *Member, IEEE*, Greg Welch, *Senior Member IEEE*, Andrei State, Turner Whitted, *Fellow, IEEE*, and Henry Fuchs, *Life Fellow, IEEE*

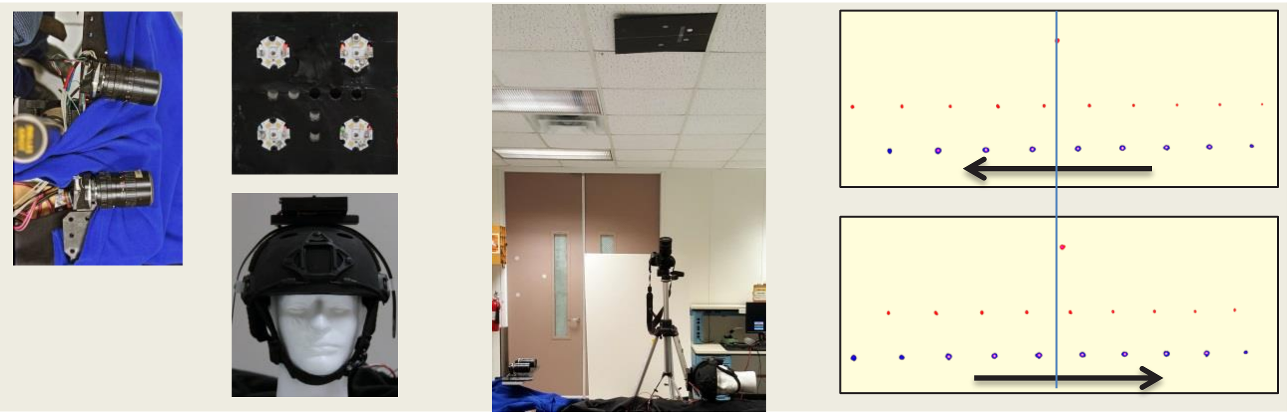

Fig. 1. Left: stereo duo-lateral photodiode sensors and optics assembly facing target helmet equipped with 4 high-intensity LED emitters. Center: Motion-to-pose latency measurement. In actual operation, sensors would be located above the user and pointing down at an upright user. Right: example video frames showing the tracker-controlled laser spot (top) and quadrature timing traces. In the measurement experiments, the tracker turns the laser on when, based on measured pose, the laser would intersect the ceiling within a 500μm-wide region. Rotational velocity is approximately 500°/sec.

**Abstract**—This paper presents the implementation and evaluation of a 50,000-pose-sample-per-second, 6-degree-of-freedom optical head tracking instrument with motion-to-pose latency of 28μs and dynamic precision of 1-2 arcminutes. The instrument uses high-intensity infrared emitters and two duo-lateral photodiode-based optical sensors to triangulate pose. This instrument serves two purposes: it is the first step towards the requisite head tracking component in sub-100μs motion-to-photon latency optical see-through augmented reality (OST AR) head-mounted display (HMD) systems; and it enables new avenues of research into human visual perception – including measuring the thresholds for perceptible real-virtual displacement during head rotation and other human research requiring high-sample-rate motion tracking. The instrument's tracking volume is limited to about 120×120×250mm but allows for the full range of natural head rotation and is sufficient for research involving seated users. We discuss how the instrument's tracking volume is scalable in multiple ways and some of the trade-offs involved therein. Finally, we introduce a novel laser-pointer-based measurement technique for assessing the instrument's tracking latency and repeatability. We show that the instrument's motion-to-pose latency is 28μs and that it is repeatable within 1-2 arcminutes at mean rotational velocities (yaw) in excess of 500°/sec.

**Index Terms**—Tracking, head tracker, lateral-effect photodiodes, augmented reality, low-latency augmented reality, dynamic tracking error, perception, motion tracking

## 1 Introduction

A low-latency, high-sample-rate, head tracker is a necessary component of Optical See-Through (OST) Augmented Reality (AR) head-mounted display (HMD) systems. In OST AR, the user sees virtual objects optically combined with his view of the real world. To maintain real-virtual alignment, the AR display must continually update the displayed locations of virtual objects such that their positions are consistent with the user's pose. Latency between user movement and display updates, also known as motion-to-photon latency, causes perceptible displacements between real and virtual objects. This is perceived as discontinuous movements, vibrations, or swimming of the virtual object. User discomfort, simulator sickness, and disruption of presence are among the possible negative impacts upon the user's experience.

Motion-to-photon latency can be decomposed into two independent latency sources: tracking latency and display latency. Tracking latency, or motion-to-pose latency (MTPL), is the time between a change in the user's pose and the tracker outputting a pose sample reflecting said change. Display latency is the time between a pose sample appearing on the tracker's output and the display's corresponding change in output due to the new pose.

Display latency has been addressed in prior work, where display systems with net motion-to-photon latencies as low as 80μs have been demonstrated [1] [2] [3]. Prior to the present work, no tracker

- *Alex Blate is with UNC-Chapel Hill. E-mail: blate@cs.unc.edu.*
- *Mary Whitton is with UNC-Chapel Hill. E-mail: whitton@cs.unc.edu.*
- *Montek Singh is with UNC-Chapel Hill. E-mail: montek@cs.unc.edu*
- *Greg Welch is with The University of Central Florida. E-mail: welch@ucf.edu*
- *Andrei State is with UNC-Chapel Hill and InnerOptic Technology, Inc. E-mail: andrei@cs.unc.edu*
- *Turner Whitted is with TWI Research, LLC. E-mail: jtw@twilab.com*
- *Henry Fuchs is with UNC-Chapel Hill. E-mail: fuchs@cs.unc.edu*

*Manuscript received 10 Sept. 2018; accepted 7 Feb. 2019.*
*Date of publication 17 Feb. 2019; date of current version 27 Mar. 2019.*
*For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.*
*Digital Object Identifier no. 10.1109/TVCG.2019.2899233*

with the requisite latency and sample rate—and allowing unrestricted, natural head motion—was known to exist; this issue was raised by the aforementioned literature. Thus, to effectively demonstrate display latencies, the low-latency displays referenced above tracked pose along one or more axes using rigidly-attached mechanical position trackers, such as rotary shaft encoders.

The realization that motion-to-photon latency is the proximate cause of perceptible real-virtual displacement in OST AR HMD systems leads to questions about perceptual thresholds and tolerances for such latency and displacements. These aspects of human visual perception are largely unexplored, particularly at sub-100μs time scales; credible research into these phenomena requires the ability to track user head pose without mechanically constraining or interfering with the user's natural movements. Other avenues of human research would also benefit from high-sample-rate/high-frequency motion tracking.

The present work addresses the tracking latency component of motion-to-photon latency and presents what is, to the best of our knowledge, the first instrument enabling the aforementioned perceptual research. Specifically, we demonstrate a six-degree-of-freedom, optical head tracking instrument with temporal performance at least twenty times better than any previously described tracker—our instrument's MTPL is about 28μs at a sample rate of 50 kHz. The only physical connection between the instrument and the tracked target is a flexible power cable. The instrument's tracking volume is limited to about 120×120×250mm but allows for the full range of natural head rotation[1] and is sufficient for research involving seated users. As discussed in section 6, the instrument's tracking volume is scalable in several ways; we explore the likely trade-offs of such scaling vis-à-vis sample rate/latency, pose uncertainty, and spatial resolution. Much potential exists beyond the present implementation.

The instrument calculates pose by triangulation from the 2D projections of four high-intensity infrared light-emitting diodes (LEDs) upon two duo-lateral photodiode sensors. The instrument's geometry is designed to maximize its signal-to-noise ratio and spatial resolution, i.e., we made a conscious trade-off between tracking volume and spatial/temporal performance. It was our intention that the instrument's performance exceed that which is required for the aforementioned perceptual studies—particularly with respect to temporal performance

Typically, the characterization of a tracker's performance includes measuring it with respect to some other reference [4] [5] [6]. We do not know of and do not have access to instrumentation with both the spatial resolution and sample rate necessary to measure our tracker's MTPL directly. We therefore developed a novel technique to measure the instrument's temporal performance and spatial repeatability without high-speed, high-precision instrumentation. We show, analytically, that dynamic tracking error (the difference between the tracker's pose output and the user's true pose) is proportional to pose velocity, tracker sample rate, tracking latency, and noise. Our technique, which uses three inexpensive laser pointers and a video camera, lets us measure dynamic tracking error (distance) and pose velocity from which we calculate tracking latency (time). The direct measurement of dynamic tracking error is, in fact, a direct measurement of real-virtual displacement, i.e., a quantity we wish to minimize in OST AR HMD systems. Figure 1 shows elements of the tracker instrument and the measurement system.

Our instrument serves as a starting point for the design and implementation of future low-latency, high-sample-rate head trackers. In its present form, our instrument is immediately useful for certain specialized AR research, such as research into low-latency OST AR displays and systems or applications that inherently involve a seated user, such as aviation. Moreover, our tracker's low latency and 50 kHz sample rate make it a novel tool to study heretofore under-explored aspects of human visual perception, including aspects directly related to OST AR.

## 2 Related Work

The negative effects of latency in virtuality, affecting both the Virtual Reality (VR) and AR modalities, have long been recognized [7] [8] [9] [10]. Perceptual sensitivity to latency in projective AR systems has been studied, e.g., by Jerald [11], although the finest temporal resolution in Jerald's experiments was on the order of 2-3ms. The practical impacts of latency in AR systems have also been explored [12]; it is clear that latency negatively impacts presence, user comfort, and task performance.

Lincoln et al. demonstrated a mechanically-tracked OST AR display with mean motion-to-photon latency of 80μs [1] [2]; the authors also specifically identified the need for low-latency tracking to enable practical use of such displays. Another mechanically-tracked low-latency projective AR display was demonstrated by Regan et al. [3]; this work provides a good introduction to the effects of latency during head rotation.

Despite being nearly two decades old, the 3rdTech HiBall™ tracker [8] is still one of the highest-performance trackers in terms of sample rate (750-2,000Hz) and motion-to-photon latency (~3μs). The HiBall, originally designed at UNC Chapel Hill, is among a relatively small class of tracking devices that make use of lateral-effect photodiodes (LEPDs) [13] [14] [15]. Our choice of duo-lateral photodiode sensors [16], a type of LEPD, was inspired in part by our understanding of the HiBall system.

Bapat et al. [17] show, through offline simulations, that rolling-shutter CMOS camera sensors have the potential to be used to implement low-latency, high-sample rate trackers; no online, real-time implementation of such a tracker has been demonstrated and it is unclear whether their GPU-based algorithms will be tractable on embedded platforms in the foreseeable future.

We were unable to identify any previously disclosed tracker with a sample rate in the kilohertz range and sub-millisecond tracking latency.

Analytical modelling of tracking and tracking errors has been explored in the literature. Allen [18] presents a stochastic tracking model useful for some a priori optimization of generalized tracking systems; dynamic error is not a component of this model. Dynamic tracking error itself is discussed in the context of video-based tracking [19], though at considerably coarser time scales than in the present work.

The actual measurement of AR and VR system latency has also been explored [20] [21] [22] [23] [7]; Welch et al. briefly discuss direct measurement of tracker latency [8]. The use of one "reference tracker" to characterize another tracker as well as potential issues with this approach are described in detail by Vorozcovs et al. [4].

Measurement and quantification of tracker accuracy and precision has also seen a number of approaches [6] [9] [24] [5] and primarily focuses upon absolute tracking accuracy under essentially static (i.e., low-velocity) conditions.

Methods for spatial-optical tracking and localization based on stereo point correspondences are abundant within the computer vision literature. Canonical examples of robust algorithms include those of Luong [25] and Zhang [26], though Hartley [27] points out that, with well-conditioned inputs, less sophisticated algorithms can be nearly as effective. All of these algorithms rely on the inference of many point correspondences between images. In our case, as will be discussed below, we do not have to solve the correspondence problem; additionally, we are working with a small number of points

[1] ±45° in nod and tilt and 360° in rotate/yaw; lateral translation is limited to about ±30mm (left/right and fore/aft). Vertical position can be tracked across total range of 250mm.

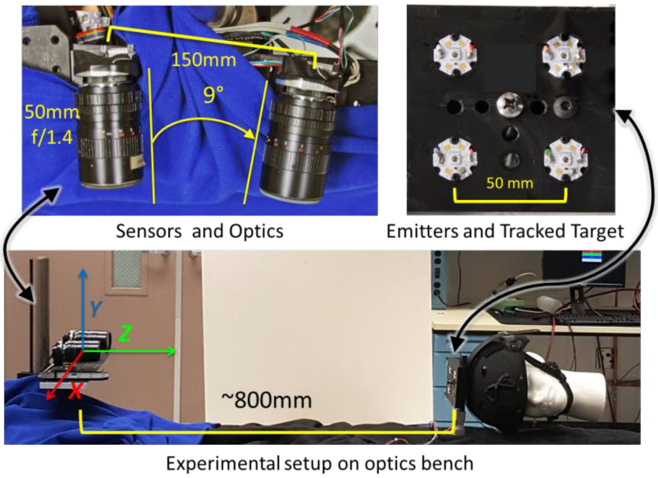
150mm
9°
50mm
f/1.4
Sensors and Optics
50 mm
Emitters and Tracked Target
Y
Z
X
~800mm
Experimental setup on optics bench

Fig. 2: Tracking instrument physical overview. Top left: top view of sensor assembly; the infrared low-pass filters are attached to the front of the lenses. The sensors are spaced 150mm apart and toed in such that their optical axes intersect at a 9° angle. Top right: front view of tracked target with emitters. The emitters are co-planar and located at the corners of a 50×50mm square. Bottom: lab setup for latency measurements.

(typically four points). The triangulation algorithm used in our tracker is based on Sutherland's work [28] and the resulting implementation is deterministic and executes in constant time.

## 3 High-Speed, Low-Latency Head Tracking Instrument

Our tracking instrument is designed to minimize tracking latency and maximize pose sample rate while offering the best possible accuracy and repeatability.

The exact perceptual thresholds for motion-to-photon latency in OST AR are not known. Our aim was to construct a tracker whose performance likely well exceeds these thresholds (i.e., the practical needs of OST AR), such that it can serve as an instrument for measuring perceptual thresholds and for verifying and benchmarking other trackers. This raises the question of what our performance objectives should be.

Bounds on head rotational velocity and visual acuity lead us to make informed engineering estimates of performance objectives. During typical daily activities, head rotational velocities of 300-500°/sec in yaw (left-right rotation) are common [30]; humans are most sensitive to real-virtual displacement during yaw [*idem*]. A person with normal (20/20) vision has static visual acuity of one arcminute [31]. Combining these known properties of rotational velocity and spatial perception guides us to a conservative nominal performance requirement for dynamic tracking error.

Let us consider the requirements for a tracker capable of tracking head rotations (yaw) of up 300°/sec with a dynamic tracking error of no more than one arcminute. 300°/sec = 18,000 arcminutes/sec = 55.5µs/arcminute. Thus, a motion-to-photon latency exceeding 55.5µs would result in a real-virtual displacement of greater than one arcminute. Because display latency will be non-zero, the MTPL must be less than 55.5µs. By the Sampling Theorem [32], pose must be sampled at 36 kHz or higher to resolve one arcminute at this velocity.

One can adjust the parameters used in the calculation above (velocity and resolution) but, for any reasonable values, one finds that the tracker must have a sample rate in the tens of kilohertz, latency in the tens of microseconds, and precision on the order of one arcminute.

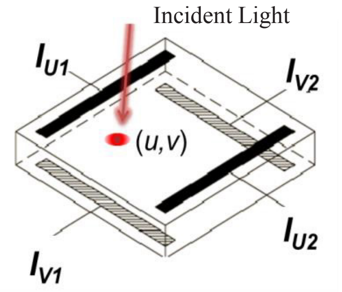

Fig. 3: Duo-lateral photodiode (diagram based on Figure 1 in [35]). The device's output signals are proportional to the location of the centroid of light hitting the device. The red ellipse represents a light spot projected upon the sensor.

### 3.1 Overview and Performance Summary

The tracking instrument, hereinafter referred to as the "tracker," tracks the head of a seated user in six degrees of freedom (DOF). The tracker follows the "outside-in" tracking paradigm, wherein stationary optical sensors are stimulated by emitters located on the moving tracked target. The only connection to the tracked target is a flexible power cable, thus allowing natural, unrestricted head motion.

The optical sensors are mounted above the user, facing down; the emitters are located on the top of the user's head facing up. From the user's perspective, the tracker's X, Y, and Z axes correspond to the user's fore-aft, left-right, and up-down axes, respectively. Rotation about the Z-axis corresponds to left-right head rotation, rotation about the Y-axis corresponds to nod, and rotation about the X-axis corresponds to tilt. The sensors' mean optical axis is aligned with the Z axis; the line connecting the sensors' optical centers is parallel to the X-axis.

The tracker outputs 6-DOF poses at a rate of 50 kHz. The tracker's motion-to-pose latency (MTPL) is 28µs. The tracker's dynamic error is about one arcminute at a yaw velocity of up to 500°/second. Pose uncertainty is less than or equal to one arcminute in orientation, under 10µm in X and Y position, and under 100µm in Z position. The tracking volume is nominally 120×120×250mm (X, Y, Z) at a working distance of 750-950 mm (800mm typical). Tracking is maintained beyond 950mm but with higher pose uncertainty.

### 3.2 Architecture

Our system consists of a fixed sensor assembly and a mobile (tracked) panel as shown in Figure 2. The panel is equipped with IR light-emitting diode (LED) emitters whose signals are observed by the sensing elements. The assembly comprising the emitters and the panel is the tracked target. Emitters and sensors are controlled and synchronized by circuitry which also calculates the tracked target's pose in relation to the sensing head.

#### 3.2.1 Sensors

The previously-stated temporal and spatial requirements immediately lead to lower bounds on the tracking sensors' sample rates and spatial resolution. We chose duo-lateral photodiode position sensing devices as our optical sensors. These sensors have both the bandwidth (400 kHz) and the spatial resolution (sub-micron over the surface of the device) we require [16]. (The device's datasheet states a -3dB bandwidth of 270 kHz with the typical supply voltage of ±15VDC. We found that increasing the supply voltage to ±17VDC yields another ~130 kHz of bandwidth. This voltage is still within the specified operating range of the device and the increased (reverse) bias voltage increases bandwidth.) In many respects, the tracker is designed around these sensors.

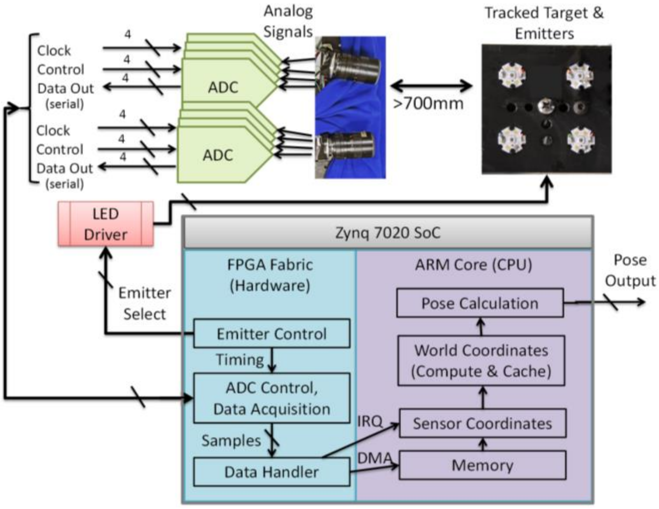

Fig. 4: Tracking instrument block diagram. Top right: sensors and tracked target with emitters. Top left: digitizers and emitter LED driver. Bottom: Xilinx Zynq system-on-chip processor (integrated FPGA and ARM CPU). Real-time control of the emitters and ADCs is performed in hardware. Sensor data is written by DMA into the ARM CPU's main memory, whereupon an

Duo-lateral photodiodes are a type of LEPD. Figure 3 shows a diagram of such a device. Light hitting the surface of the device causes currents to flow through the four indicated electrodes. The location of the centroid of the incident light is calculated from these four currents. Unlike other types of LEPDs, duo-lateral photodiodes measure X and Y independently; this improves both linearity and accuracy.

LEPD modules, each comprising a sensor and preamplifier circuitry, are attached to a mount and fitted with 50mm lenses and infrared low-pass filters. Two of these modules are shown at the top of Figure 4. Mounted rigidly in relation to each other, these modules constitute the fixed "sensing head" of the tracker. The sensing head can be relocated, as needed, without invalidating sensor calibration. The sensors have a nominal active area of 10×10mm. Sensor linearity is specified for the inner 8×8mm area. As linearity falls off rapidly on the periphery (as is typical of all LEPDs), we reject readings that fall outside of the inner region.

Each sensor has four analog outputs. We have direct access to and control over the entire analog signal path—including, for example, how and when the signals are digitized. (In contrast, the underlying analog signals in, for example, a typical CMOS image sensor, are conditioned and digitized within the sensor, making an important portion of the signal path inaccessible to design engineers.) Sensor outputs are digitized simultaneously by 18-bit, 1 MSPS (million samples per second) SAR (sequential approximation register) analog-to-digital converters (ADCs)[2] [33]. The sensors, analog signals, ADCs, and digital signals are shown at the upper left area of the system block diagram in Figure 4. The digitizer design is modular: each sensor's four channels connect to one digitizer board. The circuit board was designed in-house and uses commercial-off-the-shelf (COTS) components.

### 3.2.2 Emitters

The sensors are stimulated by high-intensity 940nm infrared LED emitters [34]. This wavelength is close to the sensor's peak sensitivity. Four emitters are securely mounted to a panel. They are arranged in a 50×50mm square and are co-planar within 500μm. The emitters fit into precisely machined shallow holes in the panel, ensuring lateral alignment within about 25μm. The resulting assembly is the tracked target and is shown in Figure 4 (top right), as well as in Figure 2 (top right). The emitters are driven at their maximum-rated pulse current by a custom board. The resulting luminous intensity is such that the sensors are stimulated to about 95% of full-scale at the minimum working distance of about 700mm and to about 80% of full-scale at the nominal 800mm working distance. The resulting signal-to-noise ratio (SNR) lies slightly above 100dBmV.

Because we use infrared emitters and infrared low-pass filters on the sensors, the tracking instrument can be (and is designed to be) used under normal lighting conditions. Note, however, that ambient infrared, such as that produced by incandescent light bulbs, will interfere with the tracker, potentially causing tracking errors.

### 3.2.3 FPGA and Embedded Software

As shown at the bottom of Figure 4, control and processing is performed by a Xilinx™ Zynq™ 7020 "System on Chip" processor, which contains both an FPGA (blue) and two ARM™ CPU cores (violet); we only use one of the two ARM cores. Emitter control, timing, ADC control, and data acquisition are performed in modules within the FPGA fabric. Digitized sensor samples are directly written to the ARM core's main memory, after which the tracker software is triggered by an interrupt. The emitter control FPGA circuitry is the source of timing for all other real-time processes, up to and including the interrupt to the ARM CPU.

The tracker software converts the raw sensor samples into sensor coordinates. These coordinates are used to calculate the 3D world position of each emitter. Pose is calculated from three or more emitter world positions using vector arithmetic. The software is a monolithic C/C++ program and, using Xilinx-provided APIs, it runs directly on the CPU, i.e., there is no operating system.

Unlike most trackers, our tracker can interface directly to a display (or to other pose consumers), e.g., via a dedicated high-speed serial or parallel interface. As such, we do not incur the latency associated with a higher-level interconnect, such as USB or Ethernet. For the purpose of analysis, we treat pose data transfer latency as a component of display latency rather than tracking latency.

### 3.2.4 Sensor and Pose Sampling

The sensors measure the position of the centroid of all light incident upon them. To obtain useful measurements, the light must originate from a single emitter at a time. Hence, the emitters are activated sequentially, one at a time. Each emitter is kept on for a constant duration; this duration is equal to the tracker's pose sample period. At some point during each such period, the ADCs begin sampling the sensors' outputs; the sensors may be sampled one or more times during the period. Importantly, the phase of the ADC's sampling with respect to the emitter's period is constant and is controlled by the emitter control circuitry in the FPGA.

Once digitized samples are written into ARM CPU memory, an interrupt triggers the software layer to process the new samples. The sensor coordinates calculated from the raw samples are the coordinates of the centroid of the projections of the respective emitter on each sensor. The stereo pair of 2D coordinates of corresponding emitter "sightings" is used to calculate the 3D position of each emitter relative to the sensing head. The 2D-to-3D calculation requires a minimum of two sensors but readily extends to three or more concurrent sensor sightings of the same emitter.

A pose can be calculated given the (non-co-linear) 3D positions of three or more emitters. In practice, we always use all four emitters to compute pose. Should one or more emitters become occluded or move outside the sensors' combined field-of-view, the tracker

[2] 18-bit precision places the ADC's noise floor below that of the (bipolar) signal. At 1 MSPS, the acquisition overhead's contribution to MTPL is only 250 ns (see section 0). The SAR ADC architecture is well-suited for measuring instantaneous voltages and has minimal latency; flash or dual-slope ADC architectures would also be acceptable for this application.

outputs an "error pose" comprising all IEEE floating point NaNs ("not a number") and will continue to do so until all emitters are again visible to both sensors. Optionally, at runtime, the software can be configured so that poses can be calculated from only three 3D emitter locations if four are not available.

Each pose is calculated from the position of the currently sampled emitter and the most recently measured positions of the other three emitters. The tracker outputs one pose per pose sample period.

We are, in fact, using "old" data in our pose calculations: once we have sampled an emitter, we implicitly assume that it does not move until we sample it again. As would be expected, if the target is in motion, this does introduce an error. This is an issue in many other trackers. For example, HiBall [8] employs an extended Kalman filter that, in simple terms, uses each new emitter "sighting" to update one pose dimension and compensate for "old" readings.

We considered compensating for this "old" data error. However, as discussed below, we determined analytically and demonstrate empirically that with the tracker's high sampling rate, the error is so small as to be insignificant

We calculated the "old" data error for two worst-case scenarios: pure yaw and pure translation parallel to the sensors (translation in X-Y). Our pose sample period is 20µs, thus the "oldest" sample used in a pose calculation will be 60µs old. Based on our calculations[3], the error due to "old" data is less than one arcminute at rotational velocities up to 555.5°/second and less than 10µm at translational velocities up to 500mm/s. These velocities are well above the extrema of human movements during normal daily activities [30]. The error itself is small (near, but still above the noise floor), is linear in pose velocity, continuous, and is independent of direction of travel. To the extent that filtering could improve tracking accuracy with little or no impact to latency, compensation for this error source is a good topic for future work.

### 3.3 Latency and Timing Analysis

A detailed a priori analysis of the tracker's latency helps us better understand how the tracker works, lets us definitively identify all sources of latency within the tracker, and lets us make predictions about how we expect the tracker to perform. Other than the pose sample period and the phase of ADC conversion, the timing figures presented below were obtained from a combination of stated "typical" values in the ADC [33] and sensor [16] datasheets, oscilloscope traces, and software instrumentation.

Figure 5 depicts the tracker's operation over time. Two time scales are shown: the upper portion of the figure shows six pose sample periods while the lower portion zooms in on the sampling and computation of a particular pose output.

The pose sample period is 20µs; the choice of this specific value is discussed below. As shown by the "Emitter" rows, the emitters are activated in a round-robin fashion with one emitter active during each sample period. Each emitter transition results in a change in position of the light spots projected upon each sensor. In response, the sensors' outputs change. It takes some time for the outputs to settle; this time is related to the large signal response in both sensor and amplifier, and to thermal effects in the LEDs. Experimentally, we found that, by about 18µs into the sample period, the rate of change in sensor output, while non-zero, is small enough for us to get consistent samples.

The ADCs begin conversion at exactly 18.9µs into the sample period. The rising edge of the "Convert" waveform (lower half of the figure) represents the moment when the conversion signal is sent to the ADCs. At this time the ADCs' track and hold buffers transition into the hold state and conversion begins. These particular ADCs' acquisition window is 500ns; the value being converted can be understood as the mean of the input signal over the preceding 500ns and is representative of the emitter's position at the midpoint of the acquisition window, i.e., 250ns prior to conversion. This is indicated by the "Acquire (midpoint)". The 250ns acquisition latency, then, is the first component of the tracking latency[4].

Conversion itself nominally takes 500ns [33]. The ADCs assert a signal when conversion is complete. When all ADCs have completed conversion, the samples are read out by the FPGA, which takes 320ns (the "Clock Out" waveform). The data is marshalled and transferred, via DMA, to ARM CPU memory; this takes approximately 300ns. An interrupt, generated at the conclusion of the DMA, triggers the tracking software, which processes the new samples and calculates a new pose. This takes approximately 6.5µs and is represented by the "Compute" waveform. The pose is immediately available on the tracker's output after computation. The net tracking latency, then, is about 8µs.

We are now in a position to estimate motion-to-pose latency (MTPL). Any motion that occurs after the acquisition of the sample used to compute pose $P_i$ will appear in pose $P_{i+1}$. As illustrated by the time intervals labelled "MTPL($P_i$)" in the figure, the latency then is the sum of the sample period and the tracking latency. In the present analysis, we estimate our tracker's MTPL to be about 28µs.

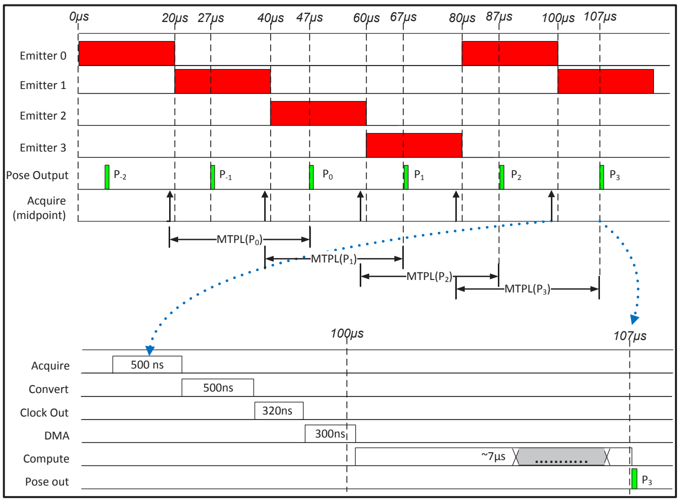

Fig. 5: Tracker timing diagram. Top: emitter round-robin sequence, phase of pose outputs, and ADC acquisition windows. Middle: motion-to-pose latency. Bottom: data

### 3.4 Spatial Resolution and Tracking Volume

As stated at the outset of this section, our tracking instrument was designed to minimize tracking latency and maximize pose sample rate, while also offering the best possible accuracy and repeatability. We have discussed the primary design decisions relating to the tracker's temporal performance. But the other half of the equation, as it were, is the tracker's spatial resolution: the tracker must be able to measure very small changes (one arcminute) in pose over short time intervals (tens of microseconds). We now discuss the inter-dependent factors that determine spatial resolution and how these affect, among other things, tracking volume.

#### 3.4.1 Spatial Resolution

Let us consider the factors that contribute to a sensor's spatial

[3] In short, the mean "age" of the samples in any given calculation is 30µs. We then find the maximum velocity at which, e.g., the rotation over 30µs would be 1 arcminute.

[4] Any sensor will have some acquisition latency, i.e., the sampled value will represent the state of the input at some time in the past. For our tracker, this latency is small, largely because we chose high-performance ADCs. But, for example, the acquisition latency of a 20,000 fps video camera would be about 25µs which, in our case, would dwarf our other latency components.

resolution—which is finite and noise-limited. For a given signal-to-noise ratio (SNR), the sensor will be able to resolve some number $N$ distinct locations along each of its axes; increasing the SNR increases $N$ (up to some limit). The sensor's field-of-view (FOV) is determined by the focal length of the lens we choose and the sensor's dimensions. If the angular FOV is $\theta_{FOV}$, the sensor will have an angular resolution of $2\theta_{FOV}/N$ (twice its angular sampling resolution). Note that $\theta_{FOV}$ is inversely proportional to focal length[5]. Using the small angle approximation for tangent, at distance $D$, this corresponds to a linear resolution of about $2D\theta_{FOV}/N$ (for $\theta_{FOV}$ in radians).

A wider FOV is desirable because it increases the tracking volume. For a given FOV, a larger working distance ($D$), is desirable for the same reason. However, as we increase FOV, $2\theta_{FOV}/N$ gets larger, i.e., our angular resolution gets coarser. Likewise, from basic geometry, our linear resolution gets coarser in proportion to $1/D$. In addition, recall that $N$ itself decreases as SNR decreases. Since the "signal" component of our SNR is the amount of light arriving at the sensor, SNR is proportional to $1/D^2$ – that is, increasing $D$ decreases the sensor's noise-limited resolution. The net effect is that spatial resolution is proportional to $1/D^3$.

Our strategy then, is to use the brightest emitters available to us, choose $D$ to maximize SNR (without saturating the sensors), and then calculate the focal length (or $\theta_{FOV}$) needed to achieve the requisite spatial resolution.

In our case, the sensor's noise floor is above our ADCs' noise floor—that is, ADC quantization noise is not a concern. Based upon a signal-to-noise ratio (SNR) of about 100dBmV at a working distance of 800mm, we have a theoretical resolution of about 16.6 bits[6]. In principle, this means that we can distinguish about 100,000 distinct locations along each of the sensor's axes.

For our tracker, a one arcminute rotation of the tracked target (yaw) translates to about a 10μm displacement of each emitter. Our linear resolution, then, must be 5μm or smaller at our nominal working distance.

We are using 50mm lenses; this gives us a usable FOV of about 9.14°. At 800mm, this translates to a linear resolution of about 2.5μm—roughly four times the "Nyquist rate" to resolve 10μm. This provides reasonable engineering margin, e.g., to allow for the fact that our effective resolution is probably below 16.6 bits.

### 3.4.2 Tracking Volume

The tracking volume is determined by three parameters: sensor FOV, inter-sensor distance, and angle between the sensors' optical axes. Given these three parameters, the tracking volume is the set of locations where all four emitters are visible to both sensors. It is a subset of the intersection of the sensors' FOVs. The tracking volume is also bounded by the distance between the sensors and the emitters to the extent that, beyond some distance, we will have insufficient spatial resolution (see above).

An angular FOV of 9.14° at a nominal working distance of 800mm corresponds to about a 128mm linear FOV. So, however we arrange the sensors, the width of the working volume will be about 128mm at a distance of 800mm. But how should the sensors be arranged? If we think about how emitter 3D positions are calculated—i.e., by triangulation from two 2D projections—it is clear that the emitters' distance from the sensors (its Z coordinate) comes from parallax. Increasing the parallax angle (determined by a combination of inter-sensor distance and angle) will improve Z precision. At the same time, increasing the parallax angle decreases the range of (pose) angles at which all emitters will be visible to both sensors and also decreases the depth of the working volume.

Given our narrow FOV and desired working distance, our chosen geometry, visible at the upper left of Figure 4, spaces the sensors 150mm apart with a parallax angle of 9° (each sensor's toe angle is 4.5°). In this configuration, the volume described by the intersections of the sensors' FOVs is approximately a rectangular prism beginning at about 700mm. This configuration comfortably accommodates the full range of head movement of a seated human and allows modest lateral translation (plus or minus ~30mm in X and Y). The net volume is approximately 120×120×250mm at a working distance of 700-950mm (800mm typical).

The orientation of the sensors with respect to the user's head was selected to best-match the tracker's per-axis sensitivity and accuracy with our understanding of human perceptual sensitivity and the possible user movements. The tracked target is located on top of the user's head and the sensors face the target. The user is oriented such that, in a neutral pose, the user is looking along the tracker's X-axis (cf. Fig. 2, bottom image). From Wallach's work [29], we know that sensitivity to rotational disparity is greatest in head rotation (yaw), second-greatest in nod (pitch) and least in tilt (roll). The user's orientation is such that head rotation is parallel to the tracker's X-Y plane and, thus, is least-sensitive to errors in Z. A seated person can translate his head in X and Y (forward and back, left and right) fairly easily but, due to physiology, one cannot move one's head up and down by any significant amount; the user's longitudinal axis is parallel to the tracker's Z axis, which happens to be its least accurate linear axis.

### 3.4.3 Tracking Instrument: Conclusion

Our tracking instrument is designed to maximize sample rate and minimize MTPL, while achieving best possible accuracy and repeatability. We have discussed the design decisions and many of the insights that led to the present implementation. We know that the pose sample rate is 50 kHz and have estimated MTPL at about 28μs. The next step is verifying MTPL by actually measuring it.

## 4 MEASUREMENT METHOD

In the previous section we analysed our tracker's latency from first principles and our knowledge of its implementation and predicted a motion-to-pose latency (MTPL) of about 28μs. To externally verify the tracker's performance, we want to conduct experiments to measure the tracker's dynamic tracking error, MTPL, and repeatability. We begin by defining a general mathematical formulation of dynamic tracking error that we then use as the analytical basis for the novel measurement technique described thereafter.

### 4.1 Dynamic Tracking Error

We define tracking error as the difference between the pose sample on the tracker's output and the true pose (ground truth) of the tracked target. To a first-order approximation, tracking error can be divided into two components: static error and dynamic error. Static tracking error is tracking error when the tracked target is stationary; static tracking error arises, among other things, from analog and quantization noise, drift, and calibration errors. Dynamic tracking error is tracking error when the tracked target is in motion. Fundamentally, dynamic tracking error arises due to the change in the target's pose between the time at which pose is sensed and the time when pose appears on the tracker's output—i.e., MTPL.

We now explain the relationship between dynamic tracking error, pose velocity, and MTPL. Let axis $A$ be one of the six pose axes (roll, pitch, yaw, $X$, $Y$, or $Z$). Let $Err_A$ denote the tracking error with respect to axis $A$. Let $V_A$ denote the pose velocity with respect to axis $A$; $V_A$ has units of rotational or linear velocity (e.g., degrees per second or meters per second, respectively). Let $T_S$ denote the pose sample period (units of time). Let $L_{TRACK}$ denote the tracking latency,

[5] $\theta_{FOV}=2\cdot\mathrm{atan}(f/r)$ where $f$ is the focal length and $r$ is half the sensor width.

[6] $20\log_{10}2 \approx 6.02$. $100/6.02 = 16.611$

i.e., the time from acquisition to pose output (see 3.3). Finally, let $N(V_A,t)$ denote the sum of all other error sources, including noise and tracking errors due to pose velocity $V_A$ at time t (e.g., motion blur).

Then $Err_A$ is bounded from below as follows:

$$Err_A \geq V_A \cdot (T_S + L_{TRACK}) + (V_A, t) \quad (1)$$

The $(T_S+L_{TRACK})$ term is exactly the MTPL. Equation (1) follows intuitively: $V_A \cdot (T_S+L_{TRACK})$ is the *distance* that the tracked target will have moved over the MTPL. The *N()* term allows us to account for other error sources such as noise. It should be clear that $Err_A$ is increasing in pose velocity, pose sample period, and tracking latency. This is why dynamic tracking error should always be stated with respect to pose velocity.

For completeness, let us consider the real-virtual displacement with respect to axis *A* due to motion-to-photon latency, $ErrRV_A$. This requires the addition of one additional term to Equation (1): display latency, denoted as $L_{DISP}$. Then we have:

$$ErrRV_A \propto V_A \cdot (T_S + L_{TRACK} + L_{DISP}) + N(V_A, t) \; (2)$$

The $(T_S + L_{TRACK} + L_{DISP})$ term is exactly the motion-to-photon latency. We use the proportional-to operator ($\propto$) because the displacement itself will depend on, e.g., the position of the virtual object with respect to *A* and the geometry of the OST AR display.

## 4.2 Measurement Principle of Operation

Referring to Equations (1) and (2), we can rewrite the relationship such that we have a lower bound on MTPL:

$$\frac{Err_A}{V_A} \geq \frac{Err_A - N(V_A,t)}{V_A} \geq (T_S + L_{TRACK}) \quad (3a)$$

$$\frac{ErrRV_A}{V_A} \geq \frac{ErrRV_A - N(V_A,t)}{V_A} \propto (T_S + L_{TRACK} + L_{DISP}) \quad (3b)$$

That is, if we can measure the dynamic tracking error, or the real-virtual displacement and the pose velocity, we can calculate MTPL or motion-to-photon latency, respectively. While this measurement could be performed with high-precision, high-speed instrumentation, we developed, and describe in the following sections, a novel technique for directly measuring $ErrRV_A$ and $V_A$ using three inexpensive consumer grade laser pointers.

In the calculations that follow, we ignore the contribution of $N(V_A,t)$. By doing so, at worst we end up computing a more pessimistic upper bound on latency, as can be seen in the inequalities above.

### 4.2.1 Displacement Measurement

Equation (2) tells us that real-virtual displacement is increasing in pose velocity – in magnitude and sign. That is, the displacement is in the direction of travel. With this in mind, our technique for measuring displacement is as follows.

A laser pointer is rigidly attached to the tracked target. All of the laser pointers used in these experiments emit visible light and are of the Class 3R/Class IIIa type (under 5mW). The laser pointer was modified so that it can be switched on and off using a logic-level signal; we generate that signal in the tracker's software. The position and orientation of the laser with respect to the tracked target are known. A software subroutine, executed after each pose output, calculates the intersection of the laser with a particular plane. If the point of intersection is within a set distance from a line on the plane, the laser is turned on; otherwise, it is turned off. In the experiments that follow, the plane is defined to be the ceiling of our lab (1650mm above the tracker's X-Z plane) and the line is parallel to the Z-axis (i.e., parallel to the mean optical axes of the sensors). The tracker is configured to turn the laser on whenever the point of intersection is computed as falling within 500μm of one side of the line; the set of such points is the intersection region.

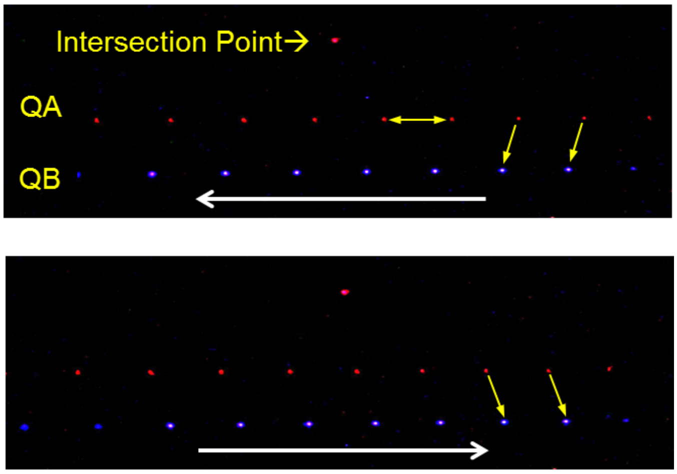

Fig. 6: Velocity measurement, quadrature example. The phase of QB is 90 degrees behind QA. The alignment of the red and blue dots tells us the direction of travel. The dot spacing of QA is used to calculate velocity. (See also Figure 1, Right.)

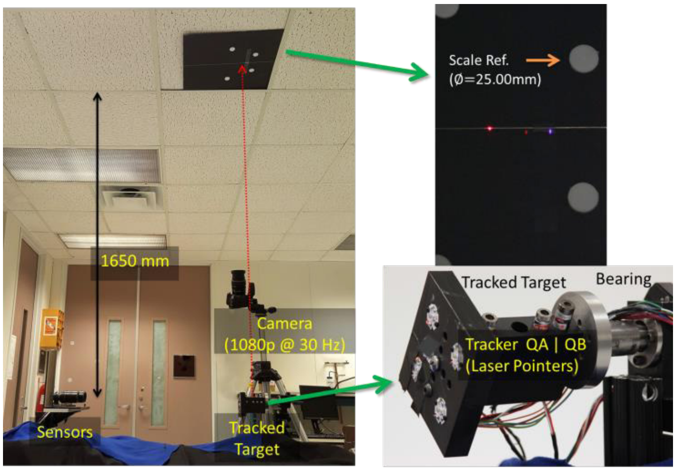

Figure 7: Experimental setup for latency and repeatability measurement. Left image: bottom left—sensor head; right, bottom to top—tracked target, position of user's head; video camera; black panel with scale markers. Top right image: view of panel with all three lasers active. Bottom right image: tracked target with attached lasers mounted on the bearing assembly.

Now, when the target is rotated about the Z-axis (yaw with respect to the user's head—the most perceptually-sensitive pose axis [29]), the laser's point-of-aim sweeps a path across the ceiling that is perpendicular to the intersection region. Laser dots appear whenever the tracker calculates that, based on the current pose, the laser intersects the ceiling within the intersection region. As discussed above, tracking latency will cause these dots to shift in the direction of rotation, i.e., to one side or the other of the intersection region, depending on the direction of travel.

The laser dots tell us the true pose of the tracked target at the time the calculated pose placed the laser in the intersection region. The separation between the dots produced when rotating in either direction corresponds to the real-virtual displacement that would be seen by the user during head rotation, i.e., $ErrRV_A$.

The experiment is recorded using a 30Hz video camera (Canon™ 7D) zoomed in on the intersection region. The locations of the laser dots are then measured by straightforward post-processing of the video (described in 4.3). In-frame scale markers let us convert from pixel dimensions to linear dimensions.

#### 4.2.2 Velocity Measurement

We wish to measure the velocity of the tracked target. Velocity is a vector, having both magnitude and direction. We need to know the pose velocity for each laser dot observed in the scheme described above. Our method lets us record velocity at the same time as recording of the laser dots, i.e., on the same video. The method is as follows:

Two additional laser pointers, QA and QB, are attached to the tracked target such that their beams are co-planar with and parallel to each other and to the tracking laser; their positions are offset 50mm and 75 mm aft of the tracking laser, respectively. We were able to obtain a blue/violet laser with a high enough switching speed to serve as QB; the color difference made post-processing easier.

An FPGA independent of the tracker and on a completely separate board controls QA and QB. QA is pulsed at 400Hz with a pulse width of 20μs. QB is pulsed at the same frequency and pulse width but 90° out-of-phase with respect to QA. This phase alignment is known as "quadrature" and allows us to calculate the magnitude of the pose velocity (which is proportional to the linear spacing between pulses) and the direction of travel (whether QA leads or lags QB). Examples with positive and negative velocity are shown in Figure 6.

The quadrature signals were extracted from video frames at the same time as the tracker dots (frames without a tracker dot or sufficiently many quadrature dots were discarded). The quadrature extraction algorithm requires sighting of at least two dots of QA and two dots of QB with one QB dot within one pulse period of a QA dot[7].

### 4.3 Data Processing

The videos from each experiment were processed using a combination of *ffmpeg* (www.ffmpeg.org) and Matlab™. The experiments were conducted with the room lights off. In the first step, *ffmpeg* was used to perform enough color space compression to eliminate sensor noise and remove duplicate (blank) frames. After this, the filtered videos were processed in Matlab™. The laser spots were located using the *imfindcircles* function.

Scale markers (white discs, visible on the upper right of Figure 7) were positioned so they were in-frame. The diameters of these discs are 25mm ±10μm. Calibration videos were taken with the lights on before and after our experiments to verify that the camera was not moved during the experiments. The mean diameter of these discs was used to calculate the conversion factor from pixels to mm. For the data reported here, the spatial resolution was approximately 0.135mm/pixel. No other camera calibration was applied[8].

## 5 Experimental Results

A series of experiments were run with the tracking instrument to measure tracking latency and repeatability. The method described in Section 4 was used to measure $ErrRV_A$, from which tracking latency ($L_{TRACK}$) is calculated.

### 5.1 Experimental Setup

A wide-angle photograph of the experimental setup is shown on the left side of Figure 7. Referring to this photograph we see: the sensors (bottom left), the tracked target (bottom right), the video camera filming the intersection region (right), and a black panel on the ceiling (top). The intersection region is located approximately along the center of the black panel and runs laterally (left-to-right); the barely visible thin yellow line is for reference purposes; it does not denote the intersection region. Note that in operation with a human subject, the sensors would be mounted looking down at the top an upright user's head. The setup shown, with the sensors and tracked target located laterally on an optical bench, is more convenient and precise for the experiments.

The tracked target and the attached lasers are shown on the lower right of Figure 7. The target itself is mounted to a bearing; the center of the target is offset by about 25mm from the rotational axis of the bearing, i.e., the target is rotating and translating. This configuration was used for the measurements discussed here because it allows consistent, reproducible, high-velocity rotations (yaw), in both directions, while keeping the axis of rotation parallel to the intersection region. The bearing makes it possible to maintain the parallel alignment even though the target was rotated by hand during the experiments.

The tracked target is located approximately 800mm from the mean focal point of the sensors. The distance from the sensors' mean optical axis to the ceiling is 1650mm.

### 5.2 Experiments

Each experiment was conducted with the room lights off; this was solely to simplify video processing as the tracker is not affected by visible light. For each experimental run, video recording was initiated and then the tracked target was rotated alternately in each direction. We were able to judge the approximate velocity from the spacing of the timing/quadrature spots. The mean velocities for all runs were about 500-550°/second.

For the purposes of measuring latency, we added a feature to the tracker's software so that we can artificially add a configurable amount of latency (in multiples of the sample period). This feature is implemented using a software FIFO. In addition to multiple runs with no added latency, we collected data with 20, 40, 80, 100, and 200μs of added latency. As we will see below, this gives us a second method for calculating MTPL while, still using the same measurement technique.

### 5.3 Results

In our first analysis, we examine the samples taken with no added latency. A visualization of the data is shown in Figure 8. The data points in this figure include only samples whose velocities are 300°/sec or greater; there are a total of 1,036 data points. The axes are in units of arcminutes. The horizontal axis is the X-position of the observed laser spot; recall that the tracker's X-axis is approximately parallel to the plane of rotation of the tracked target in this experiment. The vertical yellow rectangle represents the intersection region. Recall that this region is 500μm wide; this translates to about one arcminute at 1,650mm. The centroids of the laser spots are shown as colored triangles and squares, corresponding to positive and negative velocities, respectively. The centroids are colored according to the magnitude of the velocity; the scale is shown on the far right. Note the strong correlation between velocity and displacement (relative to x=0). The means of the centroids for positive-velocity samples and the negative-velocity samples are shown as green circles. The magenta and blue curves are the convex hulls of all the positive- and negative-velocity laser spots, respectively. We include these in the figure to give the reader

[7] Our video camera operates in rolling shutter mode and the per-frame exposure is only about 80% of a frame time. At the rotational velocities in these experiments, it was not uncommon for some dots to be missed; as stated, these frames were detected and discarded.

[8] The video was filmed using a 135mm lens. Radial distortion was assessed and was found to be very small and basically zero near the image center.

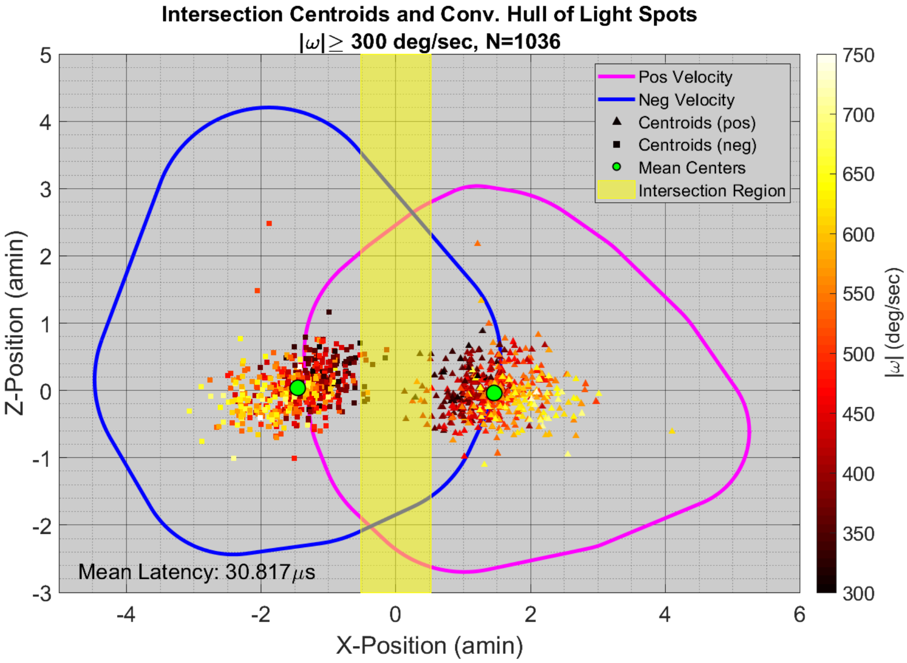

Fig. 8: Experimental results: Spot locations for positive (triangles) and negative (squares) velocities; spot color denotes abs(velocity) (scale on right). Magenta and cyan curves denote convex hull of the full extent of all laser spots. Laser spots have radii of about 1-2mm. Yellow strip denotes intersection region (500μm/1 arcminute in width).

a sense of how very small the displacement is—on the order of the radius of a spot.

The motion-to-photon latency was calculated using Equation (3b) where $ErrRV_A$ is the mean distance between spot centroids and the edges of the yellow strip. In this case, the value calculated is 30.817µs. Recall that this is motion-to-photon latency. This will be dissected further below.

In our second analysis, we considered the measured displacements and calculated motion-to-photon latencies for the samples where we artificially added tracking latency. Figure 9 shows a plot of measured latency vs. artificially added latency. Each data point is the mean of at least 200 measured latencies, again, using Equation (3b).

We have plotted a linear regression line against these data, the equation for which is shown in the legend. Note that the slope is almost exactly unity (which is as expected). The y-intercept, i.e., the predicted measured latency when the added latency is zero is 29.722µs. Again, this is motion-to-photon latency.

This second analysis is valuable because the displacement due to dynamic tracking error becomes more and more profound as artificial latency is added; this effectively increases the SNR of our measurement (since the displacements we're measuring are larger). The almost perfect linearity over so many measurements gives us confidence that the intercept is a good estimate of the motion-to-photon latency.

Our motion-to-photon latency estimates are 30.817µs and 29.772µs. We have higher measurement confidence in the latter number (as discussed above), so we will take that figure, rounded up to 30µs.

According to Equation (3b), we thus have

$$(T_S + L_{TRACK} + L_{DISP}) = 30\mu s.$$

$T_S$, the pose sample period, is known and is exactly 20µs. $L_{DISP}$, is the time from a new pose appearing on the tracker's output to the laser turning on. Via code instrumentation, we measured the software portion of $L_{DISP}$ to be approximately 1.5µs. The latency between the software asserting "laser on" and the laser actually turning on is not known precisely (we do not have a photosensor of sufficient bandwidth to measure this quantity directly). Based on a number of factors, however, we believe this latency to be no more than 1µs. So we estimate $L_{DISP}$ to be about 2.5µs. This gives us $L_{TRACK}$=7.5µs, which is very close to what we estimated in Section 3.3.

Finally, we can state that the measured motion-to-pose latency (MTPL), ($T_S$+$L_{TRACK}$), is approximately 27.5µs. Allowing for the discrepancy between our two motion-to-photon measurements and in the interest of being slightly conservative, we rounded this figure to 28µs.

In terms of repeatability, refer again to Figure 8. Even though the

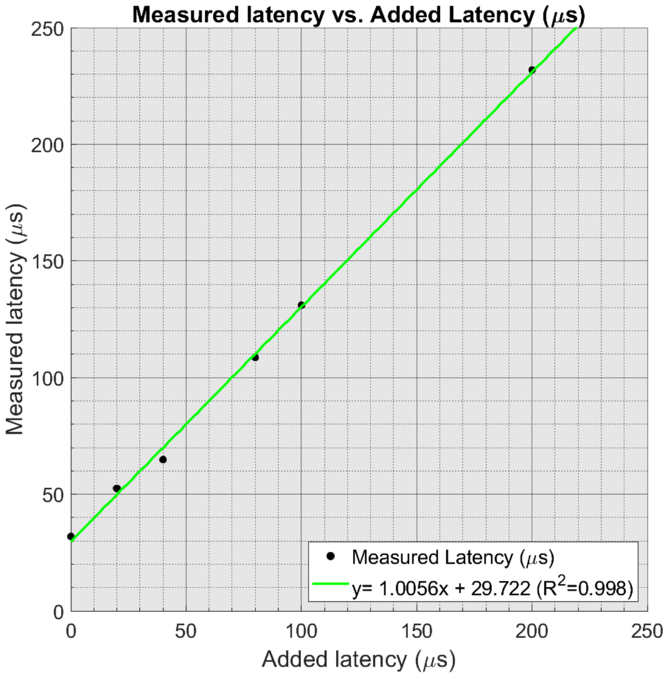

Fig. 9: Experimental results: regression analysis of measured latency vs. added latency. The y-intercept (29.772 µs) is the motion-to-photon latency with no added

velocities shown here are between 300 and 800°/second, the width of each point cloud, visually, is less than two arcminutes (0.96mm). Recall that the intersection region is one arcminute wide; this dispersion is strongly-correlated with pose velocity. If other error sources affecting repeatability were at play, the point clouds would not be this tight; indeed, if we consider just those points whose velocities are less than or equal to the mean (around 500°/second) and compensate for the latency-induced dynamic error, nearly all of the points fall within the intersection region. We are well within our repeatability objective of one arcminute (in yaw).

## 6 Future Work

Our tracking instrument demonstrates an effective approach for implementing the tracking component of a sub-100μs motion-to-photon latency OST AR HMD. In its present form, this instrument can be used for perceptual studies, e.g., into the perceptual thresholds for real-virtual displacement during head rotation.

That said, there are a number of immediately-obvious tracks of future work—both in terms of improving and extending the instrument itself and in terms of applications for the instrument or its descendants.

### Extended Tracking Volume

As discussed at the outset of this paper, our instrument's tracking volume, and particularly its 120×120mm X-Y cross-sectional area, make it unsuitable for use in the majority of AR applications. There are a number of ways in which the tracking volume can be extended,:

**Additional Sensors:** The extent of the instrument's tracking volume can be increased by adding more sensors in an appropriate geometry. Emitters' 3D positions are calculated by solving an over-defined system of linear equations by least squares; each sensor contributes two equations to the system. By its nature, the calculation scales to more than two and its accuracy should improve in cases where three or more sensor sightings of the same emitter are available. In the present geometry, each additional sensor added along the X-axis would increase the tracking volume by about 120mm in width (and in similar manner in Y).

We have considered the consequences of additional sensors in terms of, e.g., accuracy. An obvious concern is transitions from sensor pair (A,B) to pair (B,C), i.e., as the target moves laterally in X. We have reason to believe that errors of this form, if non-trivial, would be straightforward to address. First, the transition would not be discontinuous—in the sense that the target will be in view of all three sensors for some non-trivial time. This affords the opportunity to smooth the transition, for example, by calculating the weighted mean of the emitter positions given by (A,B), (A,B,C) and (B,C), where the weighting shifts according the sensor coordinates from each sensor. Additionally, we can take advantage of the combination of physiological and perceptual phenomena to filter at least the positional pose components. According to [29], our sensitivity to real-virtual displacement is at least an order-of-magnitude smaller in head/body translation compared to head rotation. Secondly, translational velocities and accelerations are minuscule compared to those of rotation [30]. Taking these observations into consideration, it stands to reason that filtering of position, at the expense of added latency (in position) would be imperceptible; and, by Equation (2), the reduced velocity, on its own, means that the dynamic error due to latency will also be small.

**Working Distance:** Lateral FOV increases linearly with distance; sensor spacing and angles could be adjusted to accommodate a larger and wider working volume at a greater distance. The cost, however, is reduced SNR. In the present system, moving from a distance of 800 to 1,200mm results in a loss of about 6dB of SNR—about one bit of spatial resolution. The spatial resolution will also be reduced by about 25% because the FOV covers a wider distance—amounting to another half a bit of resolution. The system would likely still be usable, but at the cost of diminished accuracy.

The effects of reduced SNR can be mitigated by improving noise rejection. Specifically, through trivial configuration changes, we can increase the number of sensor samples taken per pose sample period; since our digitizers can digitize at a rate of 1MHz, each additional sample costs only one microsecond. Decimation by a factor of four gives the equivalent of 6dB (one bit) of noise rejection; the cost to the instrument would be slightly higher latency (due to the additional digitizer sample periods and additional computation). Depending on how weak the signal becomes, it may be prudent to increase the gain of the transimpedance amplifiers in the sensor's analog front-end; note that this would not increase noise gain. This would ensure that one continues to use all of the ADCs' dynamic range.

**Focal Length/FOV**: Perhaps the simplest and most obvious way to increase tracking volume is to use wider-angle lenses. Again, this would result in a loss of spatial resolution; but, unlike increasing working distance, SNR would remain essentially the same. We estimate (but have not verified) that the present instrument, which has some excess spatial resolution, would see negligible performance degradation with 35mm lenses (vs. 50mm) and would most probably still have acceptable accuracy with 28mm lenses (though radial distortion correction may become necessary). With 28mm optics, the lateral FOV would be 228mm at an 800mm working distance and 257mm at a 900mm working distance. These sorts of changes, perhaps combined with additional sensors, would increase the working volume to comfortably accommodate a moderately active seated person, such as a person working at a desk.

### Perceptual Studies and Integration with Low-Latency OST AR HMDs

In our view, the most exciting use of our instrument is in user studies. A characterization of perceptual tolerances and thresholds for real-virtual displacement during head rotation and other movements would be of immediate benefit in terms of defining lower bounds for the performance of future OST AR HMDs. Among other things, once we know these parameters, we can approach the design of trackers with definite requirements for latency, spatial resolution, accuracy, etc.

Further experiments could likely be conducted using, for example a laser as the equivalent of a single-pixel display. Coupling our instrument or a descendant thereof with a low-latency OST AR head-mounted display would enable a much richer range of additional experiments. Finally, we hope that trackers similar to the present instrument will be used for research into and development of new low-latency displays and related technologies.

## 7 Conclusion

We have presented a head tracking instrument with motion-to-pose latency of 28μs and a pose sample rate of 50 kHz. The instrument is capable of maintaining a dynamic tracking error of less than one arcminute at yaw rates of over 500°/second. Motion-to-pose latency was measured using a novel laser-pointer-based technique. The performance of this instrument exceeds that of any previously disclosed non-mechanical head tracking device by at least a factor of twenty. The small tracking volume limits the instrument's applicability to certain specialized OST AR use cases involving a stationary seated person, such as in aviation; however, the instrument is immediately useful in human perceptual research and other research requiring high-frequency and/or low-latency motion tracking.

## 8 Acknowledgements

The authors would like to thank Jim Mahaney for his engineering support and advice. We are also grateful to the anonymous reviewers for their constructive and insightful feedback and suggestions for improving this paper.

## References

[1] P. Lincoln, A. Blate, M. Singh, T. Whitted, A. Lastra, H. Fuchs, et al., "From motion to photons in 80 microseconds: Towards minimal latency for virtual and augmented reality," IEEE Transactions on Visualization & Computer Graphics, pp. 1367-1376, 2016.

[2] P. Lincoln, A. Blate, M. Singh, A. State, M. C. Whitton, T. Whitted and H. Fuchs, "Scene-adaptive High Dynamic Range Display for Low Latency Augmented Reality," in Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, New York, NY, USA, 2017.

[3] M. Regan and G. S. P. Miller, "The Problem of Persistence with Rotating Displays," IEEE Transactions on Visualization and Computer Graphics, vol. 23, pp. 1295-1301, 4 2017.

[4] A. Vorozcovs, W. Stürzlinger, A. Hogue and R. S. Allison, "The hedgehog: a novel optical tracking method for spatially immersive displays," Presence: Teleoperators & Virtual Environments, vol. 15, pp. 108-121, 2006.

[5] R. D. Wiersma, S. L. Tomarken, Z. Grelewicz, A. H. Belcher and H. Kang, "Spatial and temporal performance of 3D optical surface imaging for real-time head position tracking," Medical physics, vol. 40, 2013.

[6] C. Nafis, V. Jensen, L. Beauregard and P. Anderson, "Method for estimating dynamic EM tracking accuracy of surgical navigation tools," in Medical Imaging 2006: Visualization, Image-Guided Procedures, and Display, 2006.

[7] D. Miller and G. Bishop, "Latency meter: a device end-to-end latency of VE systems," in Stereoscopic Displays and Virtual Reality Systems IX, 2002.

[8] G. Welch, G. Bishop, L. Vicci, S. Brumback, K. Keller, et al., "The HiBall tracker: High-performance wide-area tracking for virtual and augmented environments," in Proceedings of the ACM symposium on Virtual reality software and technology, 1999.

[9] M. Bauer, "Tracking Errors in Augmented Reality," 2007.

[10] M. Meehan, S. Razzaque, M. C. Whitton and F. P. Brooks Jr, "Effect of latency on presence in stressful virtual environments," in Proceedings of the IEEE Virtual Reality 2003 (VR'03, 2003.

[11] J. J. Jerald, "Scene-motion-and latency-perception thresholds for head-mounted displays," 2009.

[12] M. Nabiyouni, S. Scerbo, D. A. Bowman and T. Höllerer, "Relative Effects of Real-world and Virtual-World Latency on an Augmented Reality Training Task: An AR Simulation Experiment," Frontiers in ICT, vol. 3, p. 34, 2017.

[13] W. Wang and I. J. Busch-Vishniac, "A method for measurement of multiple light spot positions on one position-sensitive detector (PSD)," IEEE Transactions on Instrumentation and Measurement, vol. 42, pp. 14-18, 1993.

[14] J.-f. Wang, V. Chi and H. Fuchs, "A Real-time Optical 3D Tracker for Head-mounted Display Systems," SIGGRAPH Comput. Graph., vol. 24, pp. 205-215, 1990.

[15] R. A. MacLachlan and R. N. Cameron, "High-Speed Microscale Optical Tracking Using Digital Frequency-Domain Multiplexing," in IEEE transactions on instrumentation and measurement, 2009.

[16] FirstSensor, PSD Series Data Sheet; Dual Axis PSD with Sum and Difference Amplifier, 2013.

[17] A. Bapat, E. Dunn and J. M. Frahm, "Towards Kilo-Hertz 6- DoF Visual Tracking Using an Egocentric Cluster of Rolling Shutter Cameras," IEEE Transactions on Visualization and Computer Graphics, vol. 22, pp. 2358-2367, 2016.

[18] B. D. Allen, "Hardware Design Optimization for Human Motion Tracking Systems," Department of Computer Science, Chapel Hill, NC, USA, 2007.

[19] L. Davis, E. Clarkson and J. P. Rolland, "Predicting accuracy in pose estimation for marker-based tracking," in Mixed and Augmented Reality, 2003. Proceedings. The Second IEEE and ACM International Symposium on, 2003.

[20] M. Billeter, G. Röthlin, J. Wezel, D. Iwai and A. Grundhöfer, "A LED-Based IR/RGB End-to-End Latency Measurement Device," in 2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct), 2016.

[21] T. Sielhorst, W. Sa, A. Khamene, F. Sauer and N. Navab, "Measurement of absolute latency for video see through augmented reality," in 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007.

[22] S. Friston and A. Steed, "Measuring Latency in Virtual Environments," IEEE Transactions on Visualization and Computer Graphics, vol. 20, pp. 616-625, 4 2014.

[23] Y.-J. Tsai, Y.-X. Wang and M. Ouhyoung, "Affordable System for Measuring Motion-to-photon Latency of Virtual Reality in Mobile Devices," in SIGGRAPH Asia 2017 Posters, New York, NY, USA, 2017.

[24] D. Pustka, J. Willneff, O. Wenisch, P. Lükewille, K. Achatz, P. Keitler and G. Klinker, "Determining the point of minimum error for 6DOF pose uncertainty representation," in Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on, 2010.

[25] Q.-T. Luong and O. D. Faugeras, "Determining the fundamental matrix with planes: Instability and new algorithms," in Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on, 1993.

[26] Z. Zhang, R. Deriche, O. Faugeras and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," Artificial intelligence, vol. 78, pp. 87-119, 1995.

[27] R. I. Hartley, "In defense of the eight-point algorithm," IEEE Transactions on pattern analysis and machine intelligence, vol. 19, pp. 580-593, 1997.

[28] I. E. Sutherland, "Three-dimensional data input by tablet," Proceedings of the IEEE, vol. 62, pp. 453-461, 4 1974.

[29] H. Wallach, "Perceiving a stable environment when one moves," Annual review of psychology, vol. 38, pp. 1-29, 1987.

[30] W. Bussone, "Linear and angular head accelerations in daily life," 2005.

[31] L. A. Riggs, "Vision and Visual Perception," C. h. Graham, Ed., New York, New York, Wiley, 1965, pp. 321-349.

[32] C. E. Shannon, "Communication in the presence of noise," Proceedings of the IRE, vol. 37, pp. 10-21, 1949.

[33] Linear Technologies, “18-Bit, 1Msps, ±10.24V True Bipolar, Pseudo-Differential Input ADC with 95dB SNR”, 2014.

[34] LED Engin, "940nm Dual Junction Infrared LED Emitter," 2018.

[35] L. Hongfei, X. Ying and C. Zhong, "A High Precision Optical Position Detector Based on Duo-lateral PSD," 2009 International Forum on Computer Science-Technology and Applications, vol. 3, pp. 90-92, 2009.