Poster# 16

LATENCY MEASUREMENT OF A REAL-TIME VIRTUAL ACOUSTIC ENVIRONMENT RENDERING SYSTEM

Joel D. Miller   Mark R. Anderson

Elizabeth M. Wenzel   Bryan U. McClain

Spatial Auditory Displays Lab

NASA Ames Research Center

ABSTRACT

Techniques for measuring and estimating the end-to-end latency and component latencies of a virtual acoustic environment are discussed. These key parameters impact the responsiveness and, hence, “realism” of a virtual environment.

INTRODUCTION

Latency provides an important indicator of the dynamic performance of a virtual acoustic environment (VAE) and it is critical that it be carefully defined and measured. In a VAE, the end-to-end latency refers to the time elapsed from the transduction of an event or action, such as movement of the head, until the consequences of that action cause the equivalent change in the virtual environment. Latencies are contributed by individual components of the system, including tracking devices, signal processing algorithms, device drivers, and communication lines. Due to variability in the way these components interact, a system's end-to-end latency will vary over time. Thus, measurements of the mean, standard deviation, and range are needed to characterize this parameter.

Psychoacoustic data can provide guidelines regarding whether a given system's end-to-end latency meets perceptual requirements [1]. For example, examination of the head motions that listeners use to aid localization suggests that the angular velocity of some head motions (in particular, left-right yaw) may be as fast as 175°/s for short time periods (about 1s). From psychophysical studies of the minimum audible movement angle for real sound sources (listener position fixed), one can infer that the minimum perceptible end-to-end latency for a virtual audio system should be no more than about 70ms for a source velocity of 180°/s.

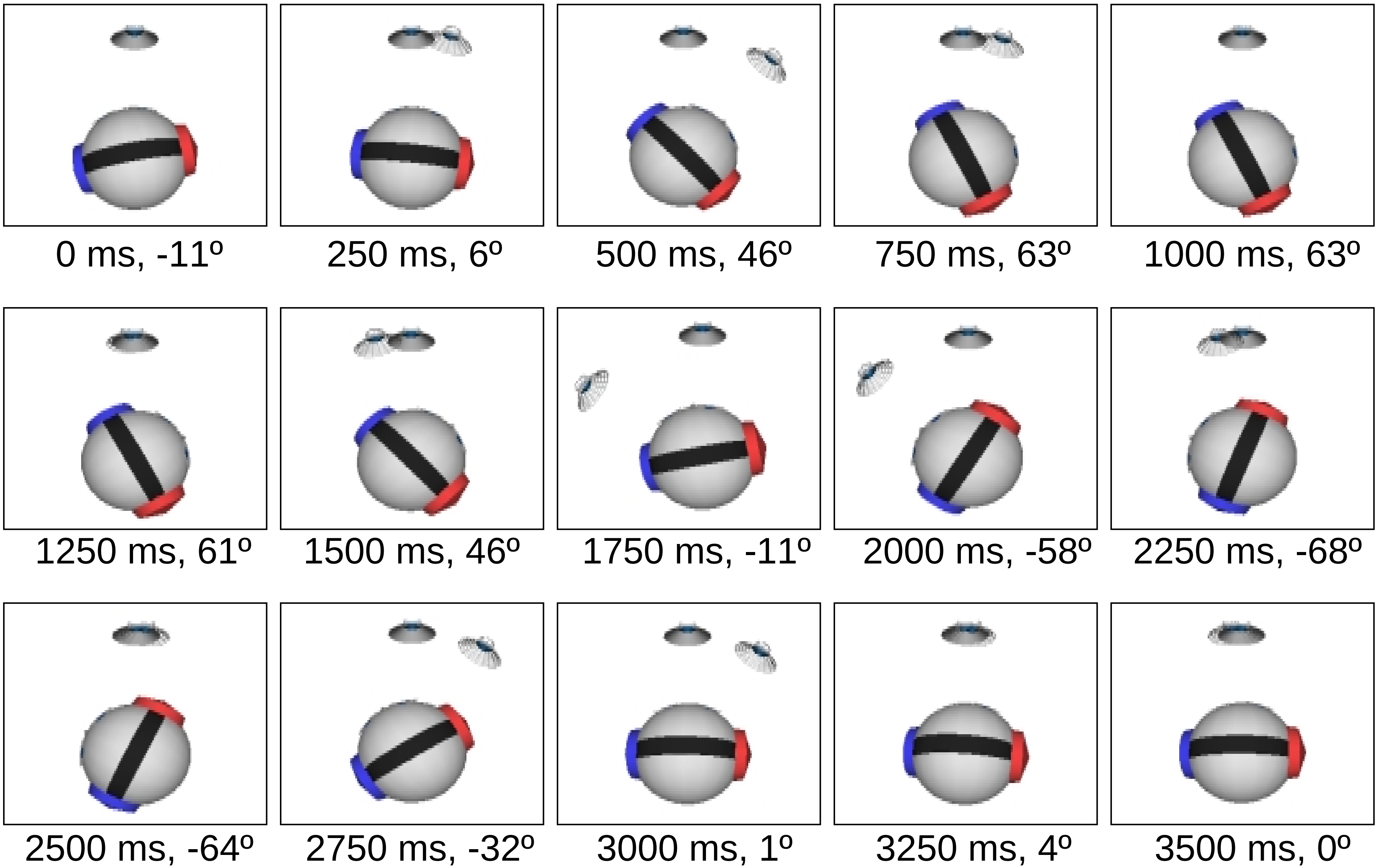
[1] E.M. Wenzel, “The role of system latency in multi-sensory virtual displays for space applications,” Proc. HCI Intl., New Orleans, LA, August 2001, pp. 619-623.

LATENCY SIMULATION ERROR

Latency can introduce positional errors in the virtual environment simulation. Figure 1 illustrates sound source movement error due to the latency between the tracking of a head-tracked listener and the headphone display. The solid speaker is the desired location of a stationary virtual sound source. The wireframe speaker represents the location of the rendered sound source displayed at the headphones.

In this example, the end-to-end latency of the rendering system is 250ms. As the listener yaws to the right, the rendered source moves with the listener due to the lag between the head tracker and headphone display. As listener motion slows (time indices 750ms thru 1250ms), the simulation catches-up. When the listener starts moving again, the rendered source begins to move as well. Thus, the rendered source fails to remain stationary and hovers about the desired sound source location.

Figure 1 - Source Movement Error Due to Latency (Latency = 250ms)



LATENCY MEASUREMENT TOOLS

When measuring latency, one needs an accurate way to measure the interval of time between two events. These events can be inside or outside of a computer. VAE end-to-end latency is an example of an interval between two external events where the first event is the user crossing a threshold and the second event is the user hearing the rendered result of the threshold crossing. API (application programming interface) latency is an example of an interval between an internal event and an external event where the first event is the time at which an API function call is made and the second event is the user hearing the rendered result of the API call.

External Events:

To measure the interval between events outside of a computer, an interval counter or digital storage oscilloscope can be used to measure the time difference between rising edges of two electrical signals. In some cases, a transducer is required to convert the event of interest into an electrical signal (e.g., an optical switch to capture the time at which an object crosses a physical threshold).

Internal Events:

When measuring event intervals inside of a computer, time functions can be used to time stamp events. The difference between event time stamps then provides the interval value. Often, multiple time functions exist, so one must be careful to select the timer with the greatest accuracy and resolution.

In Microsoft Windows, the QueryPerformanceCounter() function provides an extremely accurate timer with resolution of a microsecond or better (depends on OS and CPU).

Mixed External and Internal Events:

To measure the interval between an event inside and an event outside of a computer, the internal event needs to be externalized. An internal event can be externalized by writing to the serial port or the parallel port when the internal event occurs. Of course, the latency of the port write must be determined and stable.

Serial port loopback tests under Microsoft Windows (98/ME/2000) demonstrated that the WriteFile() function can externalize an event in less than 0.5ms.

SLAB LATENCY COMPONENTS

To measure the latency of the SLAB VAE rendering system [2], two approaches were taken, a low-level individual latency component analysis, and a high-level user parameter analysis.

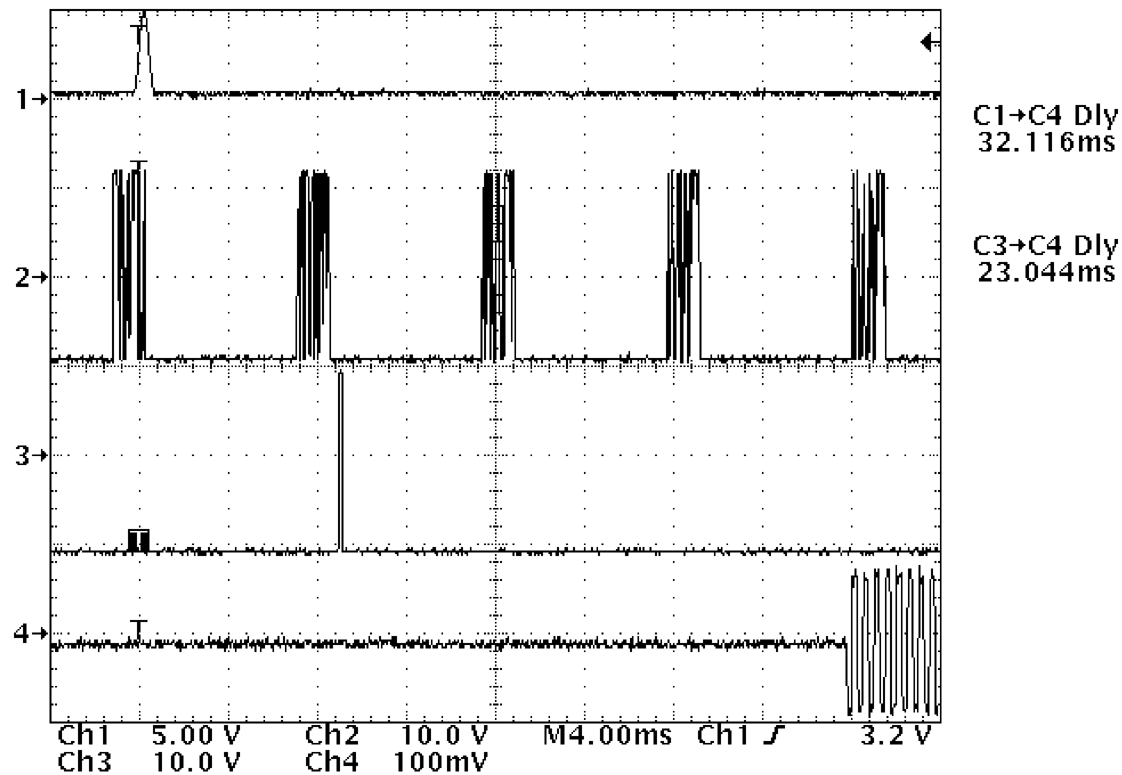
In the low-level approach, each contributing component was isolated and analyzed. In the high-level approach, end-to-end latency data was collected for several permutations of SLAB user parameters.

Swing-Arm Apparatus:

A swing-arm apparatus [3] was used to measure tracker latency and end-to-end latency. An electromagnetic Polhemus Fastrak tracker sensor is attached to a mechanical swing-arm. When the swing-arm is pushed, it passes through an optical switch (Figure 2, Ch1), triggering a single-shot oscilloscope capture of tracker serial output (Figure 2, Ch2), tracker library output (Figure 2, Ch3), and SLAB headphone output (Figure 2, Ch4).

[2] http://human-factors.arc.nasa.gov/SLAB
[3] B.D. Adelstein, E.R. Johnston, and S.R. Ellis, “Dynamic Response of Electromagnetic Spatial Displacement Trackers,” Presence, vol. 5, no. 3, pp. 302-318, 1996.

Figure 2 - Latency Measurement Oscilloscope Screenshot



Ch1: optical switch, Ch2: tracker serial communications, Ch3: tracker library serial port write, Ch4: SLAB headphone display. Output buffer size = 4096 bytes. Write buffer size = 256 bytes.

Tracker Latency:

Two components contribute to Fastrak tracker latency, tracker update rate and electromagnetic field sampling. At a 120Hz update rate, the tracker generates steady bursts of serial data at 8.3ms intervals (Figure 2, Ch2). Since the optical switch can be crossed at any time within this interval, a variable latency exists of **0.0-8.3ms** with a uniform probability of any given latency value occurring at any given time.

Attaching an electromagnetic pickup to the Fastrak source revealed that the tracker locates the sensor by sampling three electromagnetic bursts from the source. The interval from the middle of the bursts to the beginning of serial output was measured to be **3.5ms**.

Serial Communications Latency:

Tracker data packets of 170 bits are sent to the computer over a 115,200 bps (bits per second) serial line yielding a serial communications latency of **1.5ms**. This is consistent with the measured width of the serial data bursts in Figure 2, Ch2.

Tracker Driver Latency:

The tracker driver latency is the difference between the last serial bit read from the tracker (Figure 2, Ch2) and a serial port write placed after a blocked tracker driver read (Figure 2, Ch3). The tracker driver latency was measured to be **0.4-0.5ms**.

Output Buffer Latency:

Real-time audio processing systems process samples in groups of frames (aka blocks). For systems with a large frame size, frame size can contribute significant latency. SLAB’s frame size is only 32 samples and is absorbed within the output buffer management algorithm.

SLAB’s output buffer management algorithm has two parameters: output buffer size (OBS) and write buffer size (WBS). These two parameters introduce a uniformly distributed latency between **(OBS - WBS) and OBS** (in ms). The mean of this range is termed Estimated Buffer Latency (EBL) and provides a rough estimate of SLAB’s API latency. The latency jitter introduced by the output buffer management algorithm is equal to the write buffer size.

Sound Peripheral Driver Latency:

DirectSound can use one of two driver models, VxD (Win98/ME) or WDM (Win98/ME/2k). The WDM driver may include a component called the KMixer that can add up to 30ms of latency. Subtracting the Estimated Buffer Latency from SLAB’s measured API latency yielded **2.1ms** for a driver not using the KMixer and 26.3ms for a driver using the KMixer.

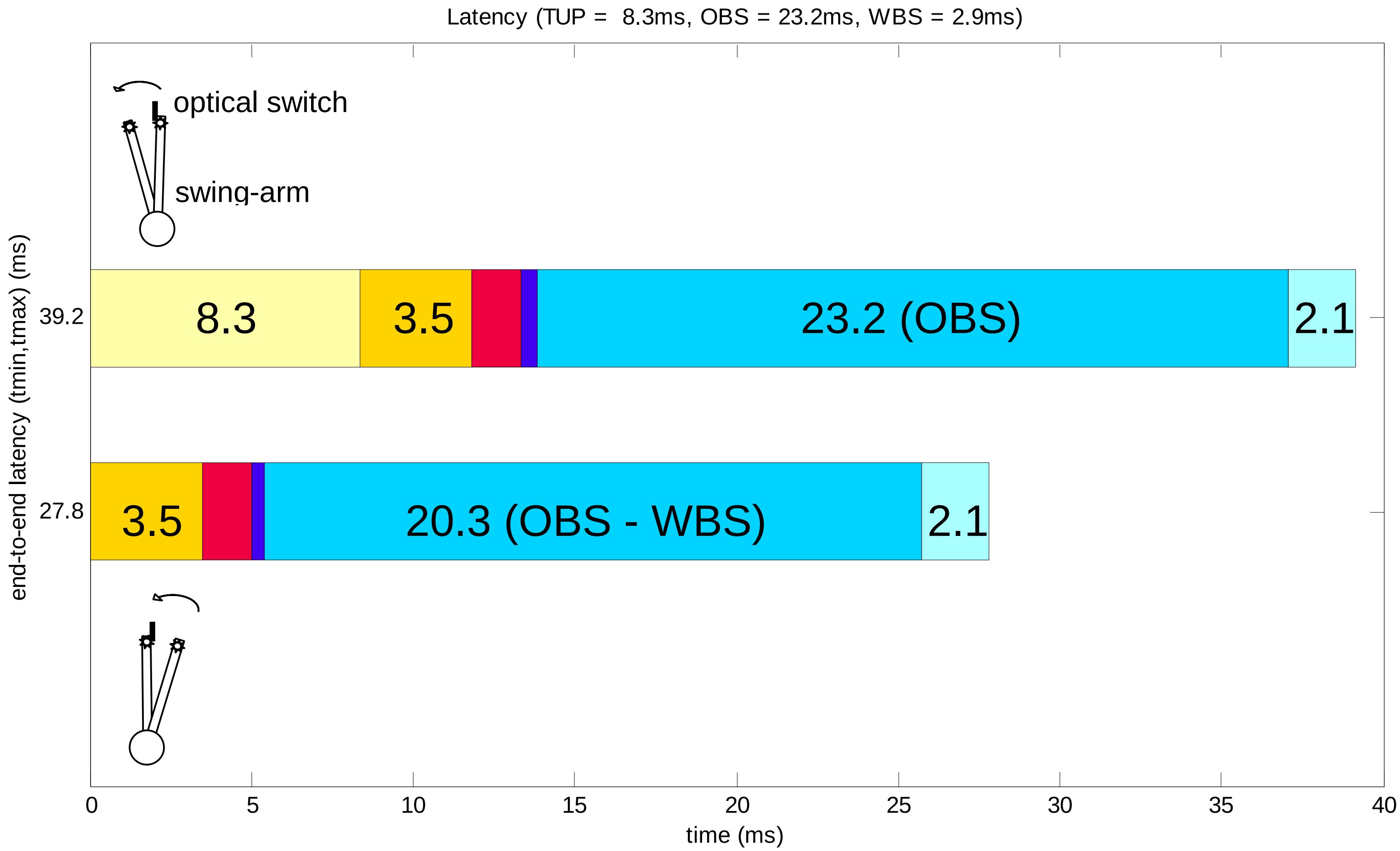
Estimated End-to-End Latency:

Summing the component latencies from the preceding sections (values in **bold type**) yields the following estimates for host-mode end-to-end latency (in ms, excluding the KMixer; Figure 3):

$$t_{min} = 0.0 + 3.5 + 1.5 + 0.4 + OBS - WBS + 2.1$$
$$t_{max} = \text{tracker update period} + 3.5 + 1.5 + 0.5 + OBS + 2.1$$
$$t_{avg} = (tmin + tmax) / 2$$
$$= 8.5 + \text{tracker update period}/2 + OBS - WBS/2$$

The latency values should be distributed in a trapezoidal or triangular distribution due to the cascade of the uniformly distributed tracker update rate and output buffer latencies (Figure 4). This distribution provides an estimate of the latency jitter in the system.

Figure 3 - Cascaded Component Latencies

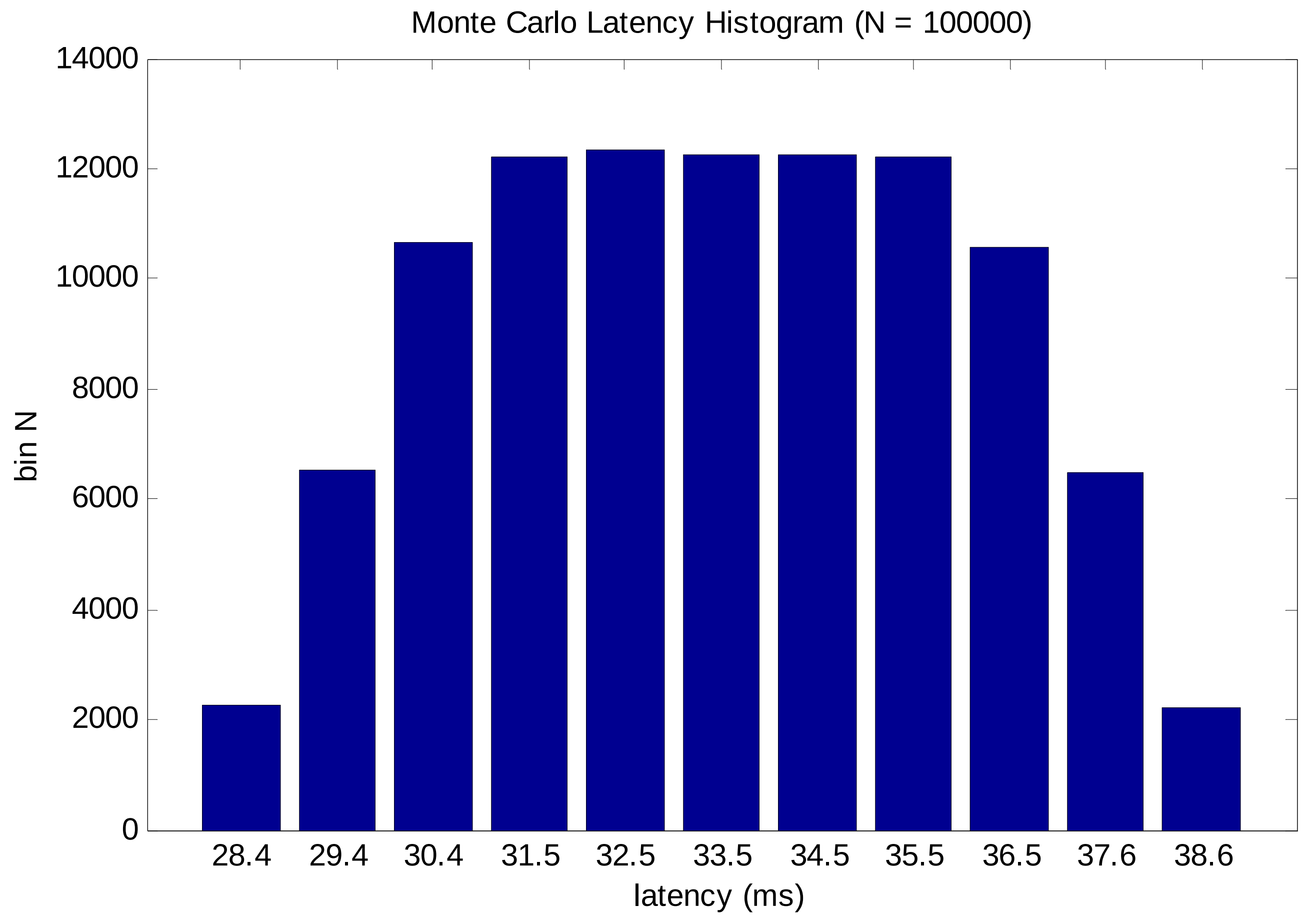


t_{min} and t_{max} end-to-end latency estimates. Tracker update period = 8.3ms (tracker update rate = 120Hz). Output buffer size = 23.3ms (4096 bytes). Write buffer size = 2.9ms (512 bytes). These estimates correspond to the t_{min} and t_{max} values plotted third from left in Figure 5.

Figure 4 - Latency Jitter Distribution

t_{min} = 27.8 ms
t_{max} = 39.2 ms
t_{avg} = 33.5 ms

mc_{min} = 27.9 ms
mc_{max} = 39.1 ms
mc_{avg} = 33.5 ms



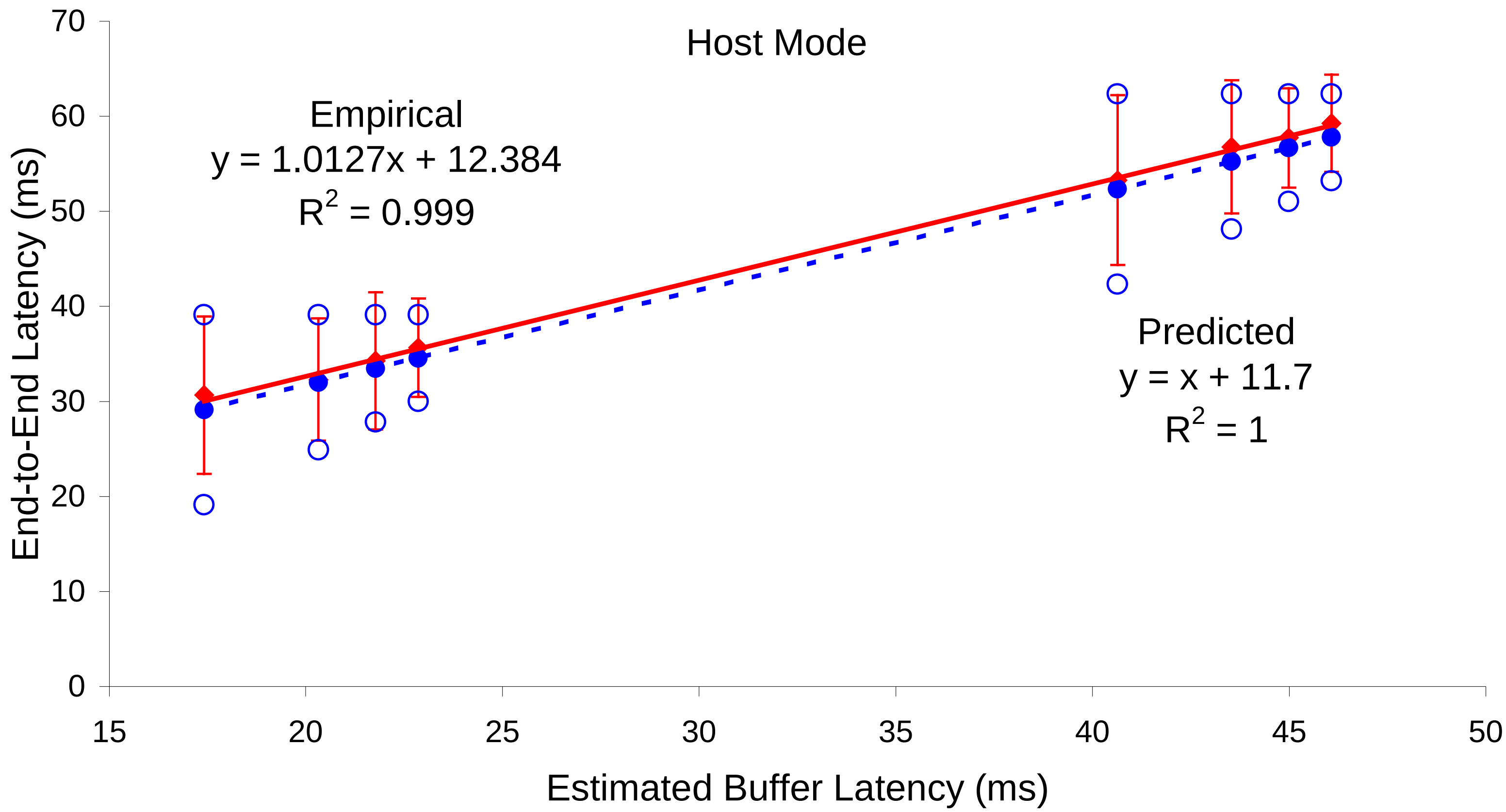
Latency distribution computed using Monte Carlo method. Tracker update period = 8.3ms (tracker update rate = 120Hz). Output buffer size = 23.3ms (4096 bytes). Write buffer size = 2.9ms (512 bytes). This distribution corresponds to the t_{min}, t_{max}, t_{avg} values plotted third from left in Figure 5.

END-TO-END LATENCY MEASUREMENTS

End-to-end latencies were measured using the swing-arm apparatus described earlier with a tracker update rate of 120Hz. Twenty-five latency measurements were taken for different combinations of the SLAB API parameters: output buffer size (OBS) (4096 and 8192 bytes) and write buffer size (WBS) (128, 512, 1024 and 2048 bytes). The results are plotted as a function of Estimated Buffer Latency in Figure 5. The cluster of points on the left side of Figure 5 refers to an OBS of 4096 while the cluster on the right refers to an OBS of 8192. For data points within these clusters, the WBS values from left-to-right are: 2048, 1024, 512 and 128.

In general, the data show that the predicted and empirical values match within less than 1 ms. For a given tracker update rate, the pattern of the data indicates that the OBS largely impacts the mean latency, while the WBS affects the latency jitter (i.e., the range of possible latency values). Larger WBS values produce more jitter but a smaller mean latency. Thus, there is a trade-off between jitter and latency for decreasing values of the WBS. It should be remembered that the tracker update rate has a significant impact on both mean latency and jitter. For example, if the tracker update rate is reduced, mean latency would increase by a fixed amount (i.e., all data points would shift upward) and the overall latency jitter would also increase.

Figure 5 - Predicted and Measured End-to-End Latency



Circles: predicted minimum and maximum latency (t_{min}, t_{max}). Points: predicted average latency (t_{avg}). Diamonds: mean measured latency. Error bars: ±2 standard deviations for the empirical data. Linear best fits for the mean predicted (dashed line) and empirical latencies (solid line) are indicated.

CONCLUSIONS

This poster describes a set of tools for performing high-precision latency measurement. Using these tools, formulas were derived to characterize and predict the end-to-end latency of SLAB, a real-time VAE rendering system. The accuracy of the formulae was verified by comparison to empirical data. Future work will include analyzing the effects of system load, parameter smoothing, and alternate buffer management techniques on latency and latency jitter.