Latency can be detrimental for the experience of Virtual Reality. High latency can lead to loss of performance and cybersickness. There are simple approaches to measure approximate latency and more elaborated for more insight into latency behavior. Yet there are still researchers who do not measure the latency of the system they are using to conduct VR experiments.

This paper provides an illustrated overview of different approaches to measure latency of VR applications, as well as a small decision-making guide to assist in the choice of the measurement method. The visual style offers a more approachable way to understand how to measure latency.
When talking about latency, we usually refer to **Motion To Photon Latency** or End To End Latency.

**Motion To Photon latency is the time** from starting a movement that is tracked, and fed into the simulation until the image corresponding to the movement is shown on screen.

**The tracked movement can originate from**
- a head mounted display (HMD),
- a rigid object,
- a motion controller,
- some other tracked object, e.g., a Vive tracker,
- or be completely synthetic.

**The screen may be**
- a computer monitor,
- a projection,
- an HMD, or similar.

For measurement, both the movement and its effect on screen are captured by either
- a camera that measures many brightness values but not that often,
- or a photodiode that measures only one brightness value but very often.
The most straightforward form of latency measurement is **Frame Counting**.

**Method 1**
He et al. [8] move a tracked wand in front of a screen.

- The screen shows the virtual counterpart.
- A camera records the scene.

The video is analysed to find:
- (a) when the wand overlaps a line on the screen.
- (b) when the virtual wand overlaps a line on the screen.

The time between the frames is the Motion To Photon latency.

**Method 2**
Similar, Wu et al. [24] use a moving bar. The position of the real and virtual bar can be extracted automatically via thresholding.

**Method 3**
Roberts et al. [18] compare the beginning of a movement while Miller et al. [12] use the end of a movement.

They use 1D CCD arrays to extract the centroid of the tracked user.

**Method 4**
Friston et al. [6], after detecting their tracked objects via thresholding the video, compare the peak of acceleration.
Many approaches use a **Pendulum**

**Method 1**

Steed [21] proposes **Sine Fitting**:

- He attaches a tracked object to a pendulum,
- and the position is rendered on a screen.

A camera records both, the real and the virtual pendulum.

The positions are extracted from the video via thresholding automatically:

He fits a sine curve (e.g. \( \cos(2\pi \times \text{frequency} \times \text{deflection} + \text{phase}) \)). The fit prevents inaccuracies of the thresholding, and low sampling frequency in the video.

**Method 2**

**phase difference**

The predictability of a pendulum is used in other approaches to measure latency. Mine et.al. [13] uses a pendulum and two photodiodes:

The motion to photon latency can be derived as

\[
\text{latency} = \frac{\text{phase}}{2\pi \times \text{frequency} \times \text{fps}}
\]
Other approaches

**Method 1**
Liang et al. [11] record a pendulum and the timestamp of the last tracking data.

**Method 2**
Chang et al. [3] rotate an HMD.
Bars drawn on the HMD make it easier to recognise motion.

**Method 3**
Pape et al. [15] use projection based VR.
The ground truth when the servo motor starts to move is provided with a switch.

**Method 4**
Seo et al. [19] use an elaborated platform to simulate real head movements.

Once the pendulum overlaps the reference

they look how much the tracking data deviated at the recorded timestamp.

The virtual scene mirrored on a monitor follows with a delay to the rotation.

The VR system changes the brightness of a projected dot once it receives the motion information.

Other approaches
Measuring latency with HMDs is difficult because the image of the screen is distorted by lenses which creates a scrambled image if the camera is not right in front of the lenses.

Method 1

Feldstein et al. [5] then push the HMD. pop out the lenses The scene follows the HMD movement with a delay.

Method 2

Kijima et al. [10] use two synchronised cameras. Upon rotation, the real one moves faster out of the image than the virtual counterpart which is used to determine the latency.

Method 3

Raaen et al. [16] use two photodiodes. One is lit by a laser pointer as long as the HMD is in rest position, while the other records the virtual counterpart. when the HMD is moved, the first photodiode is not shown on anymore. The screen brightness changes once the simulation register the movement.

One observes a real stick while the other records the virtual counterpart.
**Measuring latency continuously** is desirable to capture the time varying behaviour.

**Method 1**
- Di Luca et al. [4] move an HMD in front of a gradient, the simulation tries to show the same brightness on screen but always lags behind.
- One photodiode picks up the real gradient's brightness.
- Another photodiode picks up the screen brightness.
- They calculate the latency via cross correlation.
- Though, only calculating one value, this approach allows to correlate a reading of every frame on the screen.

**Method 2**
- Papadakis et al. [14] track the motion of a servo motor.
- The ground truth orientation is read with an encoder.
- Once the motor reaches a certain position, the simulation changes the screen colour.
- The brightness is read by a photodiode on screen.
- An oscilloscope compares the screen brightness and the motor orientation.

**Method 3**
- Becher et al. [1] encode the HMD rotation with multiple discrete values on the screen.
- Multiple photodiodes pick up the values.
- They use additional lenses to correct for the lens distortion.
- They are able to calculate the latency for every frame but only report one average.

**Method 4**
- Stauffert et al. [20] use a similar approach but encode the orientation of a Vive tracker on the HMD screen.
- They report the latency difference for every frame and visualise how latency varies.
Augmented Reality measurements work similarly.

Gruen et al. [7] use a video see-through system. A high resolution timer is observed by a camera. A synchronised camera observes the HMD screen that shows the timer with a delay.

Billeter et al. [2] use an optical see-through system where one camera can record both the real and the virtual LED timer side by side.

Method 1

Method 2

Method 3

Swindels et al. [22] record a virtual version projected on a half-silvered mirror of a real rotating object to calculate the latency from the angular difference.

Which approach to use?

Frame counting is the fastest and easiest method: Record some real object and its virtual counterpart on screen. Then analyse the video to find the time difference between some detectable event.

Sine fitting guards against imprecisions due to limited spatial and temporal resolution of cameras.

Photodiode-based methods can offer more precision and higher temporal resolution to observe latency’s temporal behaviour. They, however, need more investment in hardware, a thought through setup and appropriate software for analysis.

The best case would be to measure during VR exposure to know what has actually happened instead of measuring before.

Choose what fits for your experiment but always measure latency! No experiment should be reported without a latency measurement.
REFERENCES


