Towards Comparable Evaluation Methods and Measures for Timing Behavior of Virtual Reality Systems

Jan-Philipp Stauffert* University of Würzburg
Florian Niebling† University of Würzburg
Marc Erich Latoschik‡ University of Würzburg

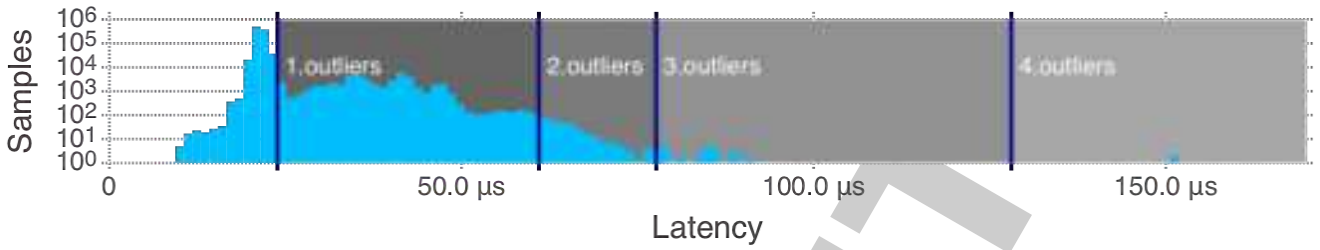

Figure 1: *Histogram visualization illustrating the distribution and categorization of latency measures using a logarithmic y-axis. The example shows the results gained by a modified z-score test.*

Abstract

A low latency is a fundamental timeliness requirement to reduce the potential risks of cyber sickness and to increase effectiveness, efficiency, and user experience of Virtual Reality Systems. The effects of uniform latency degradation based on mean or worst-case values are well researched. In contrast, the effects of latency jitter, the distribution pattern of latency changes over time has largely been ignored so far although today's consumer VR systems are extremely vulnerable in this respect. We investigate the applicability of the Walsh, generalized ESD, and the modified z-score test for the detection of outliers as one central latency distribution aspect. The tests are applied to well defined test cases mimicking typical timing behavior expected from concurrent architectures of today. We introduce accompanying graphical visualization methods to inspect, analyze and communicate the latency behavior of VR systems beyond simple mean or worst-case values. As a result, we propose a stacked modified z-score test for more detailed analysis.

Keywords: virtual reality, latency, outlier, cyber sickness

Concepts: •**Software and its engineering** → **Virtual worlds software;** •**General and reference** → *Metrics;* •**Computer systems organization** → *Reliability;* Multicore architectures; Real-time system architecture;

*e-mail:jan-philipp.stauffert@uni-wuerzburg.de
†e-mail:florian.niebling@uni-wuerzburg.de
‡e-mail:marc.latoschik@uni-wuerzburg.de

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org. © 2016 ACM.
VRST '16,, November 02-04, 2016, Garching bei München, Germany
ISBN: 978-1-4503-4491-3/16/11 ...$15.00
DOI: http://dx.doi.org/10.1145/2993369.2993402

PREPRINT

1 Introduction

Virtual reality applications are complex systems that consist of multiple interdependent parts to handle input, simulation and output. Thereby it has to be ensured that the processing is fast enough to allow for a fluid experience. In computer science in general and VR in particular timing behaviour is compared in regards to average case or to the worst case. We argue that this is not enough when it comes to latency and latency changes in particular. We need additional and more detailed information to analyse, detect, communicate and compare timing behaviour of systems which require a high timeliness in VR and related fields.

Latency spikes are not yet understood enough with respect to cyber sickness. For an in-depth analysis of their effects on VR users, latency spikes have to be measured as well as separated from the expected latency distribution inherent in the system.

In this paper, we look into this second step on how to separate outliers from other measurements and find descriptive visualisations.

The contributions of the work presented here are as follow:

1. We assess different outlier tests on their suitability to extract outlier data from latency measurements.

2. We develop, test, and propose recursive application of outlier tests based on a repeated computation of outliers on outliers to get multiple levels of severity for outliers.

3. We present visualization examples suitable to inspect and communicate latency and latency jitter data and patterns not easily captured in single measurement values currently available (see Figure 1 as an example).

This paper is structured into first discussing related research, followed by the introduction and description of our test data that will be used to assess the proposed methods that are explained afterwards. In the end there is a discussion of the findings with a conclusion and ideas for future work extending the research presented here.

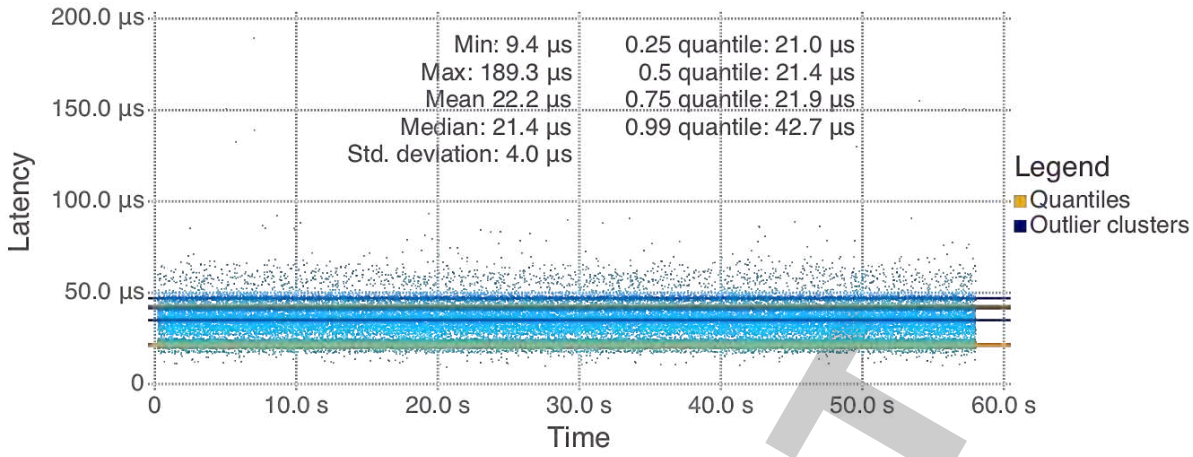

Figure 2: *The measured latencies plotted over time. The point size is chosen small to better show the structure of the data. The orange lines show the 25%, 50%, 75% and 99% quantiles. The first three quantiles are close enough together to not allow a distinction between their respective lines. Most of the measurements are below 24µs. Outliers cluster around certain values indicated by the blue lines.*

2 Related Work

Simulator sickness is a problem of VR applications where users are experiencing symptoms such as nausea [Kennedy et al. 1993]. Visual delay was found as a major contributing factor already in early simulators [Frank et al. 1988]. Latency also influences the performance of test subjects both if time variant latency is added [Ivkovic et al. 2015] or for latency spikes [Teather et al. 2009]. The assumption is consequently that latency spikes influence simulator sickness with a similar impact as the better researched time invariant latency.

The performance of VR applications is usually assessed by measuring motion-to-photon latency which tracks the time between an input on a certain input channel and the time it takes to show its effect on a display. Approaches to measure this latency are sine fitting [Steed 2008], light sensing [Di Luca 2010] and automated frame counting [Friston and Steed 2014]. In this paper, the focus is on latency that is contributed by the VR application, a subset of motion-to-photon latency. While there are many optimization techniques for the rendering stage like frameless rendering [Bishop et al. 1994], latency at the application stage is yet less researched.

VR systems often consist of multiple software components to handle various input and output modalities that run in parallel or on distributed machines [Latoschik and Tramberend 2012; Allard et al. 2004]). This facilitates latency jitter, especially in the communication of the different modules [Stauffert et al. 2016].

Outliers are defined as "observations that deviate so much from other observations as to arouse suspicion that it was generated by a different mechanism." [Hawkins 1980]. In our case, we assume that outliers are also caused by factors outside of the application such as interrupts or other software running on the same computer. They are equally dependent on the application that is examined and its surroundings which is why it is not possible to find one fixed threshold that works for all applications.

Here, we examine the results using the Walsh [Walsh and others 1950], the generalized ESD [Rosner 1983] and the modified z-score [Iglewicz and Hoaglin 1993] outlier test on their suitability to extract outliers from latency measurement data. See [Hodge and Austin 2004] for a discussion of different approaches and application fields.

3 Method

We adapted the method of [Stauffert et al. 2016] to obtain latency measurements. The test measures the time of message passing between two pairs of actors, a common task for VR systems that need to employ parallelism to maximise performance. While the additional actor scheduling will produce its very unique latency jitter and distribution patterns, this would be equally true for any alternative concurrency scheme.

As a testing platform, we use a Raspberry Pi 2 running Raspbian on a Linux kernel (version 4.4.9) with the kernel timer resolution set to 1000Hz for lower response times. The tests are based on the C++ actor library CAF [Charousset et al. 2014].

Figure 2 shows the measured latencies for each communication over time. Most measures fall in a small range indicated by the orange lines for the 25%, 50% and 75% quantiles that are too close together to be distinguishable on the plot. It is evident, that they are not sufficient to describe the distribution. Blue lines indicate clusters of outliers as can be found by peaks in the histogram depicted in Figure 1.

We will analyse the data in non overlapping windows. The time window size here is chosen arbitrarily as 1s. This was done to allow to compare the results of our tests for multiple time frames. The size of the time window needs to be chosen dependent on the property that is measured. It has to be wide enough to contain enough samples to conduct the tests but needs to be small enough to preserve temporal descriptiveness. Applications will try to keep the window as small as possible to be able to attribute certain events to outliers and react timely.

In the following, we are looking for a suitable test to classify outliers. We conduct the three examined tests over non overlapping time windows of one second with the total gathered data spanning one minute. This is to show the performance over multiple time slices that follow the same underlying mechanics but can change due to outside factors.

3.1 Distribution of Measurements

The samples describe the interference patterns of the algorithmic base that the tests are build upon. We expect the frequencies de-

PREPRINT

scribing the sending and receiving algorithms to be interfered by operating system frequencies, other software frequencies as well as hardware influences. While the interference pattern does not follow a normal distribution, we expect the message passing algorithm on its own to approach a normal distribution for a sufficiently long measurement interval. Extraneous influences then lead to a skewing of the distribution. Consequently, our measures do not follow a normal distribution as tested with the Anderson Darling test provided in the R library "nortest" [Gross and Ligges 2015] with a p-value of $< 2.2e-16$.

3.2 Walsh Outlier Test

The Walsh outlier test [Walsh and others 1950] is a nonparametric test to detect multiple outliers. In contrast to many other statistical tests that require a normal distribution, this test works on data that is not normally distributed.

The test shows whether a suspected amount of outliers is present in the data with a level of significance that is dependent on the sample size ($\alpha = 0.1$ if $60 < n \leq 220$ and $\alpha = 0.05$ if $n > 220$). The test is run with an increasing number of suspected outliers where the highest amount of suspected outliers k that still test positive is taken.

The k largest latencies are then classified as outliers. The test is computed by determining the values

$$\begin{aligned} c = \lceil \sqrt{(2n)} \rceil; r = k + c; \qquad b^2 = \frac{1}{\alpha} \\ a = \frac{1 + b\sqrt{\frac{c-b^2}{c-1}}}{c - b^2 - 1} \end{aligned} \tag{1}$$

If $X_{n+1-k} - (1+a)X_{n-k} + aX_{n+1-r} > 0$ then the k largest points are outliers, where X_i are the sorted values of the time window such that $X_1 < X_2 < \cdots < X_n$.

The criteria tests differences of samples therefore being sensitive to samples that are distant from others. However, if there are more outliers in close neighborhood these are not detected.

The Walsh test therefore helps to capture the extreme outliers in time windows. There are some windows where no outlier is found because they are grouped too densely even though the same latency was flagged to be an outlier in a different time window.

3.3 Generalized ESD Outlier Test

As described, we expect the observed algorithm to approach a normal distribution that gets influenced by outside factors. Therefore, we try a different outlier test that assumes normal distribution, which promises to separate the samples of the expected normal distribution from samples that were influenced.

We recursively repeat the generalized ESD test [Rosner 1983] on the outliers to distinguish between outliers and outliers of outliers. This leads to a separation of the majority of the measurements from their outliers but then additionally identifies severe outliers. Assuming the influencing factors themselves approach a normal distribution, we separate different influences and their accumulated interference patterns from each other. This coincides with the visual impression from Figure 2 where most outliers are clustered above the mean values with few extreme values.

To determine whether a measured latency x_i of the observed time window is an outlier, the values R_i and λ_i are calculated

$$\begin{aligned} R_i &= \frac{\max|x_i - \bar{x}|}{\sigma} \\ \lambda_i &= \frac{(n-i)t_{p,n-i-1}}{\sqrt{(n-i-1+t^2_{p,n-i-1})(n-i+1)}} \\ p &= 1 - \frac{\alpha}{2(n-i+1)} \end{aligned} \tag{2}$$

where $t_{p,\nu}$ is the $100p$ quantile of the t distribution with ν degrees of freedom, $\bar{x}$ the sample mean, σ the sample standard deviation and α the significance level here set as 0.05. The largest i that satisfies $R_i > \lambda_i$ is the number of outliers in the sample.

We tried to substitute the mean and standard deviation over the values of the time window with the respective functions over the complete test run. This would allow to run the application once to gather a representative mean and standard deviation of the values and then use those for subsequent test runs. This, however, increases the lower threshold and therefore performs worse in detecting outliers.

3.4 Modified z-score outlier Test

The last test conducted is a modified z-score outlier test [Iglewicz and Hoaglin 1993], which assumes normal distribution as well. The modified z-score Z_i is computed for each value x_i in the time window to be

$$\begin{aligned} Z_i &= \frac{0.6745(x_i - \tilde{x})}{\text{MAD}} \\ \text{MAD} &= \text{median}(|x_i - \tilde{x}|) \end{aligned} \tag{3}$$

Where $\tilde{x}$ is the median over all samples and MAD the median absolute deviation.

We changed this test to not take the median absolute deviation and median of the samples in the window to be tested but of all the measured samples. This allows to run an application using the measurements to determine the absolute median deviation and the median of this sample and use those values for subsequent application runs to assess the performance. This changes equation 3 to calculate the median and MAD for the values of the first run w_i with

$$\begin{aligned} \tilde{w} &= \text{median}(w_i) \\ \text{MAD}_w &= \text{median}(|w_i - \tilde{w}|) \end{aligned} \tag{4}$$

and then calculate the z-scores for all subsequent runs with

$$Z_i = \frac{0.6745(x_i - \tilde{w})}{\text{MAD}_w} \tag{5}$$

The threshold was chosen to be 3.5 as suggested by the authors. Recursive use of this test yields more gradations of outliers than the generalized ESD test. Here, we distinguish between the main part of the outliers, an area above with only few outliers and the rare extremes.

4 Discussion

The Walsh outlier test only captures few extreme outliers. These extremes are supposed to have the most impact and are therefore the most interesting for further examination.

PREPRINT

The generalized ESD and modified z-score tests are similar to each other. Both support the classification of outliers into multiple levels by stacking them so they can be used to determine how severe an outlier is. The modified z-score allows for finer separation. Additionally, it allows to determine base values like the MAD and median for one test run to establish a base line that can then be used for subsequent runs. When trying the same with the generalized ESD test, the lowest threshold to classify outliers moves up to then yield a value more distant than the threshold we would have chosen by inspecting the histogram.

5 Conclusion

VR applications get optimised for mean and worst case behaviour, which we argue is not enough to capture latency behaviour as it does not account for different patterns of latency outliers.

We have discussed three tests to find outliers in latency measurements. The measurements here were taken as the time needed for the communication of two actors, a common process in VR systems that consist of multiple parts. The measurements were then analyzed in time windows to assess how good they detected outliers over time.

The Walsh test allows to catch the extreme outliers. Those are supposed to have the most impact on an application's performance. Finer analysis is offered by using a stacked modified z-score test that groups outliers into categories of different severity.

We propose to first establish a base line by running an application once to calculate the median and MAD over the latency samples. Subsequent runs can then be analyzed to determine what category of outlier a latency sample belongs to.

The proposed recursive application of outlier tests by repeating a test on the detected outliers multiple times yields several categories of outliers. These different levels of severity can then be used to evaluate an application on multiple scales.

6 Future Work

The impact of latency spikes on cyber sickness is not yet tested, which is necessary to evaluate the impact of the measured latency spikes on the user. We have laid a base to measure latency spikes and gather outliers. This data can then be used to correlate it with symptoms of cyber sickness shown in tests where such spikes are artificially added, enhanced or altered in their pattern. Real-time systems where far less latency spikes are observed as discussed in [Stauffert et al. 2016] can be used to test against.

Using the gathered outlier data, it will be possible to derive an outlier fingerprint of an application on a specific hardware. Comparing this to a different application's outlier fingerprint running on the same hardware might open up the possibility to compare applications by their latency behavior, benchmark them and propose improvements.

We have used different visualisation methods to allow for an intuitive interpretation of outliers. Future research has to use user studies to show how intuitive these graphs are and how convenient they are to spot unwanted application behaviour.

References

ALLARD, J., GOURANTON, V., LECOINTRE, L., LIMET, S., MELIN, E., RAFFIN, B., AND ROBERT, S. 2004. FlowVR: a middleware for large scale virtual reality applications. In *Europar 2004 Parallel Processing*, Springer, 497–505.

BISHOP, G., FUCHS, H., MCMILLAN, L., AND ZAGIER, E. J. S. 1994. Frameless rendering: Double buffering considered harmful. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, ACM, 175–176.

CHAROUSSET, D., HIESGEN, R., AND SCHMIDT, T. C. 2014. CAF - the C++ Actor Framework for Scalable and Resource-Efficient Applications. ACM Press, 15–28.

DI LUCA, M. 2010. New method to measure end-to-end delay of virtual reality. *Presence 19*, 6, 569–584.

FRANK, L. H., CASALI, J. G., AND WIERWILLE, W. W. 1988. Effects of visual display and motion system delays on operator performance and uneasiness in a driving simulator. *Human Factors: The Journal of the Human Factors and Ergonomics Society 30*, 2, 201–217.

FRISTON, S., AND STEED, A. 2014. Measuring latency in virtual environments. *Visualization and Computer Graphics, IEEE Transactions on 20*, 4, 616–625.

GROSS, J., AND LIGGES, U. 2015. *nortest: Tests for Normality*. R package version 1.0-4.

HAWKINS, D. M. 1980. *Identification of outliers*, vol. 11. Springer.

HODGE, V. J., AND AUSTIN, J. 2004. A survey of outlier detection methodologies. *Artificial Intelligence Review 22*, 2, 85–126.

IGLEWICZ, B., AND HOAGLIN, D. 1993. *Volume 16: how to detect and handle outliers, The ASQC basic references in quality control: statistical techniques, Edward F. Mykytka*. PhD thesis, Ph. D., Editor.

IVKOVIC, Z., STAVNESS, I., GUTWIN, C., AND SUTCLIFFE, S. 2015. Quantifying and Mitigating the Negative Effects of Local Latencies on Aiming in 3d Shooter Games. ACM Press, 135–144.

KENNEDY, R. S., LANE, N. E., BERBAUM, K. S., AND LILIENTHAL, M. G. 1993. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology 3*, 3, 203–220.

LATOSCHIK, M. E., AND TRAMBEREND, H. 2012. A scala-based actor-entity architecture for intelligent interactive simulations. In *Software Engineering and Architectures for Realtime Interactive Systems (SEARIS), 2012 5th Workshop on*, IEEE, 9–17.

ROSNER, B. 1983. Percentage Points for a Generalized ESD Many-Outlier Procedure. *Technometrics 25*, 2 (May), 165.

STAUFFERT, J.-P., NIEBLING, F., AND LATOSCHIK, M. E. 2016. Reducing Application-Stage Latencies For Real-Time Interactive Systems. In *9th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)*, IEEE Computer Society.

STEED, A. 2008. A Simple Method for Estimating the Latency of Interactive, Real-time Graphics Simulations. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology*, ACM, New York, NY, USA, VRST '08, 123–129.

TEATHER, R. J., PAVLOVYCH, A., STUERZLINGER, W., AND MACKENZIE, S. I. 2009. Effects of tracking technology, latency, and spatial jitter on object movement. In *3D User Interfaces, 2009. 3DUI 2009. IEEE Symposium on*, IEEE, 43–50.

WALSH, J. E., AND OTHERS. 1950. Some nonparametric tests of whether the largest observations of a set are too large or too small. *The Annals of Mathematical Statistics 21*, 4, 583–592.

PREPRINT