

Development and Evaluation of a Ingame Latency Compensation Technique Artificial Neural Networks

Thema:

Development and Evaluation of a Ingame Latency Compensation Technique Artificial Neural Networks

Art:

BA

BetreuerIn:

David Halbhuber

BearbeiterIn:

Julian Hoepfinger

Status:

Entwurf

angelegt:

2021-04-27

Hintergrund

“Latency from a general point of view is a time delay between the cause and the effect of some physical change in the system being observed” [1]. Latency in video games impairs the performance of the player and game experience. The performance in games, which require precision and have tight deadlines like Counter-Strike: Global Offensive (CS:GO), is dependent on the latency [2, 3]. There are three main latency compensation techniques to improve game experience and performance [4].

- Delayed input techniques try to add delay to local actions to allow simultaneous execution by all clients [4, 5].
- Time-offsetting techniques enable a rollback to the previous state of the game [4, 5].
- Predictive techniques estimate the occurring events in the game from the locally available state [4, 5].

The goal of this work is to build a predictive technique for CS:GO using artificial neural networks (ANN). ANNs have been successfully used in computer vision tasks, such as pattern recognition in images [6]. One problem with ANNs is their long runtime due to internal complexity [7]. If used for latency compensation in video games, the ANNs inference needs to be fast and lightweight. If inference takes too long an ANN based latency compensation technique ultimately leads to an increase in overall latency.

This work aims to build and evaluate a predictive system to decrease perceived latency for CS:GO.

Zielsetzung der Arbeit

Firstly a suitable way to collect data from CS:GO needs to be implemented. There are two possible ways

to collect data: (1) Directly log user input and game screen while live-playing and (2) use openly available CS:GO replays. Both approaches need to be evaluated and analysed regarding their feasibility for this work's aim.

After collecting a suitable data set from either one of the above options the ANN needs to be developed. The ANN's prediction is based on the previously gathered data and ultimately should be able to infer the user's next inputs.

After training the ANN, it is evaluated in various scenarios. The goal is to determine if the developed predictive system is suitable for latency compensation in CS:GO.

Konkrete Aufgaben

- Literature research
- Create a suitable dataframe for ANN training
- Train the latency compensation model
- Evaluate the developed model
- Write the thesis

Erwartete Vorkenntnisse

- Tensorflow, Pytorch or Keras
- Programming in C++/Python and Unity
- Data analysis

Weiterführende Quellen

[1] Latency. (2021). Wikipedia. [https://en.wikipedia.org/wiki/Latency_\(engineering\)](https://en.wikipedia.org/wiki/Latency_(engineering)), Retrieved: 21.04.2021. [2] Claypool, M., & Claypool, K. (2006). Latency and player actions in online games. *Communications of the ACM*, 49(11), 40-45. [3] Claypool, M., & Claypool, K. (2010, February). Latency can kill: precision and deadline in online games. In *Proceedings of the first annual ACM SIGMM conference on Multimedia systems* (pp. 215-222). [4] Savery, C., & Graham, T. N. (2013). Timelines: simplifying the programming of lag compensation for the next generation of networked games. *Multimedia Systems*, 19(3), 271-287. [5] Sabet, S. S., Schmidt, S., Zadtootaghaj, S., Naderi, B., Griwodz, C., & Möller, S. (2020, May). A latency compensation technique based on game characteristics to mitigate the influence of delay on cloud gaming quality of experience. In *Proceedings of the 11th ACM Multimedia Systems Conference* (pp. 15-25). [6] Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)* (pp. 1-6). IEEE. [7] Cai, E., Juan, D. C., Stamoulis, D., & Marculescu, D. (2019). Learning-based Power and Runtime Modeling for Convolutional Neural Networks.

From:

<https://wiki.mi.ur.de/> - **MI Wiki**

Permanent link:

<https://wiki.mi.ur.de/arbeiten/latencycsgo?rev=1619555351>

Last update: **27.04.2021 20:29**

