Real-time lens distortion algorithm on embedded GPU systems

Young-Woo Kim, Duksu Kim

KOREATECH

SIGGRAPH 2022
VANCOUVER+ 8-11 AUG

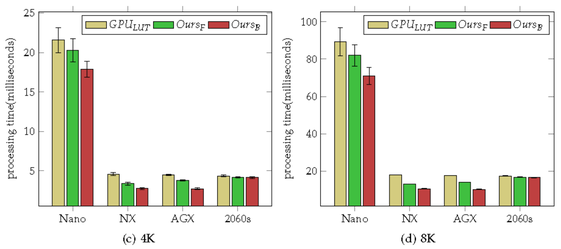
PROBLEM

- The optical lens system on the HMDs occurs radial distortion, and it requires a pre-distortion process to correct it [1]
- In most current VR systems, it performs such pre-distortion computation on a separate PC (or laptop) due to the computational overhead and power consumption
- This is one of the obstacles to the popularization of VR and HMDs

METHOD

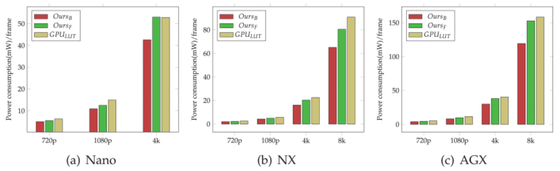
- We propose to use an embedded GPU system, whose performance has recently improved significantly
- We introduce a novel lens distortion algorithm fits to the embedded GPU system
  - Using a compressed lookup table
    ➔ Decreasing memory transaction
  - Utilizing integrated memory architecture
    ➔ Optimizing communication between CPU and GPU

RESULTS



(c) 4K (d) 8K

- Our method achieved up to 1.72 times higher performance than the prior look-up table approach

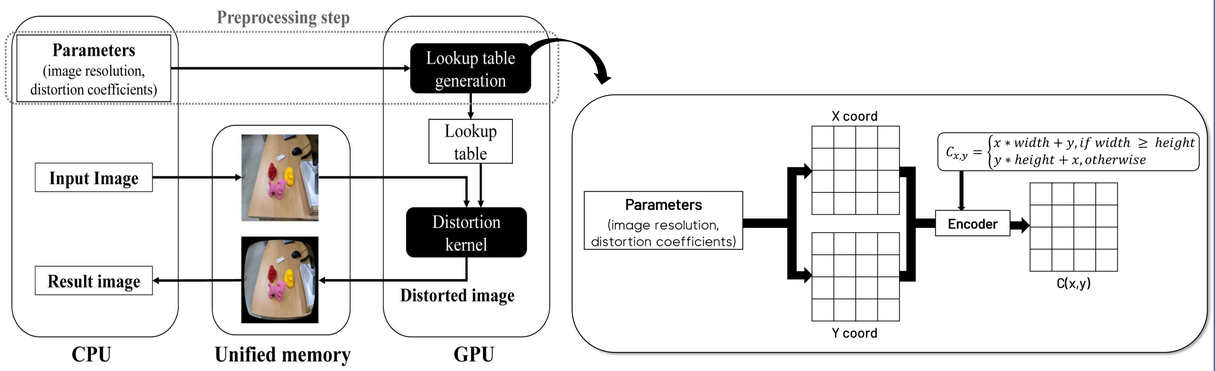


(a) Nano (b) NX (c) AGX

- On embedded GPU systems, our compressed lookup table approach takes up to 24.86% less power than prior algorithms

By compressing the lookup table and utilizing the unified memory architecture of the embedded GPU system, our method speeds up to 1.72 while reducing power consumption by 25%

System overview



Lookup table compression method

OUR APPROACH

- At the preprocessing step, it encodes the separated lookup for the x- and y-axis into a single table (compression)
  - The encoding key is the image resolution
- At runtime, it requires only a memory transaction to see the lookup table
  - Note that the lookup table-based lens distortion algorithm is a memory I/O-bounded task
  - Our approach replaces the memory access overhead with a computational one
- In our system, the input and output data are passed through the unified memory between the host (CPU) and device (GPU)
  - Most embedded systems have an integrated host-device memory

RELATED WORK

- Lookup table-based lens distortion correction method [2]
- Parallel lens distortion algorithms on specialized hardware [3] or GPU [4]
- Recent edge devices and their architectural characteristics [5]

REFERENCE

[1] Warren Robinett and Jannick P Rolland. 1992. A computational model for the stereoscopic optics of a head-mounted display. Presence: Teleoperators & Virtual Environments 1, 1 (1992), 45–62.
[2] Shehrzad Qureshi. 2013. Computer Vision Acceleration Using GPUs. http://developer.amd.com/wordpress/media/2013/06/2162_final.pdf. AMD Developer Central, Accessed: 2021-07-08
[3] Nikolaos Bellas, Sek M Chai, Malcolm Dwyer, and Dan Linzmeier. 2009. Real-time fisheye lens distortion correction using automatically generated streaming accelerators. In 2009 17th IEEE Symposium on Field Programmable Custom Computing Machines. IEEE, 149–156.
[4] Sam Van der Jeught, Jan AN Buytaert, and Joris JJ Dirckx. 2012. Real-time geometric lens distortion correction using a graphics processing unit. Optical Engineering 51, 2 (2012), 027002
[5] NVIDIA, "CUDA for tegra :: CUDA toolkit documentation," https://docs.nvidia.com/cuda/cuda-for-tegra-appnote/index.html, accessed: 2021-07-12.

ACKNOWLEDGEMENTS

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2020-0-00594, Morphable Haptic Controller for Manipulating VR·AR Contents, 100%)

Young-Woo Kim, Duksu Kim
HPC Lab., KOREATECH
https://hpc.koreatech.ac.kr

Scan this QR codes for the full paper