DYNAMIC CT PERFUSION IMAGE DATA COMPRESSION FOR EFFICIENT PARALLEL PROCESSING

Barros R. S., Olabarriaga S. D., Borst J., van Walderveen M. A. A., Posthuma J. S., Streekstra G. J., van Herk M. B., Majoie C. B., Marquering H. A.

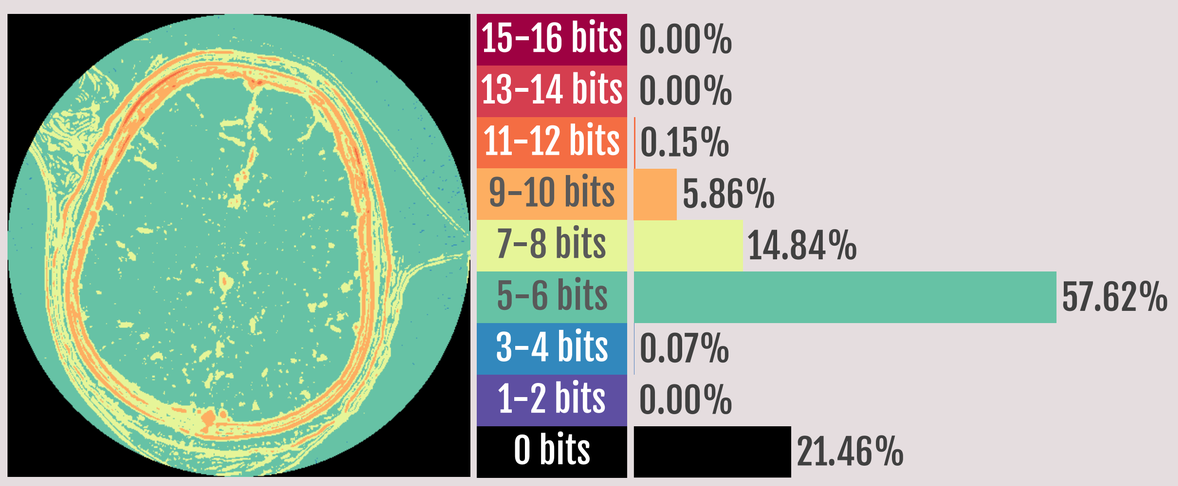
ABSTRACT: CT perfusion (CTP) image data requires new approaches to deliver timely results for acute care. Cloud architectures based on graphics processing units (GPUs) can provide the processing required for delivering fast results. However, the transfer of CTP datasets to the cloud is slow and so not suitable in acute situations. To reduce transfer time, we propose a fast and lossless compression algorithm for CTP data. To the best of our knowledge, this is the first work to present a GPU-ready method for medical image compression with random access to the image elements from the compressed data.

CONTEXT

CTP IMAGE DATA IS LARGE from **ACUTE STROKE PATIENTS** on **GPU-BASED CLOUD INFRASTRUCTURES** NEED FAST RESULTS ARE CHEAPER

DATA

24 whole-brain volumes with 320 slices of 512×512 voxels with 16 bits/voxel summing up 3840 MB per patient. Sometimes, an additional CTP dataset is produced after around 24 hours resulting in 7.5GB of data per patient.



ALGORITHM

WE REPRESENT THE IMAGE AS $I(\boldsymbol{x}, t) = C(\boldsymbol{x}) + \Delta(\boldsymbol{x}, t)$

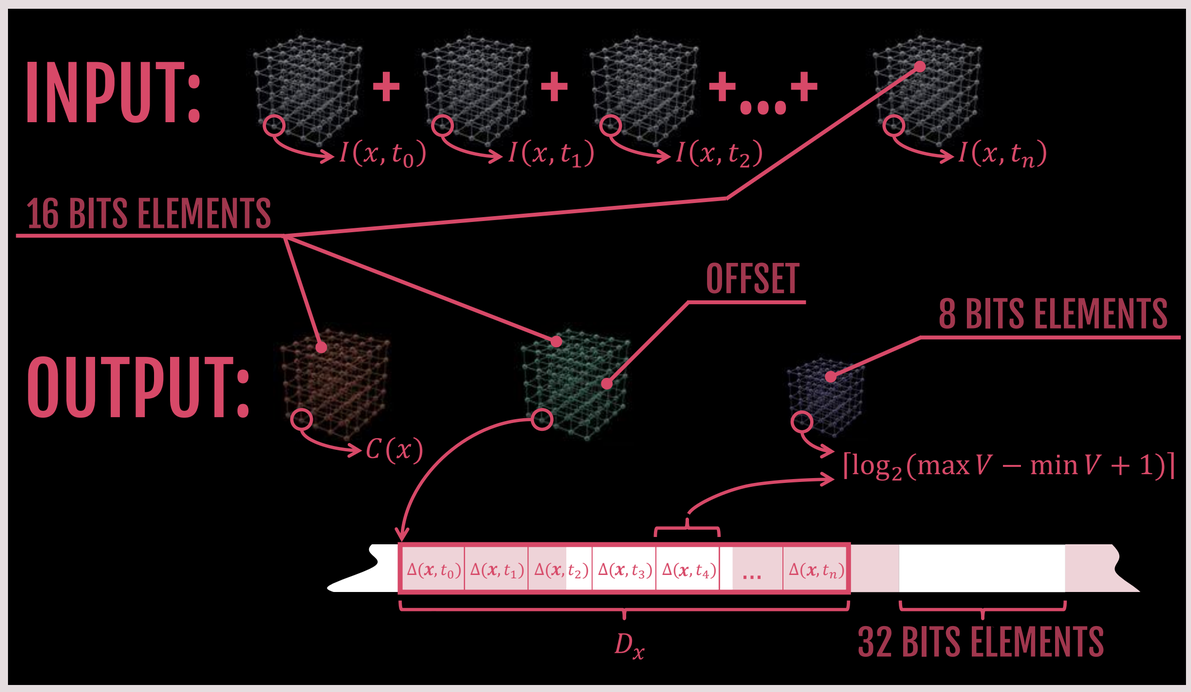
WITH $\min V_x \leq C(\boldsymbol{x}) \leq \max V_x$

AND $V_x = \{I(\boldsymbol{x}, t_0), \ldots, I(\boldsymbol{x}, t_n)\}$

FOR SIMPLICITY, WE USE $C(\boldsymbol{x}) = \min V_x$

$D_x = \{\Delta(\boldsymbol{x}, t_0), \ldots, \Delta(\boldsymbol{x}, t_n)\}$ IS REPRESENTED USING $\lceil \log_2(\max V_x - \min V_x + 1) \rceil \times n$ BITS

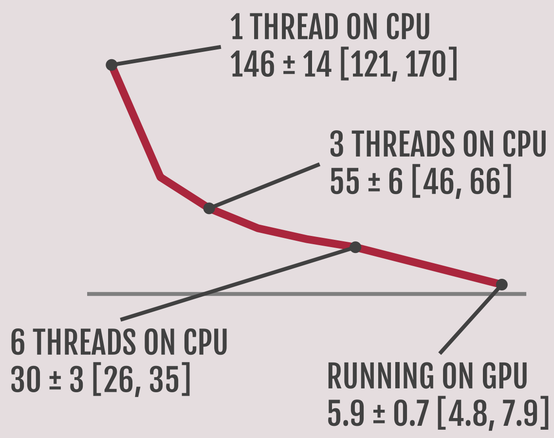
DATA STRUCTURES



RESULTS

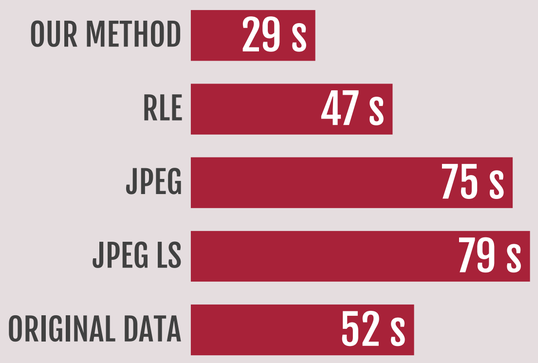
GPU-READY LOSSLESS COMPRESSION WITH RANDOM ACCESS TO VOXELS FROM COMPRESSED DATA

COMPRESSION TIME IN SECONDS AVG ± STD [MIN, MAX]

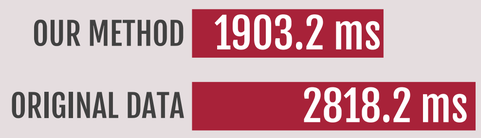


AVERAGE DATA SIZE IN GIGABYTES

AVERAGE TOTAL TIME (COMPRESSION + TRANSFER AT 622 Mbit/s + DECOMPRESSION)



GPU PROCESSING TIME IMPROVEMENT



Double threshold mask computation on GPU using the original CTP data and the compressed data generated by our method.