A Practical Encoding Approach for Texture Compression: Combining Multi-Processing and Multi-Threading

Hyeon-ki Lee[1], Jae-Ho Nah[1]

[1]Sangmyung University, Seoul, South Korea

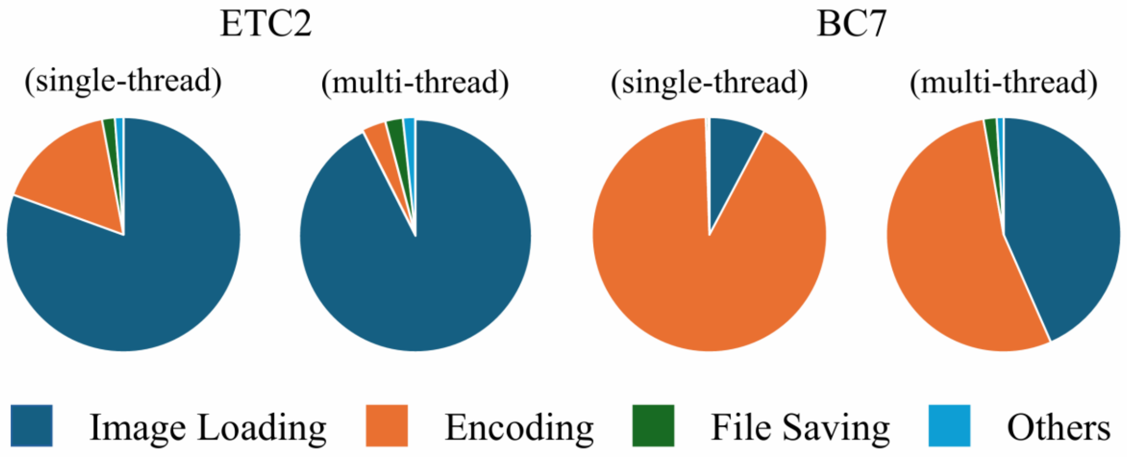
Problem

Typical Texture Compression Stages
(1) Image Loading
(2) Encoding
(3) File Saving

Prior Research Focus
- Encoding acceleration through multi-threading [1-4]

Remaining Bottlenecks
- Image loading and decoding (e.g., PNG) limit parallel performance

Time Ratio per Codec (ETC2, BC7)



Motivation

Modern CPUs have multiple cores
- Modern CPUs have multiple cores, enabling flexible parallel execution
- Multi-processing and threading improve encoding performance

Multi-processing performance can depend on file access strategy
- Ordering files by size can optimize processing sequence
- Helps balance workload and reduce CPU idle time

References

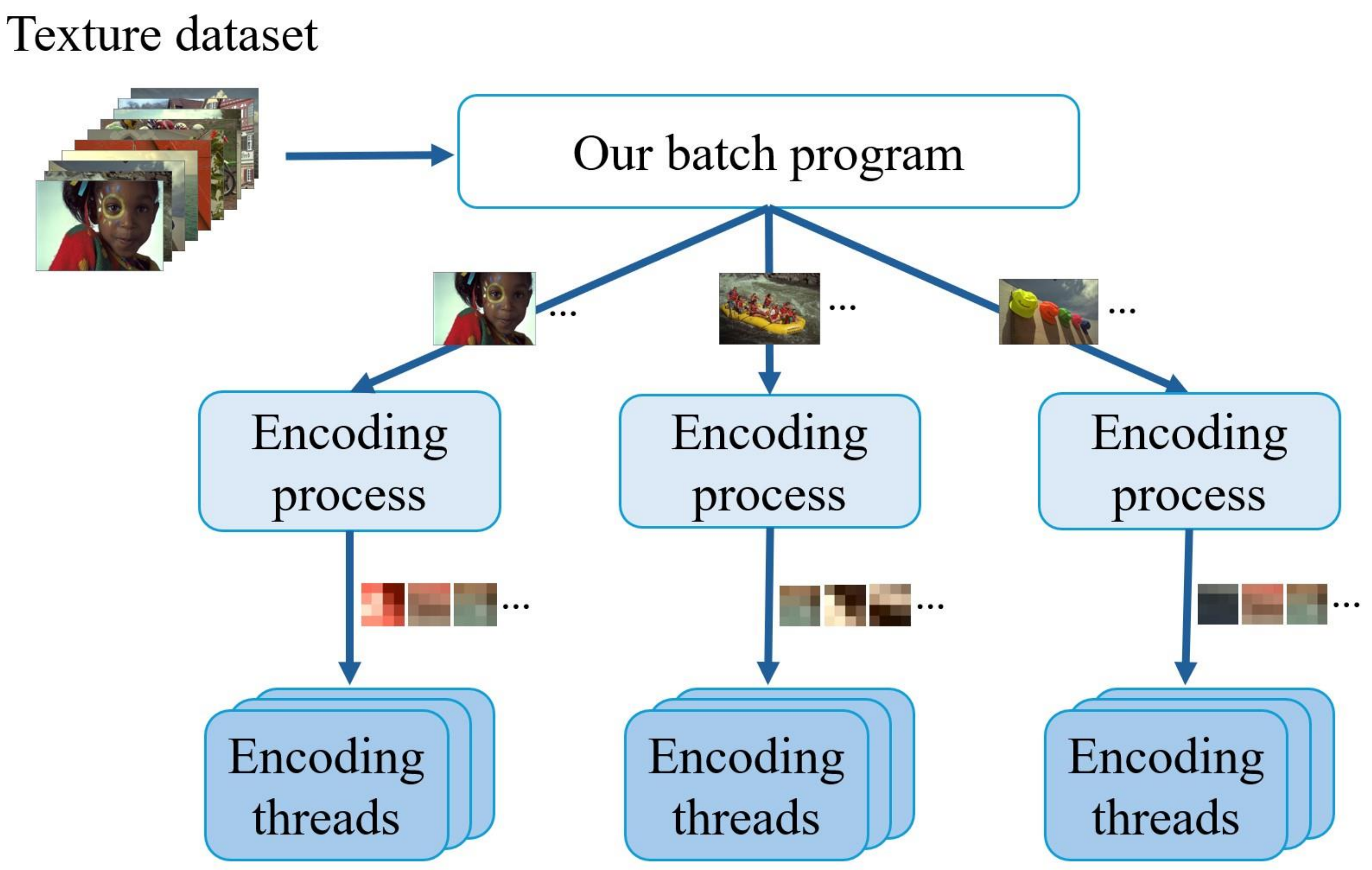
[1] Taudul, B. 2024. ***etcpak 2.0*: The fastest ETC compressor on the planet.** https://github.com/wolfpld/etcpak
[2] Lee, H. and Nah, J.H. 2023. **H-ETC2: Design of a CPU-GPU Hybrid ETC2 Encoder**. *Computer Graphics Forum (Pacific Graphics).*
[3] Nah, J.H. 2020. **QuickETC2: Fast ETC2 texture compression using Luma differences**. *ACM TOG (SIGGRAPH Asia).*
[4] Nah, J.H. 2023. **QuickETC2-HQ: Improved ETC2 encoding techniques for real-time, high-quality texture compression**. *Computers & Graphics.*

Contact email: 202533021@sangmyung.kr

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2025-00521436).

Our Approach

System Overview



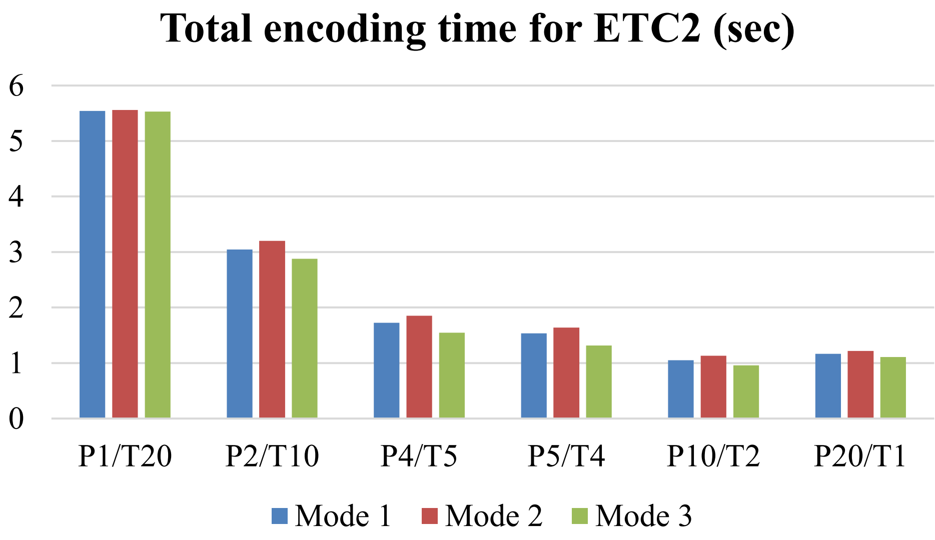
1. Sorting Files by Descending Size
- Input files are sorted in descending order before encoding
- Smaller files are processed later to balance the workload

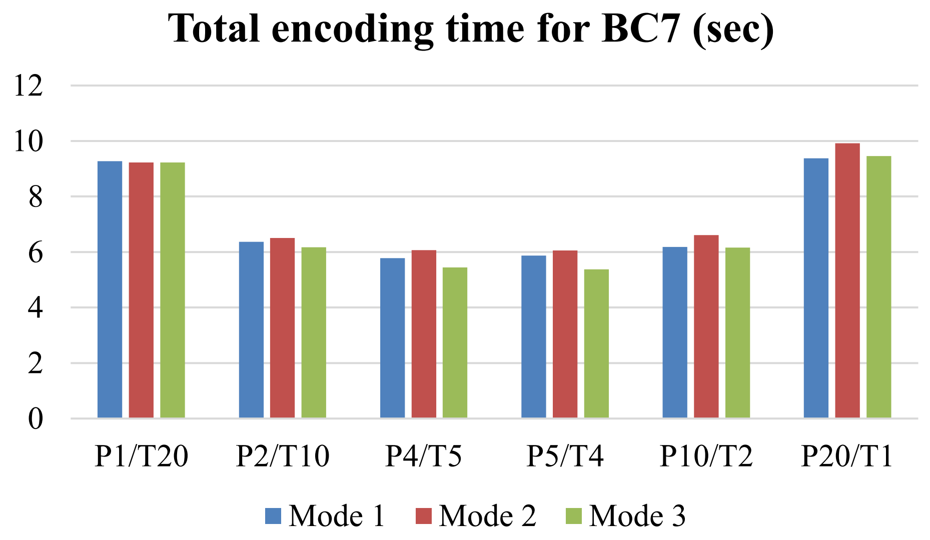
2. Combining Multi-processing and Multi-threading
- Multiple processes load and encode textures in parallel
- Each process uses multi-threading at the block level
- Reduces I/O overhead while maintaining fast encoding speed

Results

We measured encoding time (in sec)
using our batch program (based on ***etcpak 2.0*** **[1]**)
on **64 textures [3] (.png)** with an **Intel Core i7-12700 CPU (14C/20T).**

Tested Configurations:
- 6 Process/Thread (Px/Tx) setting,
- 3 file access modes: Random (1), Ascending (2), Descending by size (3)





Key Findings:

- Up to 5.76x speedup for ETC2 (P10/T2)
Image loading dominates total encoding time
Parallel loading significantly improves performance

- Up to 1.7x speedup for BC7 (P4/T5 or P5/T4)
Balanced parallelization of loading and encoding is essential
Especially important for high-complexity codecs like BC7