Improving Frame Rates with ML-based Frame Interpolation

Aditya Hota, Richard Chen, Kaan Erdogmus
CIS 565 Fall 2021

Why use ML to improve frame rates?

- Achieving high frame rates requires powerful hardware
  - Harder to attain on high settings
- Computing photorealistic effects with path tracing is very taxing
  - Difficult to perform high quality rendering in real-time
- ML models can take in video streams and increase frame rate
  - No need to render as many frames
  - Inference is less taxing, so less time needed to make a video
- We want to implement fast and functional NN directly in CUDA

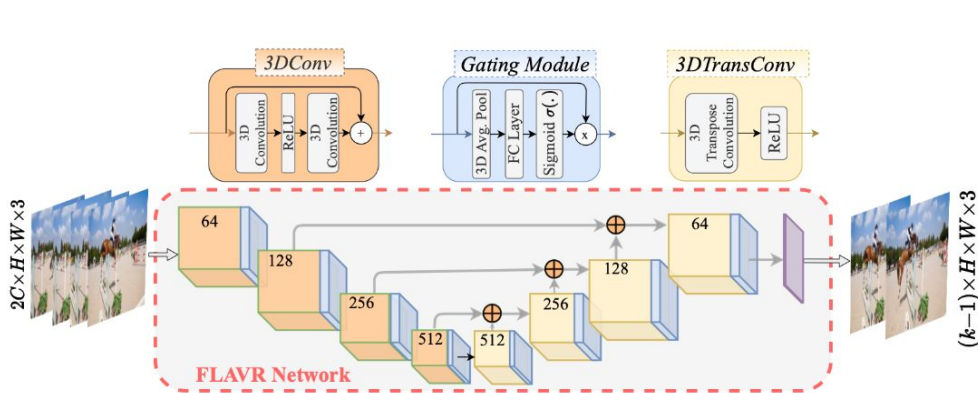# FLAVR: Flow-Agnostic Video Representations for Fast Frame Interpolation
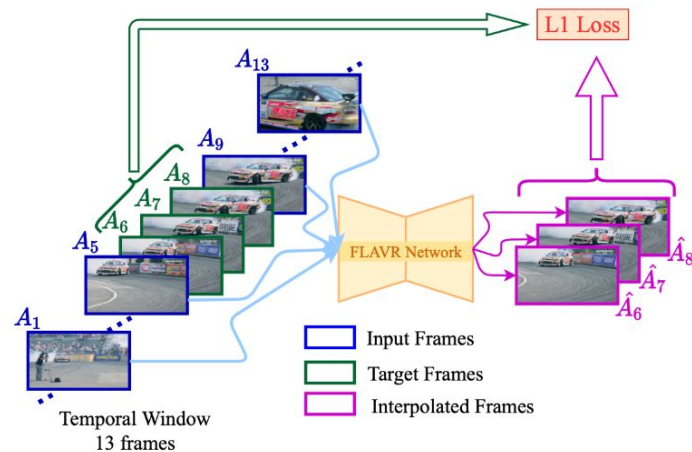
Tarun Kalluri *
UCSD

Deepak Pathak
CMU

Manmohan Chandraker
UCSD

Du Tran
Facebook AI

https://tarun005.github.io/FLAVR/

(a) Overview of the proposed architecture

(b) Sampling procedure

Goals & Technologies

- Implement FLAVR architecture in CUDA
    - Use work presented in the paper
    - Tune parameters for best performance with path tracer outputs
- Develop video processing pipeline
    - Automated process from input to output video
- Time permitting, add motion to CUDA Path Tracer
    - Camera movements or physics engine with objects

- CUDA code: based off [CUDA Neural Network Implementation](#)
- Image processing: [OpenCV CUDA library](#)

Milestones

- Milestone 1 (11/17)
    - Setup CUDA neural network code
    - Determine feasibility of implementing special layers in CUDA
    - Become familiar with CUDA NN code and train simple CNN for classification
- Milestone 2 (11/29)
    - Implement architecture for FLAVR
    - Training and classification should be working
- Milestone 3 (12/06)
    - Tune all hyperparameters for best results
    - Incorporate camera or object movement into Path Tracer
- Final Deliverable (12/12)
    - Automated pathway for generating interpolated videos
    - Performance analysis (and potential comparison to PyTorch)

References

Paper:

Kalluri, T., Pathak, D., Chandraker, M., & Tran, D. (2020). Flavr: Flow-agnostic video representations for fast frame interpolation. *arXiv preprint arXiv:2012.08512.*

3rd Party Code:

CUDA Neural Network Implementation