A GPU Programming Course Project

A Super-Resolution Video Player Based on GPU Accelerated Upscaling from Local Self-Examples

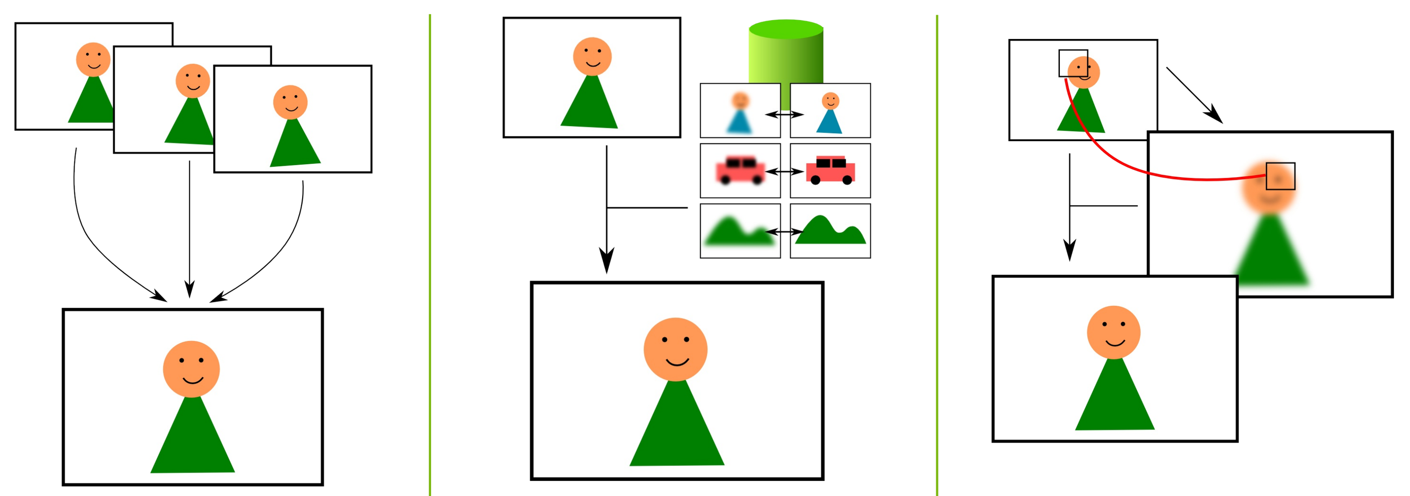
Yen-Hao Chen (陳彥豪), Yi-Da Wu (吳奕達), Chi-Wei Tseng (曾紀爲)
Instructed by Wei-Chao Chen (陳維超)

ABSTRACT

Many super-resolution algorithms have been proposed for upscaling static images, yet upscaling video footages in real-time with descent quality remains a challenging problem. In this project, we started out developing a super-resolution video player based on the work by Gilad Freedman and Raanan Fattal [4], for their proposed approach is essentially suitable for application in video upscaling. The algorithm take advantage of the fact that a nature image patch is redundant in its locality, hence, utilizes neighboring patches as hints on the high-frequency detail that is lost in interpolative upscaling. High memory locality makes this algorithm an excellent candidate to be implemented on commodity GPUs. With CUDA, we achieved a 38x acceleration and a frame rate of 10 fps for 1.5x upscaling on VCD quality video. Through analysing our resource usage, higher frame rates are anticipated through further development.

• from Image to Real-Time Video Upscaling

Classes of Approaches Proposed for Upscaling Images



Multiple Frame Image Reconstruction
Ref. [1]

Example-Based High Frequency Component Guessing
Ref. [2]

Example-Based High Frequency Component Guessing
Ref. [3]

Challenges in Extending Algorithms to Video in Real-Time

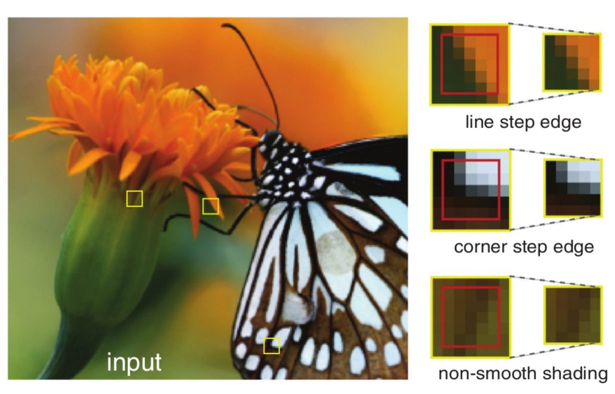
- High computational effort is required for descent quality in upscaled video.
- Quality of output video depends largely on video context.
- Incoherence between frames arises due to patch-by-patch nature in algorithms.

• Image Upscaling from Local Self-Examples

Patch Redundancy in A Restricted Window

Locally, a *slightly* downscaled image looks similar to a patch from the original image.

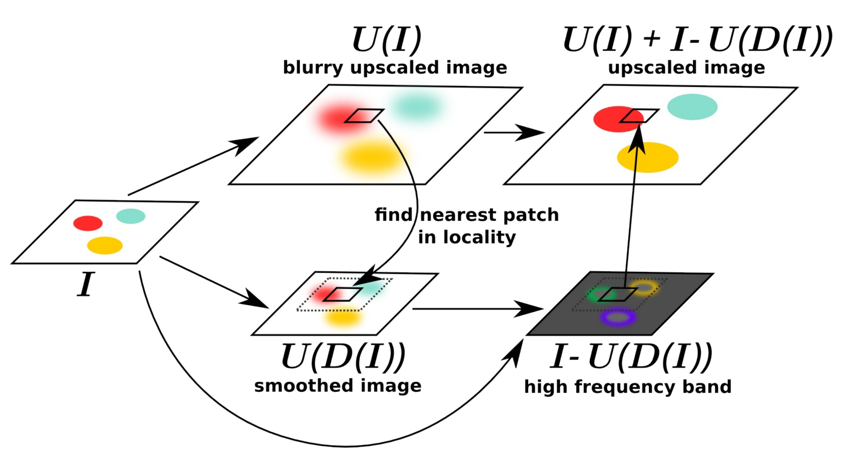
A patch from the *slightly* upscaled image looks similar to the original image within certain proximity.



Ref. [4]

- On upscaling, utilize the original image in a restricted window as a hint for high frequency detail.

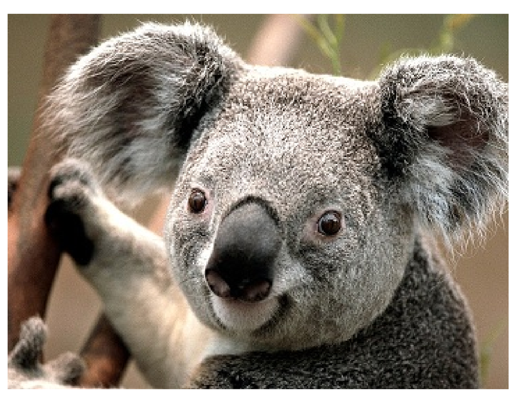
Algorithm Steps



- U: Interpolation Operator.
  D: Downsampling Operator.
- U and D are implemented with non-dyadic filter banks.
- Repeat several times for larger upscales.

Results from Single Frame

Left: Original image.
Right: Image of super resolution.



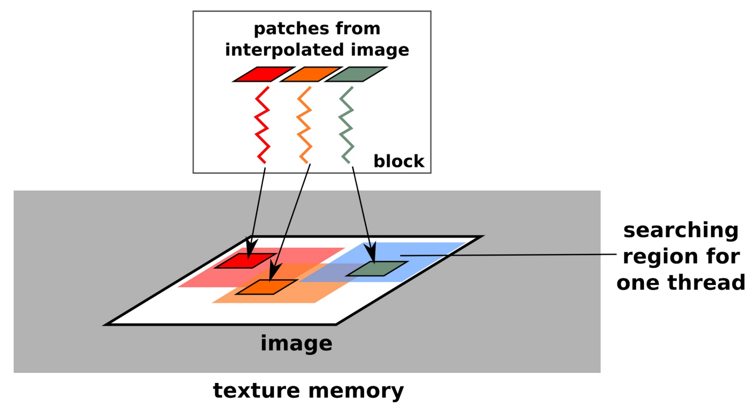
Potential in Applying to Real-time Video

- It is a highly parallelizable algorithm. Implementing on GPU is beneficial.
- Consistency is achieved between upsampled and original image: Guarantee output stability.

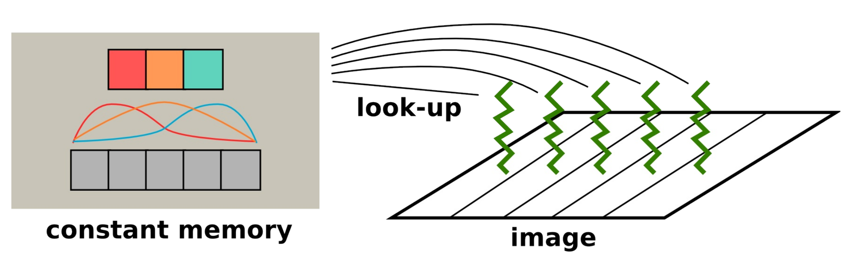
• Strategies for Accelerating with GPU

Taking Advantage of Locality in Nearest Patch Searching

- Storing input and intermediate images in texture memory: Efficient for memory access patterns with high 2D locality.
- A thread is responsible for matching one pair of patches



Implementing Efficient Non-Dyadic Filters



- Load non-dyadic filters into constant memory for efficient computation.
- A thread upsamples or downsamples one row of the image.

Utilizing CUDA-Specific Optimization Functionalities

CUDA Stream Processing

Stream processing enables us to pipeline video frame through the interpolation, downscaling and patch matching steps.

Interoperating with OpenGL

Directly shows the upscaled result in GPU. This reduces the time spent on moving frames between host and device memory.

Improvements in Efficiency

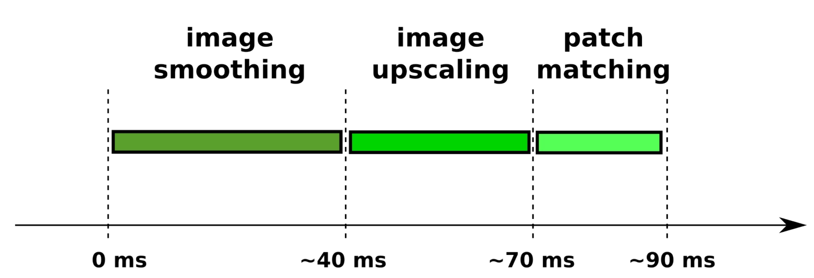
Acceleration	CPU Version	Naive GPU Version	Texture/Constant Memory Utilzed	Reorganize Register Usage
Processing Time Per Frame	4500 ms	800 ms	220 ms	120 ms
Improvement	1.0x	5.6x	20.5x	37.5x

• Challenges, Conclusion and Future Work

We have built a super-resolution video player which upscales frames from local examples. Up to poster submission, the achieved frame rate is 10 fps on upscaling video of VCD quality 1.5x, which is not satisfactory.

Bottleneck Analysis

- The smoothing and upscaling kernel, which intensively write to global memory, dominates the timeline.



Prospective

- The relatively simple image smoothing and upscaling procedure takes up most of the computation time.
- The power of shared memory is not utilized yet. Incorporating shared memory into the algorithm will be our next step.

• References

[1] "Fast and Robust Multiframe Super Resolution." S. Farsiu, M. Robinson, M. Elad, P. Milanfar, IEEE TIP 2004.
[2] "Example-based Super-resolution." W. Freeman, T. Jones, E. Pasztor, IEEE CG&A 2002.
[3] "Super-resolution from a Single Image." D. Glasner, S. Bagon, M. Irani, ICCV 2009.
[4] "Image and Video Upscaling from Local Self-Examples." Gilad Freeman, Raanan Fattal, ACMTOG 2011.

2012 GPGPU HD Koala Bears Team