Parallel Seam Carving

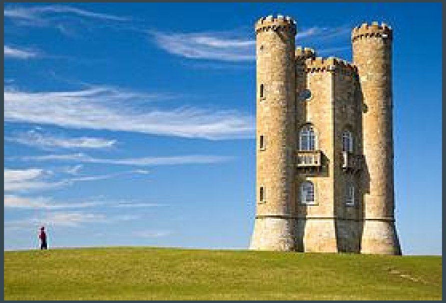
•••

Augusto Blomer
Victor Konen

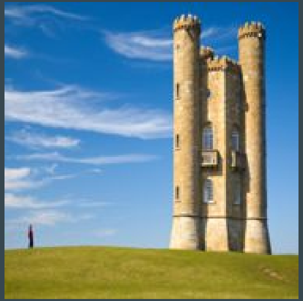
Introduction

- Resize images while maintaining content clarity
- Reduce image warping while downsizing
- Adjust aspect ratio with minimal distortion

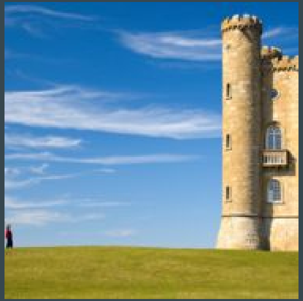
Original



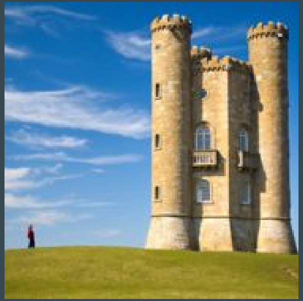
Classic Example



Scaling



Cropping

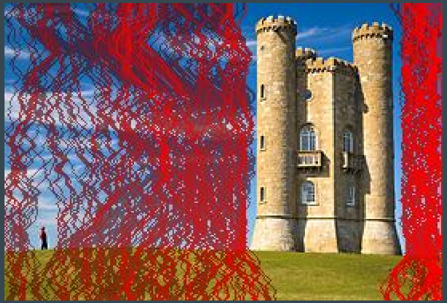


Seam Carving

Algorithm

- Retrieve and store pixel data for an image
- Remove seams from the image until it is the desired dimensions
- For each seam to be removed from the image:
  - Compute the “energy” of each pixel based on an energy function
  - Compute the seam with the least total energy
  - Remove that seam
- Store the resulting pixel data as an image

Lowest Energy Seams



Parallel Implementation

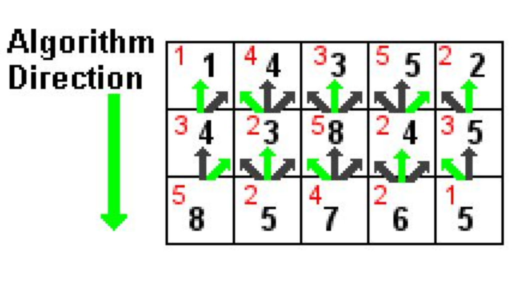
Energy Function Component

- Divide and Conquer Approach
  - Divide pixels evenly among processes
  - Compute energy for pixels assigned to each process
  - Communicate data to all processes

Parallel Implementation

Lowest Energy Seam Component

- Considerably more complex
- Dynamic programming exercise
- For each pixel in a row or column
  - Cost to get to that pixel is its energy plus the lowest cost to get to the previous adjacent pixel



Parallel Implementation

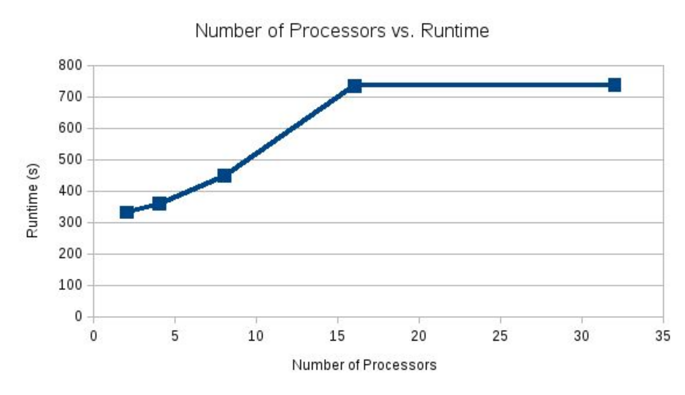
Lowest Energy Seam Component

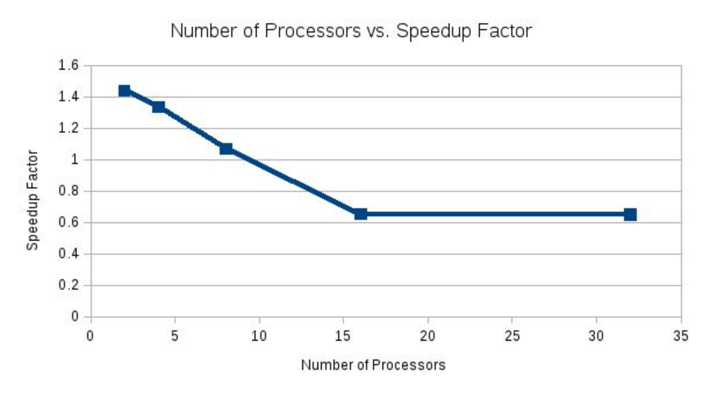
- Approach
    - Divide pixels in a given row or column evenly among processes
    - Compute lowest cost to get to each pixel from adjacent pixels
    - Communicate pixel information to processes working on adjacent portions of the image
    - Repeat for every row or column to find the lowest cost seam
    - Perform seam removal in serial

Test Results and Analysis

Energy Function Component

- Significant speedup over serial implementation for 2 to 4 processes
- Significant slowdown compared to serial implementation for more than 8 processes because of communication tradeoff

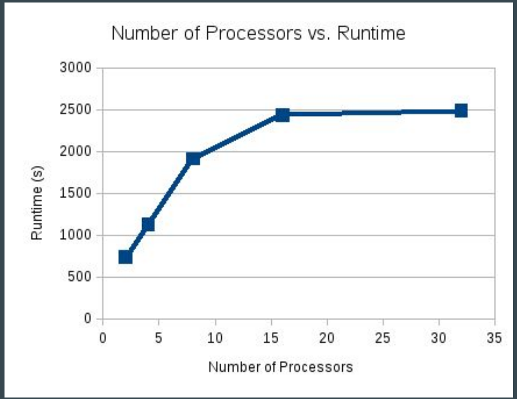


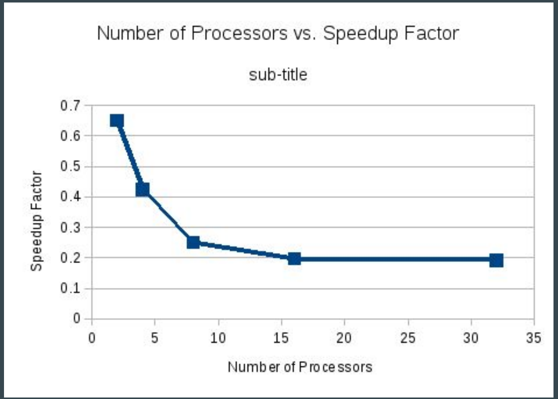


Test Results and Analysis

Lowest Energy Seam Component

- Exponential slowdown compared to serial implementation as the number of processors increases
- Communication overhead far exceeds computational gain

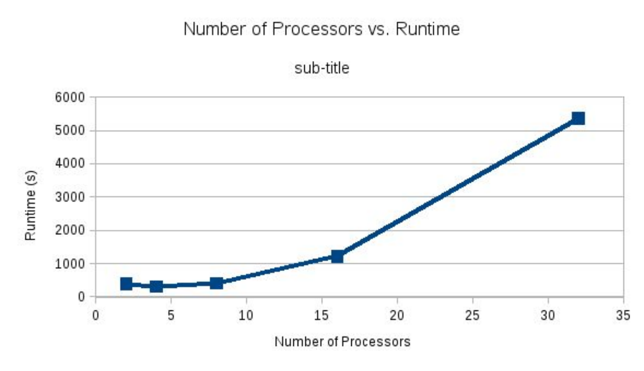


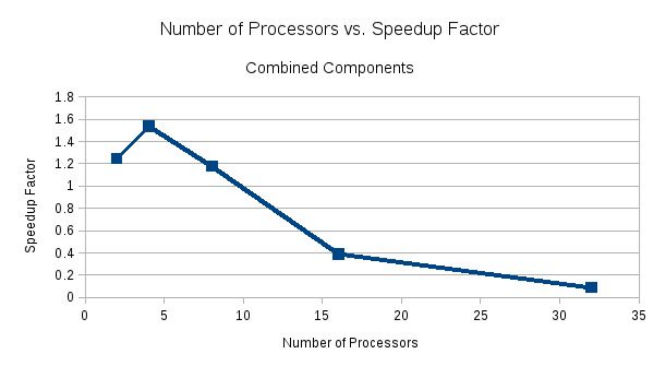


Test Results and Analysis

Combined Components

- Most significant speedup factor of 1.54 for the parallel implementation over the serial implementation for 4 processors
- Work for energy function and lowest cost seam finding computation divided
- Same communication tradeoff as number of processors increases





Acknowledgements

- Testing performed on the Bluewave cluster at the University of Maryland, Baltimore County
- CImg library used for image processing
- SL Vector library used for image processing and pixel data manipulation

Bibliography

Seam carving. (2016, January 16). Retrieved March 05, 2016, from https://en.wikipedia.org/wiki/Seam_carving

Provides a summary of seam carving, the typical algorithm, its downsides, and possible improvements. Castle example images pulled from this source.

CImg Library Project. The C++ Template Image Processing Toolkit. Project Manager David Tschumperle. Included library file for serial implementation. http://datahole.ddns.net/uploads/seam_carver/cimg.h

The CImg library used for image processing to manipulate pixel data.

Berkeley Fluid Animation & Simulation Toolkit. Primary Author James F. O'Brien. Regents of the University of California. Included library files for serial implementation. http://datahole.ddns.net/uploads/seam_carver/sl_io.h

http://datahole.ddns.net/uploads/seam_carver/sl_vector.h

Vector 3 implementation to store pixel data for energy function calculations and comparisons.

Questions