Parallel Image Processing

Jaydev Kshirsagar & Kevin Gregor

Convolution Image Filtering

- Change each pixel value to a weighted average of the pixel itself and the neighboring pixels
- Use a 2D convolution matrix (kernel) to specificy weights of all 9 pixels

Gaussian Blur

1/16	1/8	1/16
1/8	1/4	1/8
1/16	1/8	1/16

Box Filter

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Edge Detection

1/4	1/2	1/4
1/2	-3	1/2
1/4	1/2	1/4

A Good Candidate for Performance Tuning

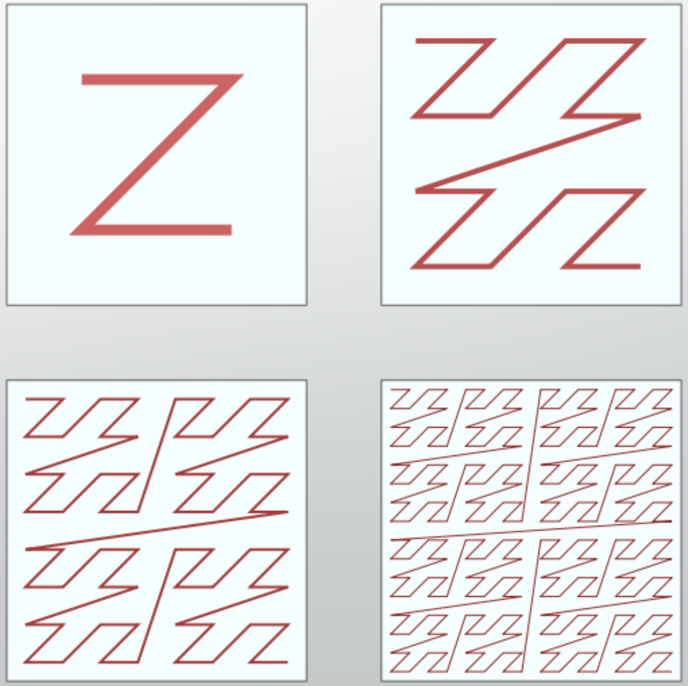
- Every pixel computation is independent of operations on other pixels
- Serial implementation could take a long time for large image sizes
- Naïve row-major memory layout is not ideal for applying 3x3 matrices

Memory Layout Adjustments

Block Linear

- Various block sizes: 8, 16, 32
- Row-major order within blocks of given size
- Blocks also stored in row-major order

Z-Morton



Memory Layout Adjustments

- Translating the input image data into these new memory layouts (and back) comes with a high cost
- There is also a cost for calculating each pixel's offset in the different layouts
- These costs could outweigh the speedup from cache hits resulting from the memory layout change

Parallelization Attempts

- **Simple parallelization of for loops that loop over image data**

```
#pragma omp parallel for collapse(2)
for (/*  iterate over height */) {
   for (/* iterate over width */) {
      process pixel
   }
}
```

Parallelization Attempts

- **OMP task-based parallelization**
  - Split image into several rectangular chunks and create an OMP task for every chunk

```
#pragma omp parallel
for (/*  iterate over chunks along height*/) {
    for (/* iterate over chunks along width */) {
        #pragma omp task
        process pixels in chunk
    }
}
```

Performance Analysis

- We observed around 2x speedup upon parallelizing the processing with task-based workload division
- We did not observe a speedup for images in Block-Linear or Z-Morton layout, but we expect a speedup as image size increases

Implementation Details

- Reading and writing images from files and argument parsing has been implemented in Python
- Image processing code is in a C++ library that is called by the Python script
- We support a variety of image formats, such as 8-,16-,32-bit grayscale images and 24-bit RGB images