

Programming Assignment 3, Due 11:59PM Nov. 7

- Purpose:
 - Synthesize the concepts you have learned so far
 - Data parallelism, locality and task parallelism
 - Image processing computation adapted from a real application
- Turn in using handin program on CADE machines
 - Handin cs4961 proj3 <file>
 - Include code + README
- Three Parts:
 1. Locality optimization (50%): Improve performance using locality optimizations only (no parallelism)
 2. Data parallelism (20%): Improve performance using locality optimizations plus data parallel constructs in OpenMP
 3. Task parallelism (30%): Code will not be faster by adding task parallelism, but you can improve time to first result. Use task parallelism in conjunction with data parallelism per my message from the previous assignment.

Here's the code

... Initialize th[i][j] = 0 ...

```
/* compute array convolution */
for(m = 0; m < IMAGE_NROWS - TEMPLATE_NROWS + 1; m++){
  for(n = 0; n < IMAGE_NCOLS - TEMPLATE_NCOLS + 1; n++){
    for(i=0; i < TEMPLATE_NROWS; i++){
      for(j=0; j < TEMPLATE_NCOLS; j++){
        if(mask[i][j] != 0) {
          th[m][n] += image[i+m][j+n];
        }
      }
    }
  }
}
/* scale array with bright count and template bias */
...
th[i][j] = th[i][j] * bc - bias;
```

Things to think about

- Beyond the assigned work, how does parallelization affect the profitability of locality optimizations?
- What happens if you make the IMAGE SIZE larger (1024x1024 or even 2048x2048)?
 - You'll need to use "unlimit stacksize" to run these.
- What happens if you repeat this experiment on a different architecture with a different memory hierarchy (in particular, smaller L2 cache)?
- How does SSE or other multimedia extensions affect performance and optimization selection?