Assignment 4: Performance Modeling: Convolution

1 Generalities

Convolution serves as a fundamental operation for a broad range of computer vision tasks. Depending on the values of a kernel, the output of convolution could make pictures to look better by removing noise as well as enabling object recognition by finding edges. Recently, the estimation of kernel values and optimizing machine learning steps have been integrated into one single algorithm called convolutional neural network which has shown a tremendous advancement on image recognition for popular applications such as Google Photos and self-driving car systems.

A convulation `C` of dimension k (k odd) is represented by a 2D array of size $k \times k$ of Single Precision floating point number. Applied on an image `in` of size $n \times m$ of Single Precision floating point number, `C` creates an image `out` of size $n \times m$ of Single Precision floating point number where:

$$out[i][j] = \sum_{x=0}^{k-1} \sum_{y=0}^{k-1} C[x][y] in[i - \left\lfloor \frac{k}{2} \right\rfloor + x][j - \left\lfloor \frac{k}{2} \right\rfloor + y]$$

A page on convolutions: `http://setosa.io/ev/image-kernels/`
Class on convolutional neural network `http://cs231n.stanford.edu/`

2 Modeling

Question: How many Flops needs to be done to compute a convolution of dimension k on a image of size $n \times m$?
Question: How much memory needs to be moved to compute a convolution of dimension k on a image of size $n \times m$?
Question: Assuming the performance numbers you measured in assignment 1 and 2, how long should computing a convolution of dimension 3 on an image of 1024x768 take?
Question: What about a convolution of dimension 11 ?

On this kind of problem, performance is usually reported in pixel processed per second.
Question: Plot maximum expected performance as a function of k.

3 Basic Code

Question: Write a basic parallel code to compute convolution of dimension k on an image of size $n \times m$. (Make sure you take n, m, and k as command line parameters.)
Question: Measure and report performance on the following size. Images of 1024×768, 2048×2048, 8192×8192, $4194304\ times 768$ and 16777216×768; and convolutions of dimension 3, 5, 7, 9, 11, 13, and 15. That is 35 combination total, report performances in pixels per second as long as expected performance.
Question: Why don't the performance always match?

4 Optimized Code

Question: Close the performance gap by writing better code.
Question: Measure and report performance of the new code.

Grades on the optimized code will be based on how many of the configuration match the expected performance.