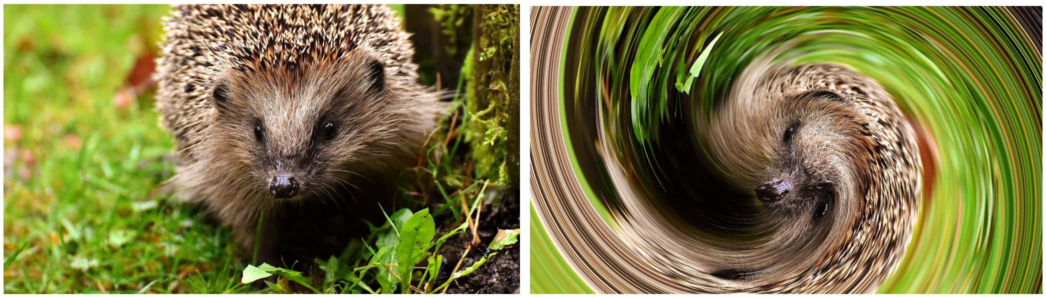
Goal

Design, implement, and compare different types of parallelism using the Intel Thread Building Blocks



Details

In this assignment, we will be using the same image distortion algorithm as Lab Assignment 1. However, instead of distorting a single image, we will be distorting a set of 20 images. Also, instead of using ISPC as a tool for parallelization, we will be relying on Intel Thread Building Blocks (TBB). Although TBB has numerous different parallel functions, constructs, containers, in this assignment we will focus only on two: TBB *parallel_for* and TBB tasks. The TBB *parallel_for* construct can be used to parallelize a loop in a SIMD fashion. This is conceptually similar to the *foreach* construct from ISPC, although it is syntactically different.

TBB tasks represent a slightly different approach to parallelism. Instead of parallelizing over a loop, any region of code can be assigned to a task. Tasks assigned to a task group are assigned to hardware threads, and can execute in parallel.

In this assignment, we will evaluate three different approaches to the image distortion of 20 images.

1. We first parallelize only the distortion algorithm using *parallel_for*. That is, image i must finish before processing image $i + 1$. However the actual processing of the image is done in parallel.

2. In this second approach, we revert the distortion algorithm to a sequential implementation. However, we now assign the reading, processing, and writing of each image to a TBB task. This should allow several separate images to be distorted in parallel.

3. In the last approach, we combine the first two approaches. That is, we assign each image to a TBB task, and within each task we process the image in parallel using the TBB *parallel_for* construct.

Steps

1. Modify the code from Lab Assignment 1 to handle multiple images. The modified application should accept a text file as a command line argument with a list of paths of images to be distorted.

   ```
   ./my_filter image_list.txt
   ```

You may use your own images, or use a directory of sample images here:

https://gitlab.com/lamb/Intro-to-Parallel/tree/master/LabAssignment2/images

2. Implement your image distortion algorithm using the TBB *parallel_for* construct. The following example from Lab3 may be useful:

   https://gitlab.com/lamb/Intro-to-Parallel/blob/master/Lab3/vectrig/vectrig_tbb.cpp

   In pseudo-code, your structure should resemble the following:

   ```
   // Read the image text file
   // For each image in image text file
     // Read image

     // Distort image in parallel
     parallel_for(...)

     // Write image
   ```

3. Next, revert the image distortion to the sequential execution, and instead parallelize across images instead of within each image using TBB tasks. Because reading, processing, and writing an image is not suitable for lambda functions, we cannot directly follow the example from lab. However, the following resource should demonstrate how to create more complex TBB tasks:

   https://software.intel.com/en-us/node/506102

   In pseudo-code, your structure should resemble the following:

   ```
   // Define a Task class
   class MyTask ... {}
     task* execute() {
       // Read image
       // Distort image sequentially
       for (...)
       // Write image
     }
   }
   // Read the image text file
   // For each image in image text file
     // Create a TBB Task
     MyTask t = ...
   ```

4. Finally, combine the parallelization schemes used in the two previous steps. That is, processes different images in parallel using TBB tasks, and within each TBB task, run the distortion algorithm in parallel using TBB *parallel_for*. The pseudo-code for this step should resemble that from Step 2, with the sequential distortion code replaced with the parallel distortion code from Step 1.

Submission

To submit this assignment, create the following directory in your git repository:

***lastname*_parallel/lab_assignment_2/**

Within this directory, please include the following:

1. A file including the three different versions of your C++ TBB implementations (parallel_for, tasking, and both).

2. A text or pdf file. Describe in your own words the three different ways the distortion of the 20 images are parallelized. Also, compare and discuss the runtimes of the three different implementations.

Assigned: January 29, 2018
Last Update: January 29, 2018
Due: ~~February ??, 2018~~

February 8, 2018