CSE 2050 - Programming in a Second Language (C++)
Programming Assignment 4

October 13, 2014

1 Due Date

- **October 24, by 11.59pm**

2 General information

- Submit only the source files (i.e., .cpp and .h files) and the associated makefile. Do not submit executables or test datasets. A single makefile should build all programs.
- Submit a single ".tar.gz" file containing the directory with all programs (and the makefile). To compress the directory, type the following command on the linux terminal:

```
tar -zcvf archive_name.tar.gz directory_to_compress
```

3 Project Summary

In this assignment, you will extend the video-processing program that you developed in Mini-Project 1. The extension consists of making the program multi-threaded. You will use Boost-library threads (not pthreads). You will also change the main storage for the video from a list to a dynamically allocated array. Specifically, the main changes to be made are:

1. Store the video in a dynamically allocated array. The array will be declared as:

   ```
   unsigned char *video = new unsigned char [width*height*numFrames];
   ```

   While `video` points to memory allocated as a one-dimensional array, you will write functions that will access the video data using indices *i*, *j*, and *k*, for row, column, and time, respectively. Here, you will write the following functions:

   ```
   void setPixelValue(unsigned int i,
                      unsigned int j,
                      unsigned int k,
                      unsigned char value);

   void getPixelValue(unsigned int i,
                      unsigned int j,
                      unsigned int k,
                      unsigned char &value);
   ```

   The dimensions of the image frames and length of the video will obtained from the video data and by reading the number of frames in the directory containing the video. All frames in the video have the same dimension.

2. The **Save-Video function** will run in a separate thread. As before, this function will save all the frames that are in memory into a directory. The video that is saved must be the current version in memory at the time the thread is created.

3. The **Compute-Average** function will also be multithreaded. Here, the function that computes the average will be a separate thread. But, this function will then divide the calculation of the mean among four separate threads. Here, the main computing function will play the role of a "manager thread" while the four threads created by the manager will be the worker threads. Upon creation, each of the four worker threads will perform the actual calculation on 1/4 of the data, independently. After creating the worker threads, the manager thread will issue a "join" so it can wait for the threads to complete their tasks, and collect the result to form a single average matrix. The diagram in Figure 1 illustrates the idea.

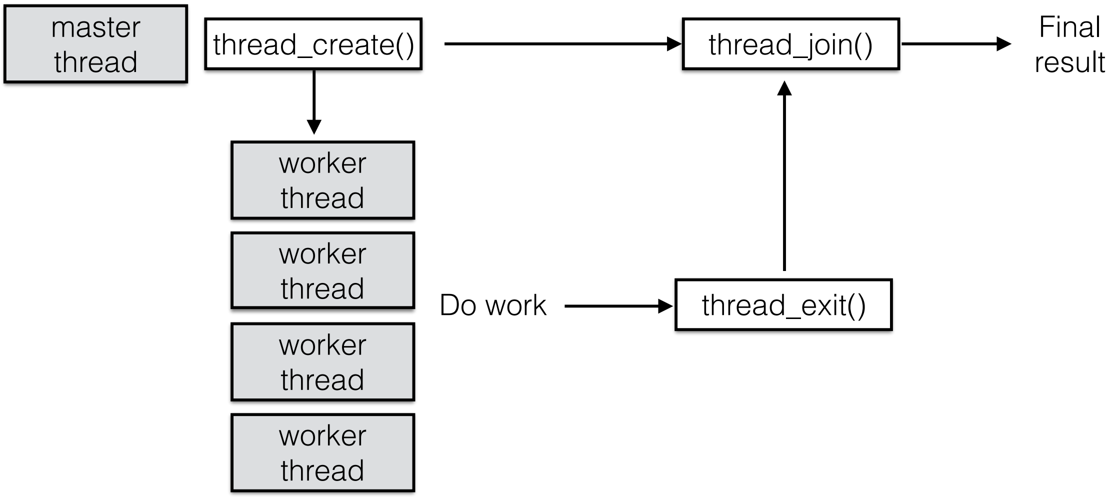

Figure 1: Master-worker thread configuration (Diagram adapted from `https://computing.llnl.gov/tutorials/pthreads/`).

Each worker thread will be given one-fourth of the video volume to perform the average calculation. These sub-arrays will not overlap and the average can be calculated without explicit collaboration among threads. As a result, there should be no major issues with synchronization, expect that the master thread must wait for the workers to finish their jobs. Figure 2 illustrate the division of the video into four equal-size sub-arrays.

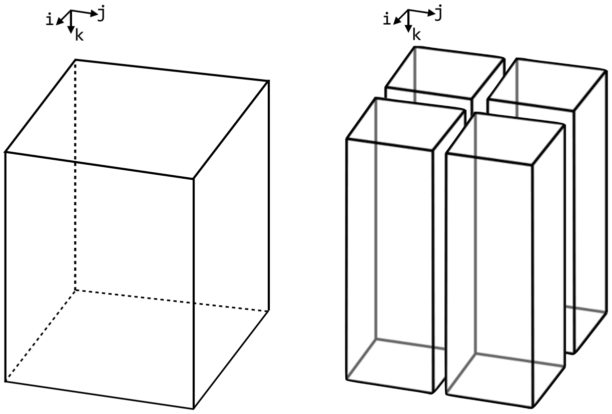

Figure 2: Video divided into four non-overlapping subarrays. Left: Original video. Right: Divided video. Each worker thread will calculate the average of a sub-array. The manager thread will combine the result and save it to an image file.

4 Background on multi-threaded programming using the Boost library

To complete this assignment successfully and learn about threads, you will need to study on you own. Please, study the following material:

- **Chapter 6: Multithreading**
  `http://en.highscore.de/cpp/boost/multithreading.html`
- Study and modify my sample program. You are welcome to use/adapt my sample code for developing your own program.