Team 1: Amber Straub, Anisha Aggarwal, Fiona Hebb, Jordan Lewis, Vinitha Gadiraju
Dr. Allen Malony
CIS 431/531
16 February 2018

Team 1 Project Proposal

Introduction

Since the early 2000s, computer hardware engineers have avoided the inherent power consumption and heat problems of high frequency single processors by moving towards a multiprocessor paradigm. Multicore processors are now commonplace, and industry trends continue to market four, eight, ten, and sixteen-core processors to consumers. The rapid speed at which the computer hardware industry is developing will continue to create processors with additional cores. Without new software to utilize the power of these processors, much of their benefit will remain unseen.

Project Outline

We will be implementing several parallel distortions over an .avi video using common patterns such as stencils and maps. This will include distortions over each visual time slice and audio time slice. The implementation will be coded in C++ utilizing OpenCV for the image reading, modification, and writing, with our choice of parallelization to be determined.

Background

Besides basic image processing, no one in our group has modified a video with a code implementation. Given this restraint learning the OpenCV library will be a learning curve. However the documentation of OpenCV implies it has everything we need to access the visual and audio data and implement a parallelization.

Modern data video processing always takes advantage of parallelism, because video files are so large it is necessary to do so. However for this project exploring the implementation of a parallel image distortion will give us a better idea of how exactly this efficiency is achieved.

Motivation

While the average user may not need the power of 16 cores, computationally complex tasks such as image/video editing and 3D graphics rendering can be completed efficiently with the use of multithreading and parallel computing patterns. By completing this project, we intend to demonstrate the benefits and facility of parallel computing, as well as to learn more about video/audio processing in a professional environment.

Plan

OpenCV's VideoCapture class offers the capability of retrieving time slices of visual data. This visual data will be concatenated into a buffer of images. Each image will be independent, thus we have the perfect opportunity to implement a parallel distortion on the buffer contents. We will be doing something similar with audio data, and then combining the data together again for our output .avi. We will be coding in C++. Some possible distortions include making the video black and white, increasing or decreasing the frequency of audio feedback, increasing video contrast, speeding up or slowing down the video, etc.

Resources

OpenCV, .avi source files, Parallel libraries: TBB, OpenMP, ISPC