Improving Image Compression via Evolutionary Computation and Graphical Processing Units

A research proposal submitted to the Office of Undergraduate Research and Scholarship

By

Brendan J. Babb
BS Computer Science Major, UAA

Faculty Mentor:
Dr. Kenrick Mock, Department of Computer Science, UAA
Abstract

Image compression is used on a day to day basis by nearly everyone. The JPEG 2000 standard is used to compress images that appear on websites, are stored on digital cameras, are displayed on cell phones, are transmitted by satellites, etc. The JPEG 2000 standard uses a wavelet for compressing and decompressing images. Previous research at UAA has been successful in evolving wavelet-like transforms that improve image compression. Recently UAA was awarded NASA funding to look develop better transforms and hyper spectral transforms. NASA money has been used to purchase Academic Matlab licenses and Toolboxes to perform research runs. This prior research used standalone computers for small, preliminary tests, but required multiple processors on the ARSC supercomputer at UAF in order to complete large scale tests. Evolutionary computation performs similar calculations across many processors; the more processors you have the faster it works. Graphical Processor Units (GPUs) also have many processors within a single card. Due to widespread use by the multi-billion gaming industry, GPUs have rapidly developed to the point where they are now capable of outperforming supercomputers for numerical calculations, by factors in excess of 100:1. It is now possible to buy a relatively cheap graphics card and insert it into an existing server to create a very fast supercomputer for performing evolutionary computation tasks in significantly less time. The goal of this project is to use GPUs and evolutionary computation to evolve significantly better wavelets for image compression.

Project Aims

Midnight is the supercomputer at ARSC in Fairbanks that I have been using. Midnight has 2280 processors, but due to it being heavily used, I can get access to about 20 processors at most, and on average about 8. Now due to funding issues, ARSC is no longer available as a resource for me. UAA recently acquired a 2 Tesla card server knows as Tesla currently. The goal of this proposal is to purchase an additional Fermi GPU for the UAA Tesla machine. The current evolutionary computation algorithms are written in Matlab. CUDA is the language used to perform operations on the GPU. Accelereyes has made a software package named Jacket which leverages Matlab to use CUDA to run operations on the GPU. This project will result in refactored Matlab code to run on the GPU. If successful this will be a proof of concept for other parallel programming research at UAA to be performed on the Tesla machine and to open this server up as a resource for other UAA research.

Introduction

Can image compression be greatly improved upon by using Evolutionary Computation and the raw computing power of Graphical Processor Units (GPUs)?

Compression of data has had huge impacts on day to day life in the last 20 years. Zip files are frequently used to compress computer files, allowing them to take up less space on a hard drive and to be downloaded more quickly. MP3s were created about 15 years ago and came in to wide use with the advent of Napster in 1999, a mere 10 years ago. By being able to compress a music file to 1/11th the size of a wave file, music became something that could be downloaded in a reasonable time. People started downloading music they had never heard of before and expanded their listening horizons. The iPod was introduced just barely 8 years ago on October 23, 2001 and has had a huge impact on our ability to carry a vast amount of music with us and how we interact with music. DVDs overtook VHS rentals in June of 2003 and make use of video compression to store movies on a single disc. Digital cameras use image compression to store many images on camera or a memory card. Digital cameras became mainstream in roughly 1994 and now most cellular phones have a built in camera. They have revolutionized the camera industry and have greatly diminished the camera film industry. Image compression is also heavily utilized in satellite, fingerprint, and medical imaging.
The current state of the art for image compression in terms of widespread use is the JPEG 2000 format. JPEG 2000 replaced the JPEG standard. Most graphical images on the web are jpgs and the format is also used for digital camera image, satellite image, fingerprint image, medical images and many other examples. The JPEG standard used the Discrete Cosine Transform (DCT) over blocks of an image along with quantization and Huffman encoding. The JPEG had a huge impact although when you heavily compressed an image you would notice a blocking effect. The JPEG 2000 standard introduced the use of a wavelet transform, followed by quantization, and entropy encoding along with other enhancements. This allowed for even greater image compression and with better looking images mostly due to the use of wavelets instead of the DCT. A wavelet is a mathematical function used to divide a given function or continuous-time signal into different scale components. A discrete wavelet transform (DWT) is used in the JPEG 2000 and is defined by a set of real number wavelet coefficients. There are many families of wavelets in use today and most wavelets are biorthogonal and energy conserving. Below is a brief diagram of how wavelet compression works.

![Image of wavelet compression diagram]

The quantization step takes the resulting data from the transform and divides by say 16 and then truncates the decimal portion. For example int(204/16)= int(12.75)=12. To dequantize the number you get 12*16=192. So the quantization step makes a smaller subset of numbers to be used for encoding, but at the expense of losing some data and introducing errors. Usually this lost data is hard for the human eye to discern or is acceptable in regards to the compression rate. The error of an image is measured as the mean square error (MSE) and is the square of the difference at each pixel of the original and lossy image divided by the total number of pixels.

Dr. Moore started studying image compression by asking if you could find better wavelet coefficients that would result in a lower MSE for an image while maintaining the same file size. The changing of the coefficients might result in breaking the biorthogonality of the wavelet and therefore making it into a wavelet like transform. Dr. Moore was able, using genetic algorithms to evolve wavelet like transforms that had a lower MSE than the original wavelet. While taking the class Genetic Algorithms in the fall of 2004, I started helping with the research as part of a class project. I continued to work on this problem with Dr. Moore and later with Dr. Peterson as well, who taught at UAA from
Evolutionary Computation is a sub-field of Computational Intelligence. Genetic Algorithms (GA) is a form of Evolutionary Computation (EC). A GA uses Darwinian methods to evolve potential solutions to a problem. You have a population of possible solutions. Crossover occurs by choosing two individual solutions as the parents and they mate and have two child solutions where each child has part of their values from both parents. There is also a chance for mutation where a solution will have one of its values slightly changed. A fitness is determined for each possible solution and the fitter candidates have a better chance of advancing and mating in the next generation. In our case, the fitness is a lower mean squared error for the compressed image. Gradually from generation to generation solutions become more and more fit and non-fit solutions die out. From this method we are able to evolve better transforms that have less error than the original wavelet. For each member of the population the fitness consists of using the possible solution and performing image compression using that solution and then reconstructing the compressed image and comparing it to the original. If your population is 300 individuals, then you basically compress and reconstruct an image 300 times with each solution for each generation. And you typically run for 200-500 generation. This process is successful but involves a massive amount of computation. Since a wavelet might have 32 coefficients you are searching a 32 dimensional space where the each coefficient can be any real number, which is a huge search space. Because of the intense but highly parallel computation required, we have been using the supercomputer at UAF which has 2280 processors to run our GA experiments. Due to the heavy use of the supercomputer from scientists around the world, we are only able to grab at most 20 processors and on average 8 processors at a time. This has still resulted in tests that once took 4 days to run to now only take 18 hours.

Graphical Processing Units (GPUs) are used to accelerate graphic rendering on computers. Due to the multibillion gaming industry, GPUs have improved at a greater speed than the Central Processing Units (CPUs) over the last 10 years. While recent advances in CPUs have included multi-cores, allowing two or four processors on one chip, GPUs currently have 240 processors on one chip. GPUs are tailor made for graphic calculations but recently have started to be used for general computing thanks to languages like CUDA from NVIDIA. CUDA is a C like language that handles some of the low level plumbing of programming a GPU. General Programming on GPUs (GPGPU) has just started to be used by researchers to do parallel programming in the last year or two. NVIDIA has announced that their next chip, code named Fermi, will have 512 processors and have additional features aimed at performing science research with GPUs. GPU cards are also much cheaper in dollars/processor than CPUs.

While at the 2008 Genetic and Evolutionary Computation conference (GECCO) I heard a talk from a student in London who had just started using GPUs for performing a GA on a desktop computer. I chatted with him over the rest of the conference and he explained several of the principles of programming in CUDA. In 2009 GECCO had a workshop dedicated to EC on GPUs. Our work is in Matlab which allows high level changes in code for possible experiments and exploring new hypothesis. Matlab is also utilized on the supercomputer at UAF. We have had great success with evolving wavelet like transforms that outperform the original wavelet on standard images, fingerprint images and satellite images. These results were used in numerous conference and journal papers. Genetic Algorithms take a massive amount of computation. Genetic Algorithms thrive on parallel processing, essentially performing the same calculations on 1000 element population. The algorithms are known as “ridiculously parallel” algorithms. If you had a 1000 processors each generation of a population could be calculated in one computation cycle all at once. Within the last year as we explore more complex problems we have noticed that we have had smaller improvements as we require much more computation. Accelereyes released a product called Jacket that is an interface from MATLAB to CUDA, allowing slight changes in MATLAB code to leverage computing on the GPU.
Project Design

With the advent of cheap graphics cards people can now improve existing servers very quickly by adding improved graphic cards. This research project will consist of creating of adding additional cards to Tesla and then rewriting existing MATLAB code to utilize the GPU to accelerate GA experiments.

The first step involves purchasing an Academic license for Jacket and an additional 2 licenses to use 3 total cards in Tesla. The second step is to acquire an NVIDIA GTX 560TI with a Fermi chip and 384 cores. The third step involves rewriting the MATLAB code to utilize Jacket and possibly external CUDA programs to speed up both the evaluation of the fitness function by performing the Discrete Wavelet Transform and also using the GA in parallel across a potential 864 processors. This will be the bulk of the work of the project, but recently Jacket released version 1.7 which handles conv2, but will require rewriting of additional code to maximize the speed up. I hope to achieve a 20 fold increase in speed according to Amdahl’s law.

The fourth step involves running experiments for evolving wavelets. The following experiments will be investigated:

1. Evolve ICER integer wavelets for use in NASA image compression currently used on the Mars Rovers.
2. Evolve ICER real value wavelets for use in NASA image compression currently used on the Mars Rovers.
4. Create a landscape map of the fitness for the wavelet. This will allow mathematical insight on how we evolve gains.
5. Evolving the forward and reverse 9/7 wavelet with MRA 3 and quantization 64.

The beauty of evolving forward transforms is that you can modify the JPEG 2000 application to use the new coefficients and compress images using this new method that take up less space but can be decompressed by the standard JPEG 2000 application. So you could use the new method and create images you could e-mail to anyone and they could still open them using their existing software. Each experiment will consist of 20 runs of the Genetic Algorithm with a statistical analysis of the average improvement over the original wavelet via MSE reduction or file size reduction.
Anticipated Results

Our previous work has yielded improvements of 7-30% MSE reduction on evolving both the forward and the reverse transforms. The larger wavelets have seen less improvement because the search space becomes so huge. With the GPU speed up of the experiments I hope to see:

1. Forward transforms that reduce file size by 3-5% while keeping the MSE the same. This improvement would be significant for the FBI who stores over 81 million people's fingerprints using the 9/7 wavelet.
2. Forward and Reverse transforms that reduce MSE by 30% while keeping the file size the same.
3. Evolved wavelets that look very different from known wavelets. Most of our evolved wavelets have looked similar to the original wavelet. We have discovered two new wavelets that looked nothing like the original. New wavelets would help us better understand the landscape of possible solutions that might lead to a mathematical algorithm for constructing better wavelets.
5. The use of GPUs in MATLAB for performing distributed Genetic Algorithms.

These results will contribute to papers for the SPIE 2012 conference in Orlando in April as well as several papers for the GECCO 2011 conference in Dublin, Ireland in July. Hopefully the cumulative research along with prior papers will lead to a new journal article. The best case scenario is that an evolved wavelet is so much better that it produces the successor to the JPEG 2000 standard, perhaps the JUAA.

The resulting built up Tesla can serve as a replacement for the ARSC supercomputer when it comes to parallel programming. I have taken Dr. Mock’s Computer Assembly class with and emphasis on CUDA GPU programming which was very helpful for my research. I predict that in 6 months to a years MATLAB and Mathematica will have evolutionary computing toolboxes that seamlessly leverage GPUs without requiring you to rewrite your code. I am excited by the potential of an 18 year old student having their own personal supercomputer or access to Tesla and the new research and inventions they can come up with.

I was reading a research paper published on November 26th, 2009 from MIT entitled, A High-Throughput Screening Approach to Discovering Good Forms of Biologically-Inspired Visual Representation. In the article they use GPUs to speed up experiments on visual object recognition in images. They discovered vision methods that beat the currently known best methods. They achieve a speed up of 222 times for their experiments and get the results used in the paper in one week on a Sony Playstation using a GPU. They estimate that it would have taken them 2 years to get the same results using the current CPU method. The use of GPUs not only allows you to perform experiments much more quickly but allows you to see that an experiment is heading in the wrong direction and stop it and try a different approach. It also encourages researchers to explore more complex problems that would have not been feasible in terms of time or the cost of supercomputer time to run the experiment. This is where I see the real potential for new discoveries. Evolutionary computation paired with personal supercomputers will revolutionize research and potential innovation.
Project References