Final Report: Exploration of the Utilization of Compressed Sensing in Image Compression

Sameer Lal

May 18th, 2013

1 Introduction

1.1 Motivation

Over the past decade, a new phenomenon known as compressed sampling has been developed to improve on the classical Nyquist-Shannon Sampling Theorem. Unlike the classical theorem, which states that the sampling frequency must be at least twice the maximum frequency present in the signal [1], compressed sampling leverages the sparsity of signals to enable sampling at much lower rates. The motivation of this project is using compressed sampling to compress images at improved levels.

Building on top of the work done by Jason Halpern in the Fall of 2012 (on the implementation of compressive sampling algorithms across the cloud [2]), the aim is to improve compressive sampling implementations by overcoming their implausible running times during the image reconstruction phase. Through parallelization of these implementations, efficient compression of signals can occur; enabling a large decrease in bandwidth across networks.

1.2 Project Tasks

In order to use compressed sampling for image compression, a process for converting images to signals must be developed. Though Halpern recommended using the Wavelet Transform, other methods of signal processing are explored in order to find a method for constructing optimal signals for compressed sampling. The second task is developing an interface for these signals to be sampled and reconstructed with the compressed sampling implementation known as CoSaMP. The final task is to run the CoSaMP implementation across the Hadoop Cluster in parallel to increase efficiency of the algorithm.

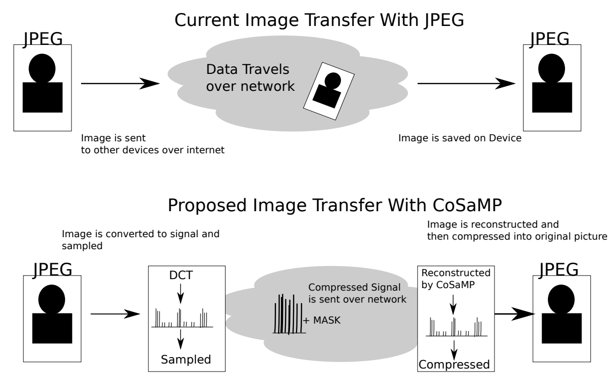

Figure 1: Diagram of data flow of traditional JPEG transfer vs. proposed CoSaMP transfer.

2 Midterm Findings

During the first half of the project, the following findings were made to construct the proposed architecture of data flow in Fig. 3.

2.1 Signal Processing Technique

In implementing compressed sensing for image compression, the first task was deciding on a method that would convert images to signals. Various signal processing techniques were evaluated. Among which the Wavelet Transform and the highly popular Discrete Cosine Transform were the two most feasible techniques. Upon further analysis of compressed sampling it was determined that sparsity was the important factor when constructing the signals. With increased sparsity, the sampling rate, and thus the amount of data necessary, decreased. Based on experimentation of both types of transformations with a set of pictures, it was discovered that the Discrete Cosine Transform would serve the purposes of the implementation better and was thus selected as the signal processing technique.

2.2 Compressed Sensing Algorithm and its Parallelization

The planned compressed sampling algorithm that would be used for signal reconstruction on the server cluster was the Java Implementation of the CoSaMP program written by Halpern in the Fall of 2012. The algorithm itself is a greedy algorithm that iteratively guesses the reconstructed signal. Because of this construction, parallelization from a higher level was deemed implausible. Instead, through experimentation of the CoSaMP implementation it was discovered that a specific subsection involving matrix multiplication and sorting consumed the most amount of time. Thus future steps would require finding a way to run this subsection in parallel, specifically the matrix multiplication and the data sort.

3 Parallelization of CoSaMP

3.1 Running Subsections in Parallel

Since the complexity of matrix multiplication is of cubic nature and even improved algorithms have not broken the boundary of quadratic complexity, the approach taken to parallelization was to break up the larger matrix (original signal) into smaller subsections. By dividing up the matrix and running the CoSaMP in parallel on the multiple subsections, the complexity of the algorithm would effectively decrease significantly due to simultaneously reconstructing parts of the signal where each part is of the most efficient signal subset to reconstruct. To explore this approach, experimentation to find the optimal set was conducted.

3.2 Experimentation on the CoSaMP

Several parameters were explored to understand the nature of the CoSaMP algorithm. The length or size of the signal was the first parameter. The second parameter was the sparsity of the signal that was constructed. The third signal was the arrangement of the coefficients within the signal, whether they were randomly distributed or placed at the beginning of the signal. The final parameter was the actual sampling rate. Two factors were measured while running the CoSaMP reconstruction on signals constucted with varying properties: the time taken to run the algorithm and the accuracy of the reconstructed signal as match to the original signal.

3.3 Results

Since naturally with the implementation of the Discrete Cosine Transform the picture is divided into blocks of size 64, signals with lengths that were multiples of 64 were constructed. For each length, the signal had varying sparsity, sampling rate, and both arrangements of the significant coefficients were tested (randomly distributed and set at the front). The results of the tests showed that having randomly distributed significant coefficients had resulted in much greater precision of the reconstructed signal. Naturally the time complexity of the algorithm decreased significantly with the smaller signal lengths. The accuracy of the reconstructed signal was alarmingly poor as the size of the signal decreased. Upon testing 100 different signals, of length 64 (1 DCT block) to 4096 (64 DCT blocks) the average of the sum of differences between the reconstructed signal and the original signal was 30.4 with a standard deviation of 12.5. In regards to the important relationship between the sparsity of the signal and the sampling rate, the CoSaMP algorithm required especially high sampling rates to accomplish signal reconstruction. Furthermore, in situations where the sampling rate was equal to the size of the signal, the reconstructed signal still did not match the original signal. Thus the sparsity of the signal was a driving factor in the effectiveness of the CoSaMP. The results of the experiment showed that only until the sparsity reached levels of .96 and sometimes even greater, was the CoSaMP efficient in reconstructing the signal.

3.4 Feasability of CoSaMP in Signal Reconstruction for Images

The results of the experimentation of the CoSaMP bring into question whether the algorithm, or even Compressed Sensing is feasible for the reconstruction of signals that reconstruct images. To attempt parallelization by reconstructing different subsections of the signal the sparsity of each subsection would have to be above a certain threshold. Due to the random nature of the coefficient space and the content of pictures themselves, it does not seem feasible. Although it parallelization of the algorithm at the step that requiring matrix multiplication and data sort may be something to consider. The cost of the algorithm with respect to the accuracy it provides must be evaluated. The results of the experiments showed that even with great amounts of sparsity, the sampling rate was much higher than anticipated, and for most pictures will not improve current compression standards.

4 Theory of Current Image Compression

Due to the findings regarding the CoSaMP the theory of current image compression as well as compressed sensing were further analyzed in order to understand whether there was a situation in which compressed sensing could be an improvement to image compression standards.

4.1 Signal Processing and Transformations

The most powerful aspect of current image compression techniques is the signal processing stage in that it allows for other schemes, such as encoding, to be effective. Through convolution over the size of the image, the sum of certain basis functions can be transformed into making up the picture. Initially in the older JPEG version, this was utilized via the Discrete Cosine Transform. The basis of this transformation was the cosine function. In the more recent JPEG scheme, signal processing is done through the Wavelet Transform, which uses wavelets (functions that represent a burst of energy) as its basis. In both situations, the transformation allows for the identification of repetative data within the two dimensional space.

4.2 Encoding

The next step in the compression process is the encoding of these significant coefficients. Encoding allows for repetative sections of this coefficient space to be reduced by large factors of compression; this includes insignificant coefficients (zeros). This aspect of the compression scheme is the most effective part of the process in reducing the amount of data. In the planned scheme of using CoSaMP to subsample the Discrete Cosine coefficient space to send a smaller signal, the "direct competition" with regards to compression, is this encoding step.

4.3 The Implications of Theory on the Project

Based on this theory it is proven that by current methods of compressed sensing (those that only exploit the sparsity of signals) there exists no situation in which a picture with M significant coefficients, can be sampled at a rate less than compression standards. Mathematically: at the very least, there must be M*log(M) measurements with compressed sensing (this was derived via Professor Arian Maleki). Taking the worst case, signals can be described as a simple set of pairs of magnitudes and their locations. This data (without any encoding or other compression schemes) is 2M. At this base level, signals with very small size may be expressed with less data via compressed sensing. But taking into account the usual size of signals, and considering that the data will undergo some compression scheme, theoretically Compressed Sensing cannot express signals with less data than current compression schemes. In fact, several individuals who were involved with the development of Compressed Sensing technology are now working on trying to use compression schemes to improve compressed sensing algorithms.

5 Compressed Sensing Technology

5.1 Measurements

After gaining an understanding of the Compressed Sensing technology, it was discovered that the fundamental gains that are to be made regard measurements. At its core, Compressed Sensing is not a compression scheme, rather a mechanism for high levels of sub sampling in situations that entail large amounts of sparsity. This positions the technology for effectiveness in scenarios that involve costly measurements. At the measurement level, compressed sensing has shown impressive feats, including using a single pixel camera and reducing the time required for an MRI. Thus use of the technology seems to be best when implementing the technology by altering sensors to provide reasonable original data while making significantly less measurements.

5.2 Future Research Potential: Exploration of Subsampling Applications

With this thinking in mind, there still remains great research potential in the field. The first avenue of research may be attempting to integrate compressed sensing and current image compression techniques rather than trying to overcome one with the other. From a theoretical standpoint, perhaps loss-less image compression schemes could be implemented on subsampled images, thus essentially reducing the information that would usually be compressed. It should be noted though, that the gains of the compression will not be the sampling-rate, but rather will depend on how the data of the signal is constructed. To do this efficiently, specific masking schemes must be developed. This type of research will be less involved in Compressed Sensing, but rather determining the optimal method of subsampling for signal compression.

Another avenue for research is actually working with specific sensors to improve current standards using subsampling/compressed sensing. Attempting to integrate the compressed sensing technology into video and depth sensors could yield some interesting and highly applicable results.

6 A New Approach

6.1 Parallelization and Signal Reconstruction

Although perhaps the exercise of using compressed sensing in image compression was less productive than anticipated in improving compression levels, the methodology by which the CoSaMP was approached can be used for signal reconstruction. The new research question becomes: Can we exploit parallel architectures in signal reconstruction?

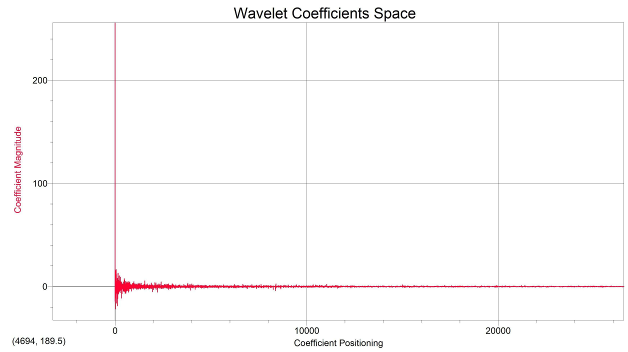

Figure 2: Wavelet Coefficient Space for a Single Picture

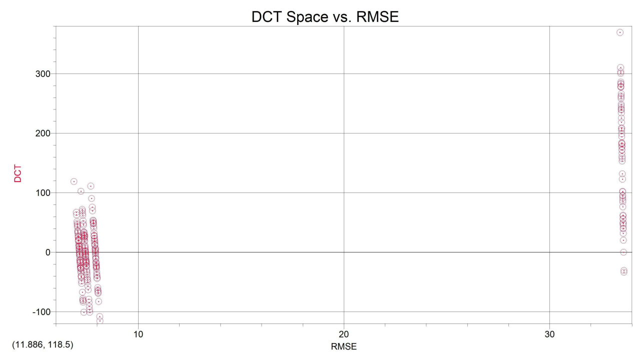

Figure 3: Discrete Cosine Coefficient Space vs. RMSE of the Data

6.2 Wavelet Coefficient Space

One of the transforms mentioned in the signal processing step was the Wavelet Transform. The benefit of this transform is that the significant coefficients are positioned at the beginning of the picture (As shown in Fig. 2). Interestingly, these coefficients take up the nature of a dampened sine wave. Due to the shape of the coefficient space, it may be applicable to apply the DCT to this coefficient space.

6.3 Initial Experimentation and Results

The initial experimentation show that by shifting certain coefficients, the coefficient space can be significantly reduced in terms of standard deviation as well as the average coefficient value (needs to be closest to zero). Fig. 3 shows how the subset of a signal changes in the coefficient space when certain coefficients are changed to better fit some basis functions.

6.4 Algorithm

The next goal is to use parallel architectures to further explore optimal data sets for describing a coefficient space. This will involve a ranking system of sorts, that will explore which coefficients should be changed first. Furthermore, the set of possible partitions should also be explored, some of the data may fit very well for specific partitions while excluding other data. Ultimately an algorithm can be developed that can find the optimal compression for the data.

References

[1] E. Candes and M. Wakin. An introduction to compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):21–30, Mar. 2008.

[2] J. Halpern. Final report: Implementing compressive sampling algorithms across the cloud, Dec. 2012. `http://www.cs.columbia.edu/~msz/projects/2012-Fall-CS/final_report.pdf`.