Transmitting, Compressing, and Denoising Images

André Shannon, Nathan Lyell, Maggie Dong, Steven Zhang, and Bill Shen
Supervised by Dr. William Ross

Math and CS Department

Transmitting data

Abstract

During transmission, data can get corrupted through a noisy channel. Usually that data is beyond recovery, but the corruption could be mitigated through the use of frames and over-sending the data.

Method

- Simulate sending data without frames and getting corrupted
  - Corrupted a percent of the data using normal distribution
- Simulate sending data through frames and getting corrupted
  - Created parseval frame using the first rows of the Fourier Discrete Cosine Transform (DCT) Matrix size N
  - $fc = \sum_{i=1}^{N} \langle d_i, F_i \rangle F_i$
  - Where d is the data, F is the frame, N is the length of the data, and fc is the frame coefficients
  - Corrupt a percent of the frame coefficients using normal distribution
  - $\hat{d} = \sum_{i=1}^{N} \langle \widehat{fc}, F_i \rangle F_i$
- Compare the two

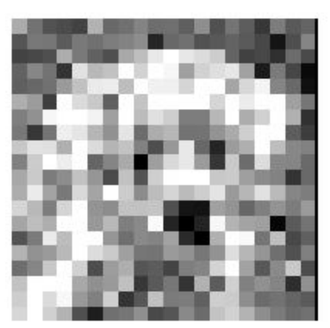
Results

Images were used for the data, each pixel in a grayscale image is represented by a number, and those were put into an array.

Original Image (20 x 20)

C1: Corruption without a frame Error: 4.2356

C2: Corruption with a frame Error: 3.3337



The error was calculated using the norm of the difference between the corrupt image and the original, so the smaller, the better.

Discussion

C1 may appear better to the eye than C2, but when given an error score C2 scores better. This is probably due to the stark differences of a few pixels in C1 being more than the small differences of more pixels in C2.

Compressing data

Abstract

Another prominent issue in IT is limited storage. Data compression helps to store as little information as possible while still having full access to the data.

Method

- Chop the data into smaller chunks for faster calculations (Figure 1)
- Calculate the Fourier coefficients for each data chunk
  - Using the Discrete Cosine Transform (DCT) matrix
- Choose a percent of those coefficients to not store
  - Typically taking the high frequency values, which are are numerically small
- Perform the inverse DCT to see the compressed data result
- Contrast the original data with the compressed result

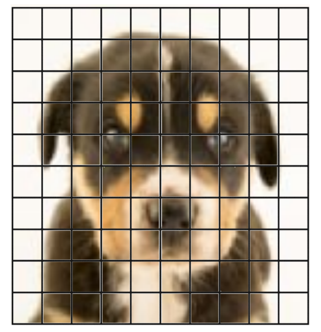


Figure 1: Partitioned Image

Results

For color images, each pixel has three values associated with it: one for each of the major color channels: red, green, and blue. In processing color images, we dealt with each color channel independently to simplify the calculations.

Original Image (100 x 100)

Compressed Image (using only 80% of coefficients)

Error score: 46.53

The image was chopped into 10x10 pixel chunks and each chunk compressed separately. At the end, these chunks were pieced back together to form the new compressed image.

Discussion

So as this shows, we are capable of storing much less data while still retaining an accurate representation of the data.

Denoising data

Abstract

If data is corrupted by noise, there is a way to "denoise" or mitigate its effects through the use of wavelets.

Method

- Take the discrete wavelet transform of the image or data
  - This yields a matrix of 4 boxes, one of averages and three of horizontal, vertical, and diagonal differences (Figure 2).
- Threshold the discrete wavelet transform
  - There are different thresholds, but they generally change the diagonal differences so that the small differences become 0 (Figure 3).
- Take the inverse wavelet transform of the thresholded data

Figure 2: Wavelet transform of image

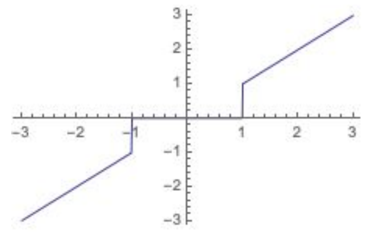


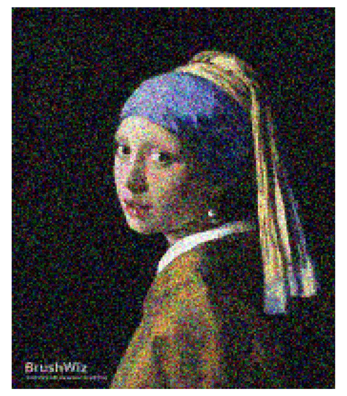
Figure 3: The "hard" threshold function

Results

Original Image

Corrupted

Denoised



Discussion

The denoised image is still left with noise, but it has reduced the amount from the corrupted one without altering the underlying image much.

References

Han, Deguang. *Frames for Undergraduates*. American Mathematical Society, 2007.

Kovacevic, Jelena, and Markus Puschel. "Sampling Theorem Associated with the Discrete Cosine Transform." Carnegie Mellon University, 2006, jelena.ece.cmu.edu/repository/conferences/06_05_Icassp_Sampling.pdf.

Yu, Guoshen, and Guillermo Sapiro. "DCT Image Denoising: a Simple and Effective Image Denoising Algorithm." *Image Processing On Line*, vol. 1, 2011, doi:10.5201/ipol.2011.ys-dct.