Hardware Decompression for Video Compressive Sensing Applications

Embedded Systems Final Project

Members
Frank Zovko
Keith Dronson
Samuel Subbarao
Federico Garcia

Background
Compressive sensing algorithms are currently being used to both capture and save still photos. The most efficient algorithms out there can take seconds to 1 minute or so in order to complete. However, these are done in software. We would like to hard code the algorithm so that we can decrease the compression time by a factor of 10 or 100. That way we can compress video instead of just still photos.

Standard compression/imaging schemes work on the following principle [1]:
Sample at the Nyquist rate (2x the bandwidth), get N samples. A non-linear compression is performed and you get K samples (K << N). This can then be stored or transmitted. However, the input was K-sparse in some basis psi to begin with.

Compressive sensing works by directly acquiring compressed data. Instead of taking N samples, collect M samples: K < M << N. Mathematically these M samples come from some matrix phi (dimension MxN) being multiplied by the signal x (with dimension Nx1). M can be a random matrix (white Gaussian noise). The result is a matrix y with dimension Mx1. To recover the signal following this algorithm:

Find the 'active' component by seeing which column of phi is most correlated with y.
Subtract that active component off of y, leaving you with y'
Repeat with y'

We will need to talk to Prof. Castro to work out the details of the algorithm.

Preliminary Goals
Connect to a video camera and get pictures at 30 fps from the video camera to the FPGA.
Compressively sense the data (in the real world this would be unnecessary since the data would already be coming in compressed by a compressive sensing imager, but we're using a standard CCD here).
Save it to an SD card
Once all the video is taken, decompress the video and display it on a computer monitor.

Reference
1. http://www.ece.rice.edu/~richb/talks/cs-tutorial-ITA-feb08-complete.pdf