Phase Unwrapping On Reconfigurable Hardware

Sherman Braganza, Miriam Leeser
Northeastern University
Boston, MA
{sbraganz, mel}@coe.neu.edu

In this paper we present a Field Programmable Gate Array (FPGA) implementation of the core computation of the minimum L^P norm phase unwrapping algorithm. This computation involves a Discrete Cosine Transform (DCT) of up to 1024 points and represents the largest DCT/iDCT implemented on an FPGA documented in the literature.

Introduction

There exist several applications that make use of coherent signals for imaging purposes. Coherent signals contain information about both magnitude and phase as opposed to incoherent ones that just contain magnitude information. Applications utilizing such signals include Synthetic Aperture Radar (SAR), Magnetic Resonance Imaging (MRI), optical interferometry and adaptive beamforming. Such applications often have a reference signal to which the received signal is compared (a stable local oscillator located in the radar unit in the case of SAR) and from that comparison the phase is extracted. However, this extraction is limited by the fact that the output phase will lie between π and $-\pi$. Hence, the raw output phase is referred to as *wrapped*.

Given a noise free signal, the original phase can be recovered by accumulating the phase difference and an integer multiple of 2π every time a discontinuity is detected. In the presence of noise however, such an unwrapping can fail catastrophically in the 2 dimensional case.

Phase Unwrapping

A set of methods has evolved as a solution to this problem, discussed and analyzed in [1]. They can be roughly classified into two groups: those that unwrap around noisy sections and those that use numerical error minimization techniques. One of the methods that consistently produces high quality results is the minimum L^P norm technique. This iterative technique, which falls under the class of error minimization algorithms, operates by first making an initial guess as to the solution and then iteratively refining it. It uses the minimum P (i.e. P=0) since that allows the final data to match the measured data in as many places as possible (if P were set to 2, this would be the equivalent of a least squares minimization). Convergence of the algorithm is tested by checking for residues, else if it is not achieved the algorithm terminates after a preset number of iterations. A residue occurs when a closed loop integral on the two dimensional image data does not produce a zero, but instead produces either 2π or -2π. The presence of residues indicates that naïve unwrapping via accumulation cannot be performed.

A simplified version of the phase unwrapping algorithm is:

Initialize solution p

For (k=0 to MAX_ITER)

> *Compute residues*
>
> *If no residues, exit loop*
>
> *Compute weights and weighted phase Laplacian c*
>
> *Subtract weighted phase Laplacian of the current solution p from the one just computed c*
>
> *Solve Qp=c using Conjugate Gradient letting p be the initial guess.*

End for

A sample of the results produced by the minimum L^P norm phase unwrap is displayed below in Figure 1.

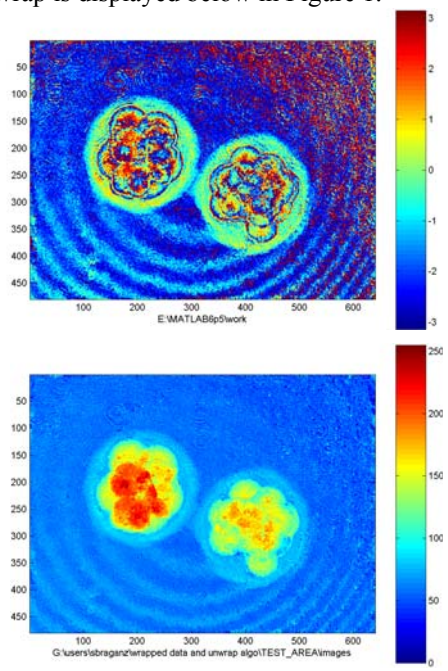

Figure 1: Wrapped phase above and unwrapped phase below. The image is of an embryo and was taken using Optical Quadrature Microscopy (OQM), an interference based microscopy method.

The conjugate gradient solver makes use of a two dimensional Discrete Cosine Transform. This transform was found, through timing profiles, to occupy over 80 percent of the total processing time and was thus the ideal candidate for implementation in hardware.

Implementation

The two dimensional DCT is *separable*, so it can be decomposed into one dimensional transforms that convert

first the rows, and then the columns of the input data. By making use of this property, the hardware implementation can be broken into two sections: one that does the transform and another that selects and organizes the input/output. The work presented in this paper focuses on the transform section.

The one dimensional DCT can be viewed as the real part of the Fourier transform of the even extension of an input signal. Makhoul [2] presented a method that solves a size N DCT using a size N/2 complex FFT. We use an adaptation of his method to perform DCTs using size N FFTs. The full optimal reduction is not performed in order to minimize the number of multiplications that must be performed due to accuracy considerations.

The algorithm can be viewed as a series of three steps, which are performed in reverse order for the inverse transform. The first step is a shuffle, which re-orders input data. This is the equivalent of extracting the odd elements from *x(n)* and mirroring them around the N/2 point. The second step is to perform the FFT. This transform component must be capable of handling one dimensional transforms of both M and N (where M, N are the dimensions of the input image matrix) and performing both forward and inverse transforms. Finally, the last step is to multiply the result by $2e^{\frac{-j\pi k}{2N}}$ which results in the equivalent of a forward DCT.

This construction lends itself to a clear component breakdown, i.e. a shuffle component, an FFT component and a rebuilding component. The shuffle component reorders input according to the equation above and writes it in shuffled order to BRAM.

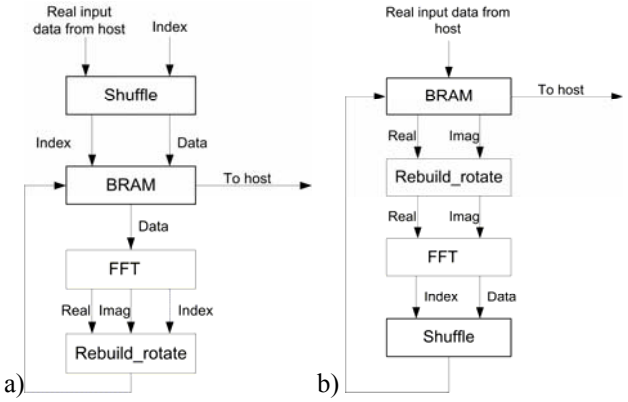

Figure 1: Dataflow diagram for the forward (a) and inverse (b) transforms

This is then transformed using the Xilinx FFT core [3] and the result is multiplied by $2e^{\frac{-j\pi k}{2N}}$ which is generated using a lookup table.

The inverse transform moves data in the other direction, starting of with a multiplication by $2e^{\frac{j\pi k}{2N}}$ followed by an IFFT and then an inverse shuffle that maps *v(n)* to *x(n)*, i.e the opposite of Equation 1. A more complete analysis can be found in [4].

Results And Conclusions

The various components and controller discussed above were implemented on a PCI-based Annapolis Wildstar II Pro. The largest component in terms of both size and latency is the FFT component so the performance of the entire transform is highly dependent on the particular FFT implementation chosen, thus designers can take advantage of high performance FFT cores. The other components provide a fixed overhead of fourteen cycles independent of the transform size, one from the shuffle, nine from the rebuild_rotate unit and the rest from inter-component registering.

The DCT component operates at over 100 MHz and occupies 32% of a single V2P70. This means that two DCT components can be put onto a single FPGA, thus increasing parallelism. This implementation has a significantly larger latency, but consumes much less area when compared to traditional matrix multiplication and Distributed Arithmetic (DA) approaches used to perform DCTs on FPGAs. However, such approaches are impractical for large transform sizes since the area requirement becomes prohibitive. For our design, latency is dependent on the size of the transform, but for 1024 point data, it takes 26us including full data transfer times into the DCT component from BlockRAM.

Currently, a two dimensional transform is done by decomposing the image into rows and columns in software and then each row is streamed into the FPGA. The next part of this project to be completed is to copy the entire image to onboard SRAM and then control the row/column decomposition in hardware, thereby minimizing data transfer overhead from host to board.

Acknowledgements

This work was supported in part by CenSSIS, the Center for Subsurface Sensing and Imaging Systems, under the Engineering Research Centers Program of the National Science Foundation (award number EEC-9986821)

References

[1] D. C. Ghiglia and M. D. Pritt. *Two-Dimensional Phase Unwrapping: Theory, Algorithms and Software*. Wiley Inter-Science, 605 Third Avenue, New York, NY, 10158-0012,1998.

[2] J. Makhoul. "A fast cosine transform in one and two dimensions". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(1):27–34, February 1980.

[3] Xilinx Inc. Fast Fourier Transform 3.2. http://www.xilinx.com/ipcenter/catalog/logicore/docs/xfft.pdf, Last accessed March 2007.

[4] Sherman Braganza and Miriam Leeser. "The 1D Discrete Cosine Transform For Large Point Sizes Implemented On Reconfigurable Hardware", *IEEE conference on Application specific, Architectures and Processors*, 2007.