Google Summer of Code 2013 OpenCV Student Application Proposal

I am prashant iyengar presently pursuing Master in Electronics and communication from IIT Bombay (Indian institute of Technology Bombay).

I am research assistant in Vision and Image processing lab and my research interests include development of hand gesture recognition interface for mobile application and development of vision based application on mobile devices.

I have prior experience of working in the software industry for 5 years and have very good programming skills in C/C++/java.I am extremely passionate about coding and image processing

GSOC provides me with a unique opportunity to expand my skills and share knowledge.I am extremely committed and passionate about programming and developing image processing and vision based application.I sincerely hope you consider my application as i think i posses the necessary skill and temperament to participate in the program.

I am submitting the proposal for 2 projects OpenCL optimization and Wavelet Transform module for OpenCV.I would like to work on one of these projects as part of GSOC Project.

I have attached description and timeline of each project, provided links to sample code and associated documents i have developed over course of my study,and attached my resume with the following proposal.

If any of proposals cannot be accepted as part of GOSC i would still be interested in working on them if there is interest in integrating them as part of OpenCV.

1>Project Title :OpenCL optimization

Reason For Choosing Project:

I began learning OpenCL as tool to implement fast image processing algorithms .I am aware of the basic of OpenCL programming and have developed a few samples applications to test the capabilities of OpenCL application on Multi-Core CPU's.

I am extremely eager to learn efficient techniques to develop image processing and computer vision algorithm using parallel programming techniques.This will enable me to understand the design methodology of developing parallel processing algorithms to solve different types of problems and share the same using this platform.

Code Samples

Below are links to code samples and documents i have developed using OpenCV and OpenCL for image processing.The programs were implemented to understand the basics of OpenCL programming and to find a way to interface OpenCL-OpenCV.The programs include BGR2GRAY,BGR2HSV color conversion,1D convolution,2D convolution,2D separable convolution,Box filter,Gaussian filter

Documents

1. http://pi-virtualworld.blogspot.com/2013/02/opencl-heterogeneous-parallel-program.html
2. http://pi-virtualworld.blogspot.com/2013/02/opencl-2d-convolution-using-separable.html
3. http://pi-virtualworld.blogspot.com/2013/02/hetrogenous-parallel-programming-for.html
4. http://pi-virtualworld.blogspot.com/2013/01/opencl-heterogeneous-parallel.html

Code

1. https://github.com/pi19404/m19404/tree/master/OpenCL-Image-Processing/ColorConversion
2. https://github.com/pi19404/m19404/tree/master/OpenCL-Image-Processing/Convolution

Introduction :

The OpenCV OCL module contains a set of classes and functions that implement and accelerate select openCV functionality on OpenCL compatible devices.

Many image processing and computer vision algorithms can be implemented by using parallel programming techniques and OpenCL provides a framework for achieving this.

The OpenCV OCL module includes utility functions, low-level vision primitives,and high-level algorithms. The utility functions and low-level primitives provide a powerful infrastructure for developing fast vision algorithms taking advantage of OCL whereas the high-level functionality includes some commonly used image processing algorithms and state-of-the-art algorithms ready to be used by the application developers.

The aim of the OpenCV Optimization would be to add to high-level functionality that can be readily used by the OpenCV application developer.Detailed step by step tutorials would be provided for the modules developed to enable users to learn and contribute towards the development of OpenCL algorithms in the future.

Description:

The proposed components described below include color conversion routines,thresholding operations,contrast enhancement,frequency analysis,feature detection and extraction,optical flow computation etc.The algorithms proposed are selected based on requirement,computational complexity,whether it exists in the GPU modules ,and feasibility of OCL modules providing significant improvement standard CPU algorithm.

i have tried to select algorithms of different types and tried to arrange them in order of increasing difficulty.This would enable in learning to design of develop better algorithms as project progress forward.

All the proposed components will be tested using standard CPU based OpenCV algorithms and using OCL OpenCV Module and performance improvement can be evaluated.

Associated with the OpenCL program a tutorial/document describing the algorithm and basic OpenCL performance optimization used in the algorithm.

Proposed Components for OCL Module

1)Color conversion

Description

The current modules have only a few color conversion routines.So we can add some important color conversion routines.As part of GSOC may be demonstrate one of color psace conversion like RGB2HSV and HSV2RGB,RGB2Lab,Lab2RGB etc with detailed tutorials .This will enable users to contribute additional routines at later stage.

References :

a> The existing OpenCL code or GPU can be used as reference to implement the mentioned color conversion routines

2>Binary thresholding :OTSU Binarization ,Adaptive Thresholding

Description :

Currently only binary thresholding is incorporated in OCL modules.We can add automatic thresholding techniques like OTSU thresholding and Adaptive thresholding techniques.

Approach for OTSU Binarization: The histogram computation and binarization can be implemented in parallel.The inputs for histogram computation is the input grayscale image.The input for the applying the threshold is input grayscale image and threshold.

After computation of histogram the computation of inclass and within class variance for each threshold value can also be implemented using OpenCL.

The computation of optimum as maximum values of computed weighted variance can be computed on using parallel minMax routine or this can be done on the CPU as it may be faster.

Approach for Adaptive Thresholding :

The lookup table computation for binary thresholding be performed on the CPU and passed as input to OpenCL algorithm.
The first step is to compute the local mean using gaussian or box filter this can be implemented using OpenCL
Finally the thresholding operations for each pixel can be performed using OpenCL

References :

a)Brij Mohan Singh, Rahul Sharma, Ankush Mittal and Debashish Ghosh. Article: Parallel Implementation of Otsu's Binarization Approach on GPU. International Journal of Computer Applications 32(2):16-21, October 2011. Published by Foundation of Computer Science,

b>Adding support for color images and data types for convolution operations

3>Increasing support for convolution

Description :
convolution currently supports CV_32FC1 data type only.The support for generic convolution operation can be increased to support CV_8UC images and multi channel images.

Approach :

The multichannel support can be added by either considering a 3D block/grid for processing or modifying the existing code so that each thread process all the channels.

Separate algorithm can be implemented for 8 bit data types which would be more efficient than converting the image to floating type and then performing the operation

References :

a)The Existing code in the GPU module or OCL modules can be used as reference to implement the above functionality.

4>FFT - 2D FFT,and inverse FFT,Spectrum multiplication

Description :
A parallel implementation of performing 2D FFT and IFFT and spectrum multiplication which would be useful for performing frequency analysis

References:

a)The 2D DFT,IDFT and spectrum multiplication exists in the GPU modules.This can be used as reference to develope the above modules.

5>Histogram backprojection

Description :
histogram computation is part of OCL module we can add histogram backprojection module .
Approach :
The input to the algorithm are the input image and computed histogram.This would a look table

operations using histogram for all the pixels and possible scaling operations.These can be implemented using OpenCL.

6>Mean Shift and Cam Shift Tracking,Weighted Mean Shift Tracking

MeanShift And CamShift Tracking Description :

The components required for Mean Shift and Camshift tracking are histogram computation,histogram backproject,moment calculation etc.These components either already exist in OCL module or are developed as part of present project.Hence the mean shift and camshift tracking algorithm can be pipelined and implemented in parallel

Since moment calculation would not be over a large section of the image this could performed on the CPU.

References :

a)Jun Zhang; Shuhua Luo; Xianru Liu, "Weighted Mean Shift Object Tracking Implemented on GPU for Embedded Sustems," *Control Engineering and Communication Technology (ICCECT), 2012 International Conference on* , vol., no., pp.982,985, 7-9 Dec. 2012
b)D.; Bruns, E.; Kurz, D.; Grundhofer, A.; Bimber, O., "Fast and robust CAMShift tracking," *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on* , vol., no., pp.9,16, 13-18 June 2010

Weighted Mean Shift Tracking Description :

A variant of mean shift tracking can be implemented which given more importance to pixels near the center pixel to target area while computation of location which provides the maximum similarity.This can be implemented using OpenCL algorithm and provide better performance than traditional mean shift tracking

Reference :

a)Zun Zhang; Shuhua Luo; Xianru Liu, "Weighted Mean Shift Object Tracking Implemented on GPU for Embedded Sustems," *Control Engineering and Communication Technology (ICCECT), 2012 International Conference on* , vol., no., pp.982,985, 7-9 Dec. 2012

7>Contrast Stretching ,CLAHE,Modified Contrast Stretching

Description :

Global contrast stretching and CLAHE can be implemented to add image processing capabilities of OpenCL module.

Approach :

The input to the algorithm for global contrast stretching are input image and the lookup table for transformation.contrast stretching can be implemented in parallel as lookup table operations.

For CLAHE the operation like computation of histogram lookup table for subregions of image, applying the lookup table,clipping the histogram ,interpolation can be implemented using OpenCL

References :

a)CUDA implementation CLAHE is present in the GPU module .This can be used as reference to develope the above module

Modified Contrast Stretching :

The standard contrast stretching performs simple pixel-wise affine transform mapping the input minimum and maximum measured pixel values .The input min and maximum values are determined using histogram of image ignoring less significant pixels at the tails of the histogram.This provides more robustness to noise.

The histogram and CDF computation can be performed in parallel,
the input minimum and maximum values for transformation value can be performed on the CPU.
The lookup table computation and applying the threshold can be performed using OpenCL

Reference :

a)Nicolas Limare, Jose-Luis Lisani, Jean-Michel Morel, Ana Belén Petro, and Catalina Sbert, "Simplest Color Balance,"Image Processing On Linevol. 2011.http://dx.doi.org/10.5201/ipol.2011.llmps-scb

8>Feature Detection :Good Feature to Track,Fast Corner detection and multiscale fast feature detection

Description

Fast Feature detection ,Multiscale Fast Feature detection and Good Features to track feature detector can be incorporated into OCL module in addition to existing harris corner feature detection

Approach :

The computation of first order derivatives ,The elements of matrix used by harris corner response function can be evaluated at each pixel location of the image. Gaussian blurring or box filter can be used next to compute average over the neighbourhood. All these operations can be efficiently performed using OpenCL.

Performing the non maxima suppression and selecting the N most significant feature can be performed on the CPU it may provide better performance on CPU or a parallel implementation of non maxima suppression can be used

For the Fast and MultiScale Fast Corner detection

The computation of image pyramids for multiscale approach,The feature detection and harris corner response can be implemented using OpenCL

The non maxima suppression and selecting the N most significant feature can be performed on the GPU as it is more suited to CPU as it may provide better performance on CPU

References:

a)GPU Module has the implementation of good feature to track and fast corner feature detection and multiscale fast used in ORB feature detection.This can be used as reference .
b)Fast Feature Detection with a Graphic Processing Unit Implementation
Charles Bibby and Ian Reid,Dept of Engineering Science,University of Oxford
c)Accelerated Corner-Detector Algorithms,Lucas Teixeira, Waldemar Celes and Marcelo Gattass .Tecgraf - Computer Science Department, PUC-Rio, Brazil
d)**http://www.sunsetlakesoftware.com/2012/02/12/introducing-gpuimage-framework**

9>ORB features and FREAK feature descriptors

Description :

OCL module does not contain any feature descriptors towards that end a parallel implementation of ORB and FREAK and descriptors using OpenCL .

Approach :

The input to the algorithm is input image and keypoints detected by feature detection algorithm.

FREAK descriptors

First component required is integral image.The computation of mean intensity about the keypoint descriptor .The estimation of keypoint orientation and extraction of descriptors at estimated orientation and be implemented in parallel by launching a kernel thread for each keypoint.The integral image representation of image is already present in OCL module.

ORB descriptors

the GPU module contains feature for routine for ORB descriptors.This can be used as reference to develope the present component.

References:

a)GPU module ORB descriptor in OpenCV

10) Optical Flow Algorithms

Description

The LK optical flow tracking is present as part of OCL modules.A algorithm for dense optical flow computation would be useful.A parallel implementation for Farneback Algorithm and Dual TV L1” optical flow algorithm can be developed using OpenCL

References :

a)The farneback algorithm is present in the GPU module.This can be used as reference to develope the OpenCL algorithm.

b)The Dual TV L1 optical flow algorithm paper contains some description about parallel implementation and GPU module also contains implementation of algorithm which can eb used as reference.

11>Image Stitching Pipeline

The image stitching pipeline consists of components like image re-sizing,finding features,matching features,wrap images,blend images which can be implemented using parallel processing algorithms. OCL already has few of these components.Hence paralyzing the image stitching pipeline would enable us to attain significant performance improvement.

TimeLine :

- Taking into consideration the design and development plan of the project average of 35 hours per week would be devoted to the project.
- The study of algorithms and design activities would be completed before the date for starting coding.Before the mid-term deadline relatively simple algorithms will be explored in order to get through understanding of OpenCL programming and anticipate any issue that may be faced in developing more difficult algorithms.About 8 components will be developed before the midterm deadline for the project
- After the mid term deadline development of more difficult algorithms will be taken up along with testing,debugging and documentation of all the components.
- The plan looks to incorporate quite a few algorithms,most of the algorithms taken up before the mid-term evaluation are relatively simple and purpose is to gain experience in OpenCL coding so that issues in more complex algorithm can be anticipated and tackled easily.

- After the mid-term evaluation the feature descriptors,optical flow and stitching pipeline are taken up.The detailed study of CPU version of algorithms and design plan will be prepared for OpenCL implementation before the start of coding.
- The timeline for these algorithms may seen sort but they reflect only development efforts as design is completed before the start of coding.

Below is tentative plan for the project.

Before Coding begins on June 17

- Understand the OpenCV OCL interface ,CPU implementation of different algorithms being implemented by May 27.
- Develop design document for the algorithms to be implemented in proposed modules. : June 10
- A high level design of algorithms being implemented will be prepared for discussion with mentors and standardizing the interface as per OpenCV standards
- Finalizing the algorithms and interface and clarification of other doubts with mentors by June 17

After Coding Begins :June 17th

- Development,debugging and basic testing of 2 components per week
- Testing and documentation for 1 of components per week
- This approach can be followed for components 1-8
- The aim is to develop the first 8 component ,perform testing and documentation for at-least 4 components by midterm evaluation deadline July 27.

After Midterm Evaluation Deadline :After July 27th

- Development,testing and debugging of at-least 1 component per week
- Testing and documentation of 2 components per week
- This approach can be followed for components 9-10.These components contain 4 algorithms .The aim is to complete development of 4 components and testing documentation of at-least 8 components by Aug 20:

After Aug 20:

- Work on development,debugging and testing of component .Code clean up and standardization and other changes
- Work on completing the testing and debugging of remaining components
- The aim is to have the all development of all the components completed by Sep 1st

After September 1st

- Through testing of all the modules,debugging,bug fixes etc
- Completion of documentation
- The aim is to complete final testing all the modules and prepare the code for evaluation by Sep 16th

After September 16

- Minor changes ,documentation ,clean up code if required

2>Title :Wavelet Transform Module for OpenCV

Reason For Choosing Project:

I had taken a postgraduate course on wavelet transform and multi-resolution analysis.I had worked on 3D MRI data compression as part of 1 month project and ever since i have been interested in the wavelet transform and its applications .

Incorporating new component like wavelet transform functionality in OpenCV will provides an unique opportunity to learn ,design and develop fast and efficient algorithms from scratch.

I expect the project to be exciting as well as challenging and i think i posses the necessary temperament and skill to work on the project.

Code Samples

Link for document on small project on 3D MRI data compression using wavelet transform using Matlab

https://github.com/pi19404/m19404/blob/master/wavelet/3D_Wavelet_Compression-a2.pdf

Below is sample programs for implementation of 1D and 2D wavelet decomposition and reconstruction using filter banks in C/C++ using OpenCV .

https://github.com/pi19404/m19404/blob/master/wavelet/

odwt.cpp ,odwt.h are files containing code for filterbank implementation for 1D and 2D wavelets main.cpp contains samples code for performing wavelet decomposition and reconstruction for haar wavelets.

Introduction

Wavelet transform is a tool for multi-resolution time-frequency analysis.
The discrete wavelet transform has a huge number of applications in engineering,mathematics,computer science,data representation,data compression,computer vision,image processing,time series analysis etc.Some of the application of wavelet transform in image processing and computer vision are image compression,image denoising,multi resolution representation,feature extraction,fingerprint detection,image watermarking. The present modules focuses on development of core modules for 1D and 2D wavelet transform and demonstrate some application based on wavelet transform.

Proposed Modules

- The aim of proposed module is to provide an efficient framework for computation of wavelet transform and demonstrate applications using the same.
- The proposed modules contain modules for 1D and 2D wavelet analysis and synthesis.
- Some simple applications like denoising,compression are developed to demonstrate the use of wavelet transform.

I have summarized some of components that could be developed the basic core components for wavelet transforms and some applications.

The significant contribution would be choosing efficient algorithms ,optimization of these components.After optimization of standard serial version of algorithms SSE optimization etc would be developed for the for improving the efficiency of algorithms.

1>Wavelet Decomposition and Reconstruction filter coefficients.

The coefficients for approximation and detail coefficients for reconstruction and decomposition filters Haar,Daub,Symlets,Coiflets,bi-orthogonal etc family of wavelets can be defined and stored in the files.

During the DWT or IDWT computation the decomposition and reconstruction high pass and low pass filter coefficients can be read from the file/database according to the wavelet family.

This approach provides flexibility of adding other wavelet families without changing the source code.

2>1D Wavelet Decomposition and Reconstruction

This component will perform wavelet decomposition and reconstruction.
The algorithms will use the standard method of performing scaling and downsampling.
This will be standard version of the algorithm without SSE optimizations.

a)Multi scale 1D discrete wavelet decomposition

The multiscale 1D discrete wavelet decomposition uses 1D DWT computation unit as a basic module.

The 1D DWT modules will perform 1D wavelet decomposition along the columns of the input data matrix by performing 1D convolution along the columns and the performing dyadic down sampling operation.

The output will be 2 matrices/vector corresponding to approximation and detail coefficient for each column of the input matrix

For a multi-scale wavelet decomposition The input to algorithm is the signal ,number of level and wavelet family.The output will be a vector of matrices consisting of approximation and detail components at various levels.The multi-level discrete wavelet transform is computed by calling recursively calling the function of single level wavelet decomposition function.

b)Multi Scale 1D Wavelet reconstruction

The wavelet decomposition and reconstruction algorithm is based on the concept of perfect reconstruction filter bank as conjugate quadrature filters or conjugate mirror filter bank.

The basic module required for reconstruction is a single level 1D wavelet reconstruction .

The multi-scale wavelet reconstruction is performed by calling the function for 1D wavelet reconstruction recursively.

References:

1. Mallat, S. G., "A theory for multiresolution signal decomposition: The wavelet representation," IEEE Transactions on Pattern Recognition and Machine Intelligence, Vol. 11, No. 7, 1989, 674-693.

c)Visualization of filter impulse response of wavelet scaling and approximation function

This component is used to compute the impulse response of wavelet scaling and approximation using iterated filter bank scheme for visualization.Based on the type of wavelet (orthogonal or bi-orthogonal etc) the wavelet family or used defined reconstruction filters different schemes can be used.

References:

- Wavelets and filter banks: Theory and design (1992)by M Vetterli, C Herley

d)Multi Level Wavelet transform Decomposition and reconstruction using lifting scheme

This component will look at implementation of wavelet decomposition and reconstruction using lifting scheme.This provide a computationally efficient methods for computation and performance evaluations can be performed in comparison the basic algorithm to determine the benefits.
This algorithm can be easily pipelined and parallelized.

References:

1. Factoring wavelet transforms into lifting steps by Ingrid Daubechies and Wim Sweldens
2. Wim Sweldens. The Lifting Scheme: A Construction Of Second Generation Wavelets

3. Building Your Own Wavelets At Home by Wim Sweldens and Peter Schröder

e)Optimizations for 1D discrete wavelet transform

This components will look at techniques for parallel implementation of discrete wavelet transform decomposition and reconstruction using SIMD intrinsics (SSE,SSE2).Efficiency can be achieved by considering parallelization and pipelining the approach using standard perfect reconstruction filter bank or using the approach using lifting scheme.
The comparison can be performed for both approaches and optimum algorithm can be selected based generic criteria on factors like wavelet family,wavelet type,filter length etc

References

1. Onchis, D.; Marta, C., "Multiple 1D data parallel wavelet transform," Symbolic and Numeric Algorithms for Scientific Computing, 2005. SYNASC 2005. Seventh International Symposium
2. Chakrabarti, C.; Vishwanath, M., "Efficient realizations of the discrete and continuous wavelet transforms: from single chip implementations to mappings on SIMD array computers," Signal Processing, IEEE Transactions on , vol.43, no.3, pp.759,771, Mar 1999.
3. Judith M. Ford, Ke Chen and Neville J. Ford - "Flexible paralelization of fast wavelet transform", Joint Work UMIST, 2001
4. O. Rioul and P. Duhamel "Fast algorithms for discrete and continuous wavelet transforms", IEEE Trans. Inform. Theory, vol. 38, no. 2, pp.569 -586 1992
5. G. Sullivan Vector and parallel implementations of the wavelet transform, 1991
6. SIMD Implementation of the Discrete Wavelet Transform by Jake Adriaens Diana Palsetia
7. Performance Comparison of SIMD Implementations of the Discrete Wavelet Transform
8. SIMD Parallelization of Common Wavelet Filters Rade Kutil , Peter Eder, Markus Watzl

2>2D Wavelet Transform

a)2D wavelet decomposition

The basic unit of performing 2d wavelet decomposition is single level 2D wavelet decomposition unit.

This component will perform a single level wavelet decomposition of a 2D matrix.The input is the 2D image matrix and wavelet family while the output is vector of matrix corresponding to approximation (LL) and detail components (LH,HL,HH).

The 2D wavelet transform is computed by first applying the 1D analysis filters to the columns of 2D data matrix followed by downsampling and then to the rows followed by downsampling along the rows.

The multi scale wavelet decomposition is performed by iteratively passing the LL component of previous scale input to the single level wavelet decomposition unit

b)2D wavelet reconstruction

The basic component of performing 2D wavelet reconstruction is single level 2D wavelet reconstruction.

Single level 2D wavelet transform is computed by upsampling and filtering the rows and then columns of the approximation and details coefficient by respective reconstruction filter.

This reconstructed filter of the previous level/scale is selected as LL or approximation coefficient of next lower scale during reconstruction.

The wavelet decomposition and reconstruction are implemented as 2D 4 channel perfect reconstruction filter bank.

References:

1. Mallat, S. G., "A theory for multiresolution signal decomposition: The wavelet representation," IEEE Transactions on Pattern Recognition and Machine Intelligence, Vol. 11, No. 7, 1989, 674-693.
2. Ikehara, M.; Nguyen, T.Q., "On 2D perfect reconstruction linear phase filter banks," Signals, Systems & Computers, 1997. Conference Record of the Thirty-First Asilomar Conference on , vol.1, no., pp.721,725 vol.1, 2-5 Nov. 1997

c>Visualization of impulse response of scaling and approximation filter

This component computes the impulse response of scaling and approximation filters for 2D 4 channel perfect reconstruction filter bank.This is implemented using concept of iterated filter banks.

References:

1. Wavelets and filter banks: Theory and design (1992)by M Vetterli, C Herley

d)2D wavelet transform using lifting scheme

This component the 2D wavelet transform is computed using lifting scheme.Also performance comparison can be performed with filter bank implementation

References :

1. Ben Hnia Gazzah, I.; Souani, C.; Besbes, K., "DSP implementation and performances evaluation of 1D and 2D DWT using the lifting scheme," *Design and Test Workshop, 2008. IDT 2008. 3rd International* , vol., no., pp.166,172, 20-22 Dec. 2008
2. A Survey on Lifting-based Discrete Wavelet Transform Architectures TINKU ACHARYA AND CHAITALI CHAKRABARTI Department of Electrical Engineering, Arizona State University, Tempe, Arizona 85287-5706
3. Yan-Kui Sun, "A two-dimensional lifting scheme of integer wavelet transform for lossless image compression," *Image Processing, 2004. ICIP '04. 2004 International Conference on* , vol.1, no., pp.497,500 Vol. 1, 24-27 Oct. 2004
4. Parallel Implementation of the 2D Discrete Wavelet Transform on Graphics Processing Units: Filter Bank versus Lifting
5. Implementation and Comparison of the 5/3 Lifting 2D Discrete Wavelet
6. Transform Computation Schedules on FPGAs by MARIA E. ANGELOPOULOU AND PETER Y. K. CHEUNG

e)optimization for 2D wavelets

This component will look at optimizations for reconstruction and decomposition for 2D wavelet transform using parallel processing techniques using SIMD primitives (SSE,SSE2 etc).

The optimized version for filter bank implementation and lifting schemes can be developed and performance can be evaluated based on generic criteria like wavelet family,filter length etc.

References :

1. Parallel Implementation of the 2D Discrete Wavelet Transform on Graphics Processing Units: Filter Bank versus Lifting
2. Masselos, K.; Andreopoulos, Y.; Stouraitis, T., "Execution time comparison of lifting-based 2D wavelet transforms implementations on a VLIW DSP," *Circuits and*

Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on , vol., no., pp.4 pp.,926, 21-24 May 2006
3. Franco, J.; Bernabe, G.; Fernandez, J.; Acacio, M.E., "A Parallel Implementation of the 2D Wavelet Transform Using CUDA," *Parallel, Distributed and Network-based Processing, 2009 17th Euromicro International Conference on* , vol., no., pp.111,118, 18-20 Feb. 2009doi: 10.1109/PDP.2009.40
4. Jinook Song; In-Cheol Park, "Implementation of efficient architecture of two-dimensional discrete wavelet transform," *SoC Design Conference, 2008. ISOCC '08. International* , vol.03, no., pp.III-60,III-61, 24-25 Nov. 2008
5. SIMD Implementation of the Discrete Wavelet Transform by Jake Adriaens and Diana Palsetia
6. Vectorization of the 2D Wavelet Lifting Transform using SIMD extensions C. Tenllado, D. Chaver, L. Piñuel, M. Prieto and F. Tirado

Applications Components:

a)Displaying the approximation at detail coefficient at various levels
Different approaches to thresholding can be demonstrated from crude thresholding methods to automatic methods can be demonstrated.

The wavelet decomposition provides approximation and detail coefficients at various levels using multi-resolution/scale decomposition.

The approximation coefficient contain the average image information while detail coefficient contains
information about edges etc.

The application would display the approximation and detail coefficients at different levels.

b)Image approximation at specified level

The decomposition at level N provides details coefficient at all levels and approximation coefficient at the lowest level.

As we decompose the wavelet coefficient the details coefficients become insignificant at successive levels.

The application would also allow the user to reconstruct approximation at level N. it would ignore the detail coefficient at level N and perform the reconstruction.

c)Wavelet coefficient thresholding

A common strategy for compression and denoising application is thresholding the wavelet coefficients.Though the selection of thresholds may vary for different application.
This application will demonstrate the soft and hard thresholding .
The user can specify the level dependent threshold and view the reconstructed image after thresholding

d)Threshold Selection for image denoising

This will demonstrate various algorithms for threshold selection like universal threshold,visushrink,sureshrink,Bayes shrink

e)Simple image compression scheme

This will demonstrate a simple image compression scheme by thresholding,quantization,encoding the coefficients.
The threshold selection and application can be performed using methods described above.
A simple quantization scheme to convert the coefficients to integer representation is used
The encoding can be performed using entropy encoder,RLE or huffman coding.The compression ratio is computed to determine the performance.

References:

1.) Maarten Jansen. Noise Reduction by Wavelet Thresholding, volume 161. Springer Verlag, United States of America, 1 edition, 2001.
2. Martin Vetterli S Grace Chang, Bin Yu. Adap-tive wavelet thresholding for image denoising and compression. IEEE Transactions on Image Pro-cessing, 9(9):1532–1546, Sep 2000. Carl Taswell.
3. The what, how and why of waveletshrinkage denoising. Computing in Science and Engineering, pages 12–19, May/June 2000.
4. David L Donoho. De-noising by soft threshold-ing. IEEE Transactions on Information Theory, 41(3):613–627, May 1995.
5. Adaptive Wavelet Thresholding for Image Denoising and Compression S. Grace Chang, Student Member, IEEE, Bin Yu, Senior Member, IEEE, and Martin Vetterli, Fellow, IEEE

TimeLine :

Taking into consideration the design and development plan of the project average of 35 hours per week would be devoted to the project.

The basic study of concepts and code optimization etc will be performed before the application is accepted (May 27th).

After the approval of application detailed study of algorithms and optimization techniques will be performed and high level design document will be prepared before the start of coding deadline.

The plan is provided on weekly basis .Before the mid term evaluation 1D wavelet transform would be completed.This will enable in learning implementation specifics,anticipating issues and provide a good foundation to develop algorithms for 2D wavelet transform.Hence sufficient time is provided for the development of 1D wavelet transform modules.

After completion of 1D wavelet transform 2D wavelet transforms will be taken up,The learning from 1D case should enable faster development and debugging of 2D module.
Some sample application demonstrating the use of wavelet transforms are also taken up.

Below is tentative plan for the project.

Before Coding begins :Before June 10

a)Understand in the details the concepts required for implementation ie filter bank implementation,lifting scheme,SSE optimization basics : Before May 27
b)Develop design document for the algorithms to be implemented in proposed modules. : June 10
* A high level design of algorithms being implemented.
* Details about the functions ,input/output arguments

* Test Plan
c)Discussion with mentors about algorithms for algorithms and standardizing
the interface as per OpenCV standards by June 10
d)Finalizing the algorithms and interface and clarification of other doubts with mentors by June 17

After Coding Begins :June 17th

Week 1

a)Interface for retrieving the decomposition and reconstruction filter components from database file.
b)1D wavelet transform using filter bank implementation

Week 2 : Lifting Scheme and Optimization using SIMD intrinsics (SSE,SSE2)

a)1D wavelet transform using lifting scheme
b)Optimizing the filter bank implementation

Week 3 : Optimization using SIMD intrinsics (SSE,SSE2) and basic testing

a)Optimizing the lifting scheme implementation
b)Basic testing of filter bank implementation and performance statistics

Week 4:Testing and 2D wavelet transform

a)Testing of 1D optimized wavelet transform using lifting scheme and performance statistics
b)Filter bank implementation of 2D wavelet transform(without optimization)

The aim is to complete the development and basic testing of 1D wavelet transform before the completed for mid-term evaluation deadline.

After Midterm Evaluation Deadline : July 27th - Sep 1

Week 1

a)2D wavelet transform using lifting scheme
b)Testing of filter bank implementation and performance statistics
c)Optimization for filter bank implementation of 2D wavelet transform

Week 2

a)Testing of lifting scheme and performance statistics
b)Optimization for lifting scheme for 2D wavelet transform
c)Testing of optimized filter bank implementation and performance statistics

Week 3

a)Testing of optimized lifting scheme implementation for 2D wavelet transform

Week 4

a)development and testing Application components a,b,c mentioned in project proposal
c)Testing and performance statistics of all the components

week 5

a)Application component d and e development and testing
b)Testing of all the components .

The aim is to complete the 2D wavelet transform testing ,optimization and performance analysis and develop some basic applications.

After Sep 1:

1>Work on development,debugging and testing of component .Code clean up and standardization and other changes
2>Through testing of all the modules,debugging,bug fixes etc
3>Completion of documentation

The aim is to complete final testing all the modules and prepare the code for evaluation by Sep 16th

After September 16

Minor changes ,documentation ,clean up code if required