ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)

On 21st & 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

A Comparative Study on Mat lab And Its Open Source Alternative Scilab for Image Fusion Application

Jothi. M[#1], B. Chandra[#2],R.Dheepthe [*3], Krishnaveni.P [*4]

(ASP- Assistant Professor), Department of IT, K.L.N .College of Engineering, Madurai. India.
(ASP- Assistant Professor), Department of IT, K.L.N .College of Engineering, Madurai. India.
UG Scholar, Department of IT, K.L.N .College of Engineering, Madurai, India.
UG Scholar, Department of IT, K.L.N .College of Engineering, Madurai, India.

Abstract— Image processing has been used in wider application. To process the images, Mat lab has become one of the most inevitable tool. But to use the MATLAB effectively the original software is required. This problem can be overridden by Open source software like SciLab. This paper gives the overview of scilab and it variants relevant to mat lab for a case study. We proposed image fusion application as a case study. This paper will contain short and crisp about the image processing with mat lab and the fusion techniques. The results were discussed with its sample output. This paper is organized as follows: It starts with an introduction, basic filtering process and followed by an enhancement. Last section deals with the fusion of images using mat lab.

Keywords— Image filtering, Image enhancement, Image fusion, Scilab

I. INTRODUCTION

Image processing is a technique in which the data from an image are digitized and various mathematical operations are applied to the data, generally with a digital computer, in order to create an enhanced image that is more useful or pleasing to a human observer, or to perform some of the interpretation and recognition tasks usually performed by humans which is known as picture processing. [1] Image Processing is a technique to enhance raw images received from cameras/sensors placed on satellites, space probes and aircrafts or pictures taken in normal day-to-day life for various applications. Various techniques have been developed in Image Processing during the last four to five decades. Most of the techniques are developed for enhancing images obtained from unmanned spacecraft, space probes and military reconnaissance flights. Image Processing systems are becoming popular due to easy availability of powerful personnel computers, large size memory devices, graphics software etc. Image Processing is used in various applications such as:

- Remote Sensing
- Medical Imaging
- Non-destructive Evaluation
- Forensic Studies
- Textiles
- Material Science.
- Military
- Film industry
- Document processing
- Graphic arts
- Printing Industry

II DESCRIPTION OF PACKAGES AND COMPUTING ENVIRONMENT

A. Mat Lab:

"MATLAB is a high-level language and interactive environment that enables one to perform computationally intensive tasks faster than with traditional programming languages such as C, C++, and Fortran." The web page of the MathWorks , Inc. at www.mathworks.com states that Mat lab was originally created by Cleve Moler, a Numerical Analyst in the Computer Science Department at the University of New Mexico. The first intended usage of Mat lab, also known as Matrix Laboratory, was to make LINPACK and EISPACK available to students without facing the difficulty of learning to use Fortran. Features of matlab follows

- ➢ It include high-level language.
- ➢ 2-D/3-D graphics, mathematical functions for various fields.
- ➢ Interactive tools for iterative exploration, design, and problem solving.

- For integrating MAT LAB-based algorithms with external applications and languages.
- In addition, Mat lab performs the numerical linear algebra computations using for instance
  Basic Linear Algebra Subroutines (BLAS) and Linear Algebra Package (LAPACK).

B. SciLab:

"Scilab is an open source, cross-platform numerical computational package as well as a High-level, numerically oriented programming language." Scilab was written by INRIA, the French National Research Institution, in 1990. The web page for Scilab at www.scilab.org states the syntax is largely based on Matlab language. This software is also intended to allow Matlab users to smoothly utilize the package. In order to help in this process, there exists a built in code translator which assists the user in converting their existing Matlab codes into a Scilab code. According to Scilab's web page, the main features of Scilab include hundreds of mathematical functions; high-level programming language; 2-D/3-D visualization; numerical computation; data analysis; and interface with Fortran, C, C++, and Java. Just like Octave, Scilab is also a free software distributed under CeCILL licenses. Scilab is fully compatible with Linux, Mac OS, and Windows platforms. Like Octave, the source code is available for usage as well as for editing. To download Scilab go to www.scilab.org/products/scilab/download Scilab also uses, for instance, the numerical libraries, LAPACK and BLAS. Unlike Octave, the syntax and built-in Scilab functions may not entirely agree with Matlab.

C. Octave:

"GNU Octave is a high-level language, primarily intended for numerical computations," as the reference for more information about Octave is www.octave.org. This package was developed by John W. Eaton and named after Octave Levenspiel, a professor at Oregon 7State University, whose research interest is chemical reaction engineering. Due to complexity of other software and Octave's interactive interface, it was necessary to redevelop this software to enable the usage beyond solving chemical reactor design problems. The first release of this package, primarily created by John W. Eaton, along with the contribution of other resources such as the users[5] Octave, written in C++ using the Standard Template Library, uses an interpreter to execute the scripting language. It is a free software available for everyone to use and redistribute with certain restrictions. Similar to Matlab, GNU Octave also uses for instance the LAPACK and BLAS libraries. The syntax of Octave is very similar to Mat lab, which allows Mat lab users to easily begin adapting to the package. Octave is available to download on different operating systems like Windows, Mac OS, and Linux. To download Octave go to http://sourceforge.net/projects/octave. A unique feature included in this package is that we can create a function by simply entering our code on the command line instead of using the editor.[6]

D. FreeMat:

FreeMat is a numerical computational package designed to be compatible with other commercial packages such as Matlab and Octave. The supported operating systems for FreeMat include Windows, Linux, and Mac OS X. Samit Basu created this program with the hope of constructing a free numerical computational package that is Matlab friendly. The web page of FreeMat at www.freemat.org. states that some of features for FreeMat include eigen value and singular value decompositions, 2D/3D plotting, parallel processing with MPI, handle-based graphics, function pointers, etc. To download FreeMat, go to http: //sourceforge.net/projects/freemat.[5]

E. Comparison of Matlab with scilab

- Scilab is as good as Matlab in quality and reliability, before talking about the price, open nature, etc.[5]
- Being free itself may be a sufficient reason for some, but not for everyone.
- "Use of Scilab for space mission analysis"
- Scilab uses LAPACK, just as Matlab does.
- Scilab uses some of the state of the art packages like ODEPACK and DASSL, some of which may not be available in Matlab itself.
- the licensing restrictions of some excellent open source software packages will prevent them being included from a commercial software package like Matlab - no such restrictions exist for using them with Scilab.
- As a result, Scilab may actually turn out to be better than Matlab in some areas.

The computations for this study are performed using Matlab R2011a , GNU Octave 3.0.4, FreeMat v4.0, and Scilab-5.3.1 under the Linux operating system Redhat Enterprise Linux 5. The Cluster tara in the UMBC High Performance Computing Facility (www.umbc.edu/hpcf) is used to carry out the computations and has a total of 86 nodes, with 82 used for computation. Each node features two quad-core Intel Nehalem X5550 processors (2.66 GHz, 8,192 kB cache per core) with 24 GB of memory.

E Basic Result Comparison of MatLab, Scilab, Octave, FreeMat

In the paper [8], the comparison result of the four packages such as MatLab , Scilab, Octave, FreeMat. For the purpose of plotting the graph, first the eigen values found and that are converted into 2D plots and corresponding values for each package is made . It shows that freemat is verymuch compatible with matlab. But it lacks fewer function like kronfor Kronecker products, Pcg for the conjugate gradient method and the sample function f(x)=xsin(x^2) have been considered with 1025 equally spaced data points in Fig. 1.

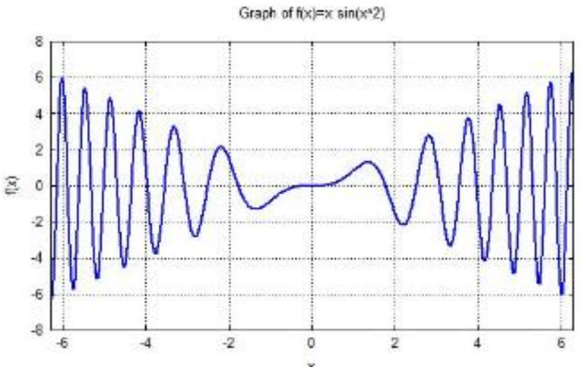

Fig. 1. f(x)=xsin(x^2) result in freemat

F. OPEN SOURCE SOFTWARE FOR MATLAB

Scilab offers many attractive features for an econometrician. Three of them are particularly important. First it is free, and therefore portable, a feature that is not guaranteed with any Commercial package, since none of these has become a standard for the profession. Econometricians working on different institutions are therefore not always able to exchange pieces of works, which they can do with Scilab. Similarly an econometrician using Scilab who changes their working location will have no problem with program reuse Second , Scilab is very similar to Gauss or Mat lab.

III. IMAGE PROCESSING TECHNIQUES:

A Noise filtering:

Noise filtering is used to filter the unnecessary *Noise* information from an image. It is also used to remove various types of noises from the images. Mostly this feature is interactive. Various filters like low pass, high pass, mean, median etc., are available[1]

Low Pass Filters:

- **Reconstruction filtering**, where an image is restored based on some knowledge of the type of degradation it has undergone. Filters that do this are often called "optimal filters".
- **Enhancement filtering**, which attempts to improve the (subjectively measured) quality of an image for human or machine interpretability. Enhancement filters are generally heuristic and problem oriented
- **Neighborhood-averaging filters** These replace the value of each pixel, *a[i,j]* say, by a weighted-average of the pixels in some neighborhood around it, i.e. a weighted sum of *a[i+p,j+q]*, with $p = -k$ *to* k, $q = -k$ *to* k for some positive *k*; the weights are non-negative with the highest weight on the $p = q = 0$ term. If all the weights are equal then this is a *mean* filter. "linear"
- **Median filters** This replaces each pixel value by the median of its neighbors, i.e. the value such that 50% of the values in the neighborhood are above, and 50% are below. This can be difficult and costly to implement due to the need for sorting of the values. However, this method is generally very good at preserving edges.
- **Mode filters** Each pixel value is replaced by its most common neighbor. This is a particularly useful filter for *classification* procedures where each pixel corresponds to an object which must be placed into a class; in remote sensing, for example, each class could be some type of terrain, crop type, water, etc..[3]

B *Image Enhancement Techniques:*

Sometimes images obtained from satellites and conventional and digital cameras lack in contrast and brightness because of the limitations of imaging sub systems and illumination conditions while capturing image. Images may have different types of noise. In *image enhancement*, the goal is to accentuate certain image features for subsequent analysis or for image display [1,2]. [2]Examples include contrast and edge enhancement, pseudo-coloring, noise filtering, sharpening, and magnifying. Image enhancement is useful in feature extraction, image analysis and an image display. The enhancement process itself does not increase the inherent information content in the data. It simply emphasizes certain specified image characteristics. Enhancement algorithms are generally interactive and application-dependent.[4]

C Accelerating MATLAB IPT Functions

The MATLAB functions that uses media processing operations, from image enhancement, restoration to image transforms, filtering, arithmetic, morphology, etc. Many of these functions are performance-critical and have already been accelerated in MATLAB using C code through the MATLAB MEX interface [3]. These functions also serve as building blocks to many other algorithms and thus have a strong impact in the scientific computing community. We classified those functions into four categories based on their inherent data-parallelism and algorithm characteristics. We name the first category (Category A) of functions data independent. The functions in this category operate independently on each data element (or pixel) with no communication between elements. Such functions fit naturally on a GPU(modern Graphics Processing Unit) and require relatively little effort in order to optimize and achieve high kernel performance. Bandwidth, including CPU-GPU transmission and off-chip GPU memory access, is usually the limiting factor for those functions, due to small computational load per data element. We call the second category (Category B) of functions data sharing. The functions in this category also possess abundant parallelism. However, the computation of a single element requires neighborhood information. In order to achieve good speedups on such functions, we need to carefully balance thread-level parallelism and data reuse, which are used to either hide high off-chip GPU memory access latency or to reduce the number of memory accesses. We also need deep knowledge of the underlying GPU architecture, so that we can map GPU resources such as arithmetic logic units (ALU), registers, shared memory, on-chip caches and threads efficiently in order to achieve high performance. The third category (Category C) of functions, algorithm dependent, have inherent parallelism, however in order to extract this parallelism we need to fundamentally re-think the algorithm with respect to the sequential MATLAB implementation on CPU. The speedups that we can expect from this category of functions may be limited by data dependencies, communication or sequential portions of the algorithm. We call the final category (Category D) of functions data dependent. These applications have very difficult to extract parallelism, due to fine grain communication, synchronization and data dependencies. In the following sections we elaborate on our experience through case studies in each of the categories. GPU exhibit good speedups against MATLAB CPU implementations. The reason may be due to the compiler difference. For different GPU, we develop different kernel functions. The reason is due to the hardware impact we use the vector type float4 instead of the scalar/float or float2 type and we assign more workload in each thread and thread group/block for ATI/AMD code.

Most of the available equipment is not capable of providing such data convincingly. The image fusion techniques allow the integration of different information sources. The fused image can have complementary spatial and spectral resolution characteristics. However, the standard image fusion techniques can distort the spectral information of the multi spectral data while merging. The multi spectral data are transmitted with coarser resolution. This will usually be two or four times lower.

IV. FUSION OF IMAGES:

Multisensor **Image fusion** is the process of combining relevant information from two or more images into a single image. The resulting image will be more informative than any of the input images. In remote sensing applications, the increasing availability of space borne into a sensors gives a motivation for different image fusion algorithms than single image. The resulting image will be more informative than any of the input images.[5]

A. Why image fusion

Multi sensor data fusion has become a discipline which demands more general formal solutions to a number of application cases. Several situations in image processing require both high spatial and high spectral information in a single image. This is important in remote sensing. However, the instruments are not capable of providing such information either by design or because of observational constraints. One possible solution for this is data fusion.[5].

Fig 2 shows the sample images before fusion with distorted spectral and spatial resolution.

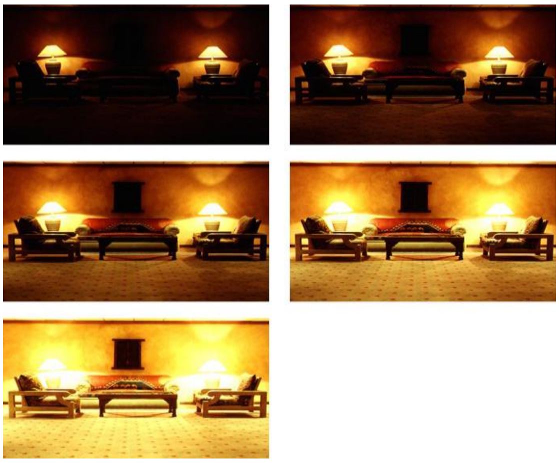

Fig. 2 The sample images before fusion

Here Fig 3 shows the sample image after fusion with full spectral and spatial resolution.

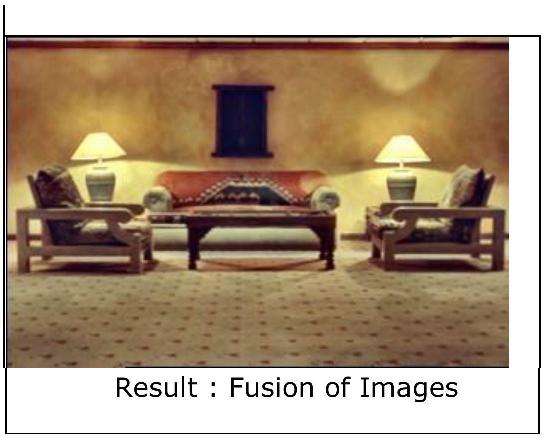

Fig. 3 The images after fusion

B. Standard image fusion

Image fusion methods can be broadly classified into two groups - spatial domain fusion and transform domain fusion. The fusion methods such as averaging, Brovey method, principal component analysis (PCA) and IHS based methods fall under spatial domain approaches. Another important spatial domain fusion method is the high pass filtering based technique. Here the high frequency details are injected into up sampled version images. The disadvantage of spatial domain approaches is that they produce spatial distortion in the fused image. Spectral distortion becomes negative factor while we go for further processing, such as classification problem. Spatial distortion can be very well handled by frequency domain approaches on image fusion [7]. The multi resolution analysis has become a very useful tool for analyzing remote sensing images. The discrete wavelet transform has become a very useful tool for fusion. Some other fusion methods are also there, such as Lapacian pyramid based, curvelet transform based etc. These methods show a better performance in spatial and spectral quality of the fused image compared to other spatial methods of fusion.

The images used in image fusion should already be registration. Misregistration is a major source of error in image fusion. Some well-known image fusion methods are:

- High pass filtering technique
- IHS transform based image fusion
- PCA based image fusion
- Wavelet transform image fusion
- pair-wise spatial frequency matching

C Complex image fusion

Standard Image Fusion generates a fused image in which each pixel is determined from a set of pixels in each source image. With the use of the Artificial Neural Networks it is possible to apply a new complex method based on the **Active Connection Fusion** (ACF).[3]The ACF method is an algorithm based entirely on the pixels and it is a particular application of the JNET network. The new system belongs to the family of **Active Connection Matrix** (ACM) systems:[4]

- each pixel of a source image is compared with the pixels in its neighborhood, with corresponding pixels of the source images and the other with the neighborhood of each of them.
- during this comparison are defined weights merged image.
- each new pixels of the fused image, after this comparison, is generated from the average of its own weights.

The assumption of the algorithm ACF is: the overall variance of all pixels of the source images implies the existence of a virtual image excellent, having the best amount of detail than any other source images.

V. METHODS:

A. Ordinary Differential Equations

One important feature to test would be the ODE solvers in the packages under consideration. For non-stiff ODEs, Matlab has three solvers: ode113, ode23, and ode45 implement an Adams-Bashforth-Moulton PECE solver and explicit Runge-Kutta formulas of orders 2 and 4, respectively. For stiff ODEs, Matlab has four ODE solvers: ode15s, ode23s, ode23t, and ode23tb implement the numerical differentiation formulas, a Rosenbrock formula, a trapezoidal rule using a “free” interpolant, and an implicit Runge-Kutta formula, respectively.

According to their documentations, Octave and Scilab solve non-stiff ODEs using the Adams methods and stiff equations using the backward differentiation formulas. These are implemented in lsode in Octave and ode in Scilab. The only ODE solver in FreeMat is ode45 which solves the initial value problem probably by a Runge-Kutta method. It becomes clear that all software packages considered have at least one solver. Matlab, Octave, and Scilab have state-of-the-art variable-order, variable-time step methods for both non-stiff and stiff ODEs available, with Matlab’s implementation being the richest and its stiff solvers being possibly more efficient. FreeMat is clearly significantly weaker than the other packages in that it does not provide a state-of-the-art ODE solver, particularly not for stiff problems

.

C. Parallel Computing

Parallel Computing is a well-established method today. It takes in fact two forms: shared memory and distributed-memory parallelism. On multi-core processors or on multi-core compute nodes with shared memory among all computational cores, software such as the numerical computational packages considered here automatically use all available cores, since the underlying libraries such as BLAS, LAPACK, etc. use them; studies for Matlab in [8] demonstrated the effectiveness of using two cores in some but not all cases, but the generally very limited effectiveness of using more than two cores. Since the investigations in [8], the situation has changed such that the number of cores used by Matlab, for instance, cannot even be controlled by the user any more, since that control was never respected by the underlying libraries anyway.1 Thus, even the ‘serial’ version for Matlab and the other numerical computational packages considered here are parallel on the shared memory of a compute node, this has at least potential for improved performance, and this feature is included with the basic license fee for Matlab. More recently, Matlab has started to provide built-in support for graphics processing units (GPUs). This is cutting-edge and should give a very significant advantage of Matlab over any other packages.

Still within the first form of parallel computing, Matlab offers the Parallel Computing Toolbox for a reasonable, fixed license fee (a few hundred dollars). This toolbox provides commands such as par for as an extension of the for loop that allow for the farming out of independent jobs in a master-worker paradigm. This is clearly useful for parameter studies, parameterized by the for loop. The parfor extension uses all available compute nodes assigned to the Matlab job, thus it does go beyond using one node, but each job is ‘serial’ and lives only on one node. The shared-memory parallelism discussed so far limits the size of any one job also that of a worker in a master-worker paradigm, to the memory available on one compute node. The second form of parallelism given by distributed-memory parallelism by contrast pools the memory of all nodes used by a job and thus enables the solution of much larger problems. It clearly has the potential to speed up the execution of jobs over ‘serial’ jobs, even if they use all computational cores on one node. Matlab offers the MATLAB Distributed Computing Server, which allows exactly this kind of parallelism. However, this product requires a substantial additional license fee that — moreover — is not fixed, but scales with the number of compute nodes (in total easily in the many thousands of dollars for a modest number of nodes).

The potential usefulness of distributed-memory parallelism, the historically late appearance of the Matlab product for it, and its very substantial cost has induced researchers to create alternatives. These include for instance pMatlab2 that can be run on top of either Matlab or other "clones" such as Octave. Particularly for pMatlab, a tutorial documentation and introduction to parallel computing is available [5]. In turn, since this distributed-memory parallel package is known to run in conjunction with Octave, this is one more reason to stir the reader to this numerical computational package to consider as alternative to Matlab.

D. Applicability of Package

Numerical computational packages see usage both in research and in teaching. In a research context, an individual researcher is often very concerned with the portability of research code and reproducibility of research results obtained by that code. This concern applies over long periods of time, as the researcher changes jobs and affiliations. The software Matlab, while widely available at academic institutions, might not be available at some others. Or even if it is available, it is often limited to a particular computer (as fixed-CPU licenses tend to be cheaper than floating license keys). Freely downloadable packages are an important alternative, since they can be downloaded to the researchers own desktop for convenience or to any other or to multiple machines for more effective use. The more complex test case in Section 3 is thus designed to give a feel for a research problem. Clearly, the use of alternatives assumes that the user's needs are limited to the basic functionalities of Matlab itself; Matlab does have a very rich set of toolboxes for a large variety of applications or for certain areas with more sophisticated algorithms. If the use of one of them is profitable or integral to the research, the other packages are likely not viable alternatives. In the teaching context, two types of courses should be distinguished: There are courses in which Matlab is simply used to let the student solve larger problems (e.g., to solve eigen value problems with larger matrices than 4 × 4) or to let the student focus on the application instead of mathematical algebra (e.g., solve linear system quickly and reliably in context of a physics or a biology problem). We assume here that the instructor or the textbook (or its supplement) give instructions on how to solve the problem using Matlab. The question for the present recommendation is then, whether instructions written for Matlab would be applicable nearly word-for-word in the alternative package. Thus, you would want function names to be the same (e.g., eig for the eigen value calculation) and syntax to behave the same. But it is not relevant if the underlying numerical methods used by the package are the same as those of Matlab. And due to the small size of problems in a typical teaching context, efficiency differences between Matlab and its alternatives are not particularly crucial[11].

Another type of course, in which Matlab is used very frequently, is courses on numerical methods. In those courses, the point is to explain — at least for a simpler version — the algorithms actually implemented in the numerical package. It becomes thus somewhat important

what the algorithm behind a function actually is, or at least its behavior needs to be the same as the algorithm discussed in class. Very likely in this context, the instructor needs to evaluate himself/herself on a case-by-case basis to see if the desired learning goal http://www.ll.mit.edu/mission/isr/pmatlab/pmatlab.html

of each programming problem is met. We do feel that our conclusions here apply to most homework encountered in typical introductory numerical analysis courses, and the alternatives should work fine to that end. But as the above discussion on ODE solver shows, there are limitations or restrictions that need to be incorporated in the assignments. For instance to bring out the difference in behavior between stiff and non-stiff ODE solvers, the ones available in Octave are sufficient, even if their function names do not agree with those in Matlab and their underlying methods are not exactly the same; but FreeMat's single ODE solver is not sufficient to conduct the desired computational

comparison between methods from two classes of solvers

.

VI. CONCLUSION

Image processing is the study of Representation and Manipulation of pictorial information. Digital Image Processing is performed on digital computers that manipulate images as arrays or matrices of numbers. This paper focuses on study of mat lab tools and open source tools for image fusion application. Open source software is user friendly since easy of installation. We can use all the features available in image processing techniques.

REFERENCES

[1] B. Boashash, "Estimating and Interpreting the Instantaneous Frequency of a Signal-Part I: Fundamentals", Proceedings of the IEEE, Vol. 80, No. 4, pp. 519-538, April 1992

[2].R.C.Gonazales, R.E.Woods ,Digital Image processing, reading M.A.Addition-Wesley,1995.

[3]. T. Saito and J. Toriwaki, New algorithms for Euclidean distance transformations of an n-dimensional digitised picture with applications, Patt. Recog. 27, 11, 1551–1565., 1994.

[4]Guenther, Raidl (May 1998). "An improved genetic algorithm for the multiconstrained 0-1 knapsackproblem". Evolutionary Computation Proceedings: 207

[5] Jeremy Kepner. Parallel MATLAB for Multicore and Multinode Computers. SIAM, 2009.

[6] Andrew M. Raim and Matthias K. Gobbert," Parallel performance studies for an elliptic test problem on the cluster tara. Technical Report HPCF–2010–2, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2010.

[7] Neeraj Sharma. "A comparative study of several numerical computational packages". M.S. thesis, Department of Mathematics and Statistics, University of Maryland, Baltimore County, 2010.

[8] Neeraj Sharma and Matthias K. Gobbert. "Performance studies for multithreading in Matlab with usage instructions on hpc. Technical Report HPCF–2009–1", UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2009.

[9] Neeraj Sharma and Matthias K. Gobbert. A comparative evaluation of Matlab, Octave,FreeMat, and Scilab for research and teaching. Technical Report HPCF–2010–7, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2010.

[10] M.J. Gooding et al. "Investigation into the fusion of multiple 4-D fetal echocardiography images to improve image quality", Ultrasound in Medincine and Biology, 2010;36(6):957-66

[11] J.B. Maintz, M.A. Viergever, 'A survey of medical image registration', Medical Image Analysis, 1998;2(1):1-36.