Digital Image Processing Using Convolution Algorithm

Mahesh Kumar[1], Deepika[2]

[1,2]Department of Computer science & Engineering,, MRKIET, Rewari, Haryana, India.)

ABSTRACT

Image processing suite is an image editor that offers all the standard editing and paint tools, as well as image layers and several other features that are usually not found in free image editors. The program also includes screen capture tools, an image browser, and as well as photo frames, channel adjustments and more. The product is part of the image processing software family such as Adobe Photoshop, Corel Draw and likewise. The project is an effort to build a Photoshop on a smaller case, with an easy to use user interface and an easy to understand user manual guideline to help the user how to play with image pixels and colors and other parameters The captured image can be loaded into the internal image editor for further editing, saved as image file (JPG, PNG, GIF, BMP) or automatically uploaded to your FTP server (upload the image and copy the URL to the clipboard). The Objective of this project is to create desktop-based software for general pc users and imaging professionals to play with parameters of an image by using the following features (the features are subject to change):

Keywords: Digital Image Processing, Photo editor, Image processing techniques, Histogram Equalization, Image Blurring.

INTRODUCTION

Image Processing and Analysis can be defined as the "act of examining images for the purpose of identifying objects and judging their significance" Image analyst study the remotely sensed data and attempt through logical process in detecting, identifying, classifying, measuring and evaluating the significance of physical and cultural objects, their patterns and spatial relationship.

In a most generalized way, a digital image is an array of numbers depicting spatial distribution of a certain field parameters (such as reflectivity of EM radiation, emissive, temperature or some geophysical or topographical elevation. Digital image consists of discrete picture elements called pixels. Associated with each pixel is a number represented as DN (Digital Number) that depicts the average radiance of relatively small area within a scene. The range of DN values being normally 0 to 255. The size of this area effects the reproduction of details within the scene. As the pixel size is reduced more scene detail is preserved in digital representation.

PURPOSE:

The purpose of this Report is to provide a detailed overview of our software application, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements.

Background

With the advent of mobile embedded multimedia devices that are required to perform a range of multimedia tasks, especially image processing tasks, the need to design efficient and high performance image processing systems in a Short time-to-market schedule needs to be addressed. Image Processing algorithms implemented in hardware have emerged as the most viable solution for improving the performance of image processing systems. The introduction of reconfigurable devices and system level hardware programming languages has further accelerated the design of image processing in hardware. Most of the system level hardware programming languages introduced and commonly used in the industry are highly hardware specific and requires intermediate to advance hardware knowledge to design and implement the system. In order to overcome this bottleneck various C-based hardware descriptive languages have been proposed over the past decade. These languages have greatly simplified the task of designing and verifying hardware implementation of the system.

Digital image processing is an ever expanding and dynamic area with applications reaching out into our everyday life such as medicine, space exploration, surveillance, authentication, automated industry inspection and many more areas. Applications such as these involve different processes like image enhancement and object detection. Implementing such applications on a general purpose computer can be easier, but not very time efficient due to additional constraints on memory and other peripheral devices. Application specific hardware implementation offers much greater speed than a software implementation. With advances in the VLSI (Very Large Scale Integrated) technology hardware implementation has become an attractive alternative. Implementing complex computation tasks on hardware and by exploiting parallelism and pipelining in algorithms yield significant reduction in execution times.

There are two types of technologies available for hardware design. Full custom hardware design also called as Application Specific Integrated Circuits (ASIC) and semi custom hardware device, which are programmable devices like Digital signal processors (DSPs) and Field Programmable Gate Arrays (FPGA‟s). Full custom ASIC design offers highest performance, but the complexity and the cost associated with the design is very high. FPGAs have traditionally been configured by hardware engineers using a Hardware Design Language (HDL).

Problem Statement

- Implementation of basic image-processing program to use with the aim of providing the user with a basic knowledge of the fundamental techniques of image filtering.

- To provide the user with an easy to use graphical user interface (GUI) with which the user can filter images using ready loaded filters or custom filters created by the user.

Objective of the Project:

- To explore and implement a basic image-processing program to use with the aim of providing the user with a basic knowledge of the fundamental techniques of image filtering.

- To provide the user with an easy to user graphical user interface (GUI) with which the user can filter images using ready loaded filters or custom filters created by the user.

- To gain experience in the Dot net framework.

Image processing is a very highly processor intensive activity. There are thousands of calculations to be completed when filtering an image with a simple 3x3-convolution matrix. There are many commonly available image-processing libraries which implement many of the functions within this project, such as DirectX, WinG and Intel‟s own Image Processing Library (IPL) which use hardware functions on the Intel CPUs.

This document is intended for professionals or general end pc users having keen interest in image processing and photography, which would like to change certain parameters and composition of their images.

2. ELEMENTARY CONCEPTS OF IMAGE PROCESSING

Overview

Analysis of the image data is done using various image processing techniques and methods that includes:

- Analog image processing
- Digital image processing.

Visual or Analog processing techniques is applied to hard copy data such as photographs or printouts. Image analysis in visual techniques adopts certain elements of interpretation, which are as: The use of these fundamental elements of depends not only on the area being studied, but the knowledge of the analyst has of the study area. For example the texture of an object is also very useful in distinguishing objects that may appear the same if the judging solely on tone (i.e., water and tree canopy, may have the same mean brightness values, but their texture is much different. Association is a very powerful image analysis tool when coupled with the general knowledge of the site. Thus we are adept at applying collateral data and personal knowledge to the task of image processing. With the combination of multi-concept of examining remotely sensed data in multi-spectral, multi-temporal, multi-scales and in conjunction with multidisciplinary, allows us to make a verdict not only as to what an object is but also its importance. Apart from this analog image processing techniques also includes optical photogram metric techniques allowing for precise measurement of the height, width, location, etc. of an object.

Digital Image Processing[9] is a collection of techniques for the manipulation of digital images by computers. The raw data received from the imaging sensors on the satellite platforms contains flaws and deficiencies. To overcome these flaws and deficiencies in order to get the originality of the data, it needs to undergo several steps of processing. This will vary from image to image depending on the type of image format, initial condition of the image and the information of interest and the composition of the image scene. Digital Image
Processing undergoes three general steps:

- Pre-processing[9]
- Display and enhancement[9]
- Information extraction[9]

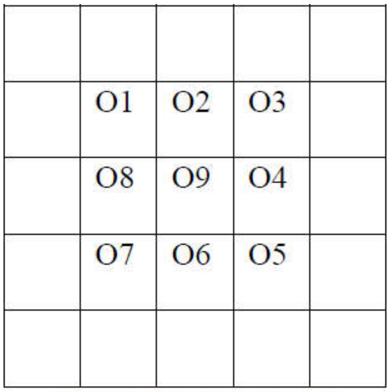
Pre-processing consists of those operations that prepare data for subsequent analysis that attempts to correct or compensate for systematic errors. The digital imageries are subjected to several corrections such as geometric, radiometric and atmospheric, though all this correction might not be necessarily is applied in all cases. These errors are systematic and can be removed before they reach the user. The investigator should decide which pre-processing techniques are relevant on the basis of the nature of the information to be extracted from remotely sensed data. After pre-processing is complete, the analyst may use feature extraction to reduce the dimensionality of the data. Thus feature extraction is the process of isolating the most useful components of the data for further study while discarding the less useful aspects (errors, noise etc). Feature extraction reduces the number of variables that must be examined, thereby saving time. Image Enhancement operations are carried out to improve the interpretability of the image by increasing apparent contrast among various features in the scene. The enhancement techniques depend upon two factors mainly as:

- The digital data (i.e. with spectral bands and resolution)
- The objectives of interpretation

As an image enhancement technique often drastically alters the original numeric data, it is normally used only for visual (manual) interpretation and not for further numeric analysis. Common enhancements include image reduction, image rectification, and image magnification, transect extraction, contrast adjustments, band rationing, spatial filtering, Fourier transformations, principal component analysis and texture transformation. Information Extraction is the last step toward the final output of the image analysis. After pre-processing and image enhancement the remotely sensed data is subjected to quantitative analysis to assign individual pixels to specific classes. Classification of the image is based on the known and unknown identity to classify the remainder of the image consisting of those pixels of unknown identity. After classification is complete, it is necessary to evaluate its accuracy by comparing the categories on the classified images with the areas of known identity on the ground. The final result of the analysis consists of maps (or images), data and a report. These three components of the result provide the user with full information concerning the source data, the method of analysis and the outcome and its reliability.

Special Features
Filtering

There are two main types of image filtering, linear and non-linear. The method implemented in my project is the linear method, which is achieved by the use of a convolution kernel. Image filtering is generally used to clean up or enhance images, by removing the effects of noise etc. There are many different filters available, from edge detection, enhancement to blur filters and emboss filters etc. Each of the filters are implemented as a convolution mask or convolution kernel.



	O1	O2	O3	
	O8	O9	O4	
	O7	O6	O5	

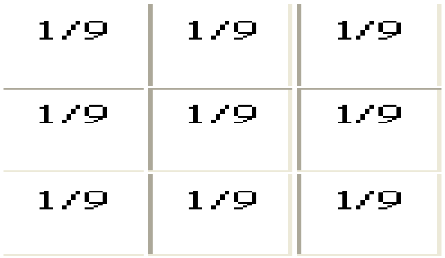
Figure 1 Convolution Kernel

The above mask is a simple 3x3 mask/kernel, where O9 is the pixel in question. This project will be using different filters from 3x3 kernels (above) to larger kernel such as 9x9 and 11x11.

Most off-the-shelf image packages like Paint Shop Pro does not support the larger masks due to the mass number of calculations required to check/correct each pixel. The larger the mask, the larger the number of calculations and ultimately the longer it takes to filter an image.

Standard Convolution Algorithm

The basic operation of a convolution kernel filtering algorithm is based around a kernel or NxN matrix where N is an odd number. The matrix represents the filter coefficients, which will be applied to the image. The matrix is shifted over the image a pixel at a time and the middle value of the matrix is calculated during each iteration. This involves getting the pixel value in the centre of the matrix as well as, in the case of a 3x3 kernel, the values of its eight neigh boring pixels. Each pixel is multiplied by its value or weight in the kernel, and then these values are added together. This result is divided by some divisor and finally a biasing factor is added. The final result becomes the new value of the centre pixel, and the matrix slides over to the next pixel and the process is repeated. 3x3 Convolution Kernel Image pixels



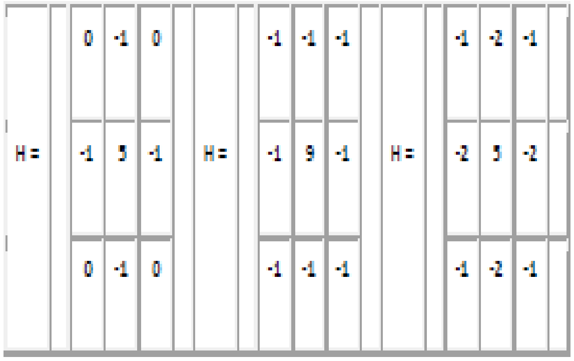
1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Figure 2 Convolution Kernel

The convolution kernel passes across the image from left to right. It then moves down to the new row of pixels and continues till the full image has been processed.

Common Filters
Edge Detection

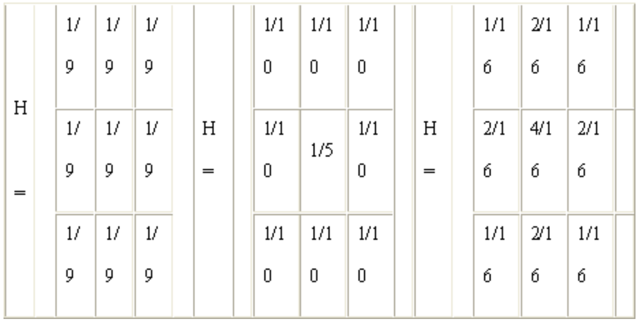
Psychophysical experiments indicate that a photograph or visual signal with crispier edges is often more subjectively pleasing that an exact photometric reproduction. Edges are not as sharp to the eye. By applying a discrete convolution filter such as one of the following to an image, edge become much clearer.



	0	-1	0		-1	-1	-1		-1	-2	-1
H=	-1	5	-1	H=	-1	9	-1	H=	-2	5	-2
	0	-1	0		-1	-1	-1		-1	-2	-1

Mean Filter

The average filter is also known as a mean, box or blur filter. The most common use for the average filter is to reduce noise in an image. The averaging filter does exactly as its name suggests, it moves over each pixel in the image and assigns the middle pixel in the kernel the average value of its eight adjacent neighbors. The kernel coefficients can be seen below. There are however a few drawbacks to using this filter. The first being that edges within the image may become blurred because of the changing value regions in that area also if there is a second value (one of the neighbors) which is highly unrepresentative of the area, the average won‟t be a true average.



H =	1/9	1/9	1/9		1/10	1/10	1/10			1/16	2/16	1/16	
	1/9	1/9	1/9	H =	1/10	1/5	1/10		H =	2/16	4/16	2/16	
	1/9	1/9	1/9		1/10	1/10	1/10			1/16	2/16	1/16	

Figure 3 Mean Filte

Image Filtering

Signals that are transferred over almost all forms of communication can be open to noise; an image may be subject to this noise and interference from several sources. These noise effects can be minimized by statistical filtering techniques or by application of spatial adhoc processing techniques.

Image noise arising from noisy sensors or channel transmission errors usually appears as discrete isolated pixel variations that are not spatially correlated. Pixels that are in error often appear markedly different from their neighbors. Many noise-cleaning algorithms make use of this fact.

By examining a pixel and checking to see if the brightness of this pixel is greater then the average brightness of its immediate neighbors by some threshold level, we can see if this pixel is valid or if it may be noise. If the pixel is noise then we replace this pixel with the average of the neighbors. Noisy images have a higher spatial frequency spectrum than normal images. Hence a simple low-pass filter can smoothen out the noise.

3. TOWARDS THE SOLUTION

System Design

A system design defines the components and the modules of the system and the relationships between them. We identified all the modules in our project and identified the relationships between them. We also determined the flow of data through the whole system and tried to minimize the confusions in it.

Project Modules

Here are the detailed description of each and every module of the proposed system.

Module 1: Image Zooming

This is a special type of zoom feature which will enable the user, zoom in an image according to his/her choice. In this module, the user will input a numeric value or more specifically, the factor, by which he wants to zoom-in the image. After the image is selected, loaded & displayed onto the screen, the zoom function will be applied on the complete image wherein the height & width of the image will be enlarged or multiplied according to the zoom factor. A notable thing in this function is, if an image is big and/or has a high resolution, large zoom factors would depend upon the system"s memory i.e. if we apply a higher zoom factor on an image which is large in size and / or higher in resolution, it will require a large amount of memory to get displayed on the screen. If there is not enough memory in the system, the application would throw a run-time error or an Out-of-memory exception.

Module 2: Image Auto-Correction

This feature will enable the user enhance the clarity of a zoomed image. Under this function, once an image is loaded and zoomed according to the user-inputted zoom factor, it will be further clarified or enhanced, which will improve its view ability. When an image is zoomed, every pixel in the image matrix enlarges which results in the pixels being viewed as large squares with boundaries depending upon the zoom factor. Hence, this function will remove the pixel boundaries and will make the entire image smooth thereby improving its clarity.

Module 3: Image Thresholding

Thresholding consists of segmenting an image into two regions-A particle region and a background region. In its most simple form, this process works by setting to white all pixels that belong to a gray level interval, called the threshold interval, and setting all other pixels in the image to black. The resulting image is referred to as binary image. For colour images, three thresholds must be specified, one for each component, ie Red, Green and Blue. In this module , we have implemented the „‟OTSU‟method for performing global thresholding which selects an optimum threshold value based on the spread of pixels, ie variance such that the interclass variance is maximum and intraclass variance is minimum.

Module 4: Image Rotation

Rotation performs a geometric transform which maps the position of a pixel in an input image onto a position in an output image by rotating it through a user-specified angle about an origin.

The rotation operator performs a transformation of the form:

x2=cos(α)*(x1-x0)+sin(α)*(y1-y0)+x0
y2=sin(α)*(x1-x0)+cos(α)*(y1-y0)+y0

where (x0,y0) are the coordinates of the center of rotation (in the input image) andαis the angle of rotation with clockwise rotations having positive angles.
Module 5 : Image Sharpening

Sharpening is one of the most impressive transformations you can apply to an image since it seems to bring out image detail that was not there before. What it actually does, however, is to emphasize edges in the image and make them easier for the eye to pick out -- while the visual effect is to make the image seem sharper, no new details are actually created.

To extract a sharpened view of an image, we smooth this image and then subtract the smoothed result from the original image.

$$g(x,y) = f(x,y) - f_{smooth}(x,y)$$

where ,fsmooth=smoothed version of f(x,y)

Now we are left with only high frequency edge descriptions. we now add some proportion of this gradient image back onto our original image with the high frequency components enhanced. The image has been sharpened according to this formula,

$$f_{sharp}(x,y) = f(x,y) + k * g(x,y)$$

where the scaling constant k is set to 0.7

Module 6: Image Smoothing

Through this module, we perform smoothing on the loaded image by applying low pass filter, i.e mean filter.

4. IMPLEMENTATION DETAILS OF PROPOSED SYSTEM

System Implementation

The implementation means how to implement the system, its pre effects and its post effects. It also includes all those activities that take place to convert the old system into new system.

The software for implementation includes -:

Distribution of software to the law related personnel by giving them a demonstration of the software and its utility for them.

- ❖ Install it on their PC.
- ❖ Handover the documentation
- ❖ User Training
- ❖ Training of Personnels To The New System

The plan of implementation should start with planning for training of concerned employees about the new system. This is generally the neglected aspect which ultimately results in the failure of the system. Good system may fail because of lack of training. Hardware resources can be calibrated; software can be tuned but human ware is such a resource that it needs special attention and treatment.

Following are the various facets of training required during the implementation. Training is required to be given, not only to the system operators but also to the users who will be really concerned with the performance of the system.

1) System operators will include computer operators, server administrators and any other system personnel who are directly or indirectly related to the system under consideration. Training to this group should include:-

- ❖ New equipments
- ❖ “Run” procedures
- ❖ Troubleshooting

2) User group will include all end users-primary as well as secondary users. In simple words, a primary user is one who actually operates the system and a secondary user is one who generally makes these operations meaningful.
The users should be given training about:

- ❖ Fundamentals
- ❖ Equipments
- ❖ System operations
- ❖ “Run” procedures

3) Training methodology may consist of:-

- ❖ Case studies
- ❖ Tutorials
- ❖ Training manuals
- ❖ Computer based training

No system will be effective unless the concerned personnel use it wisely and properly and that requires good training. Post Implementation review is very important after any system implementation. It can be reviewed by some of the technique like operational evolution, organizational impact and user management assessment. Although in my case, it is basically the user-review and feedback which provides me the performance and modification needs of the Software.

Details of Software

In this Project we are using:

Front End: C#

Tool Used: Microsoft Visual studio 2008

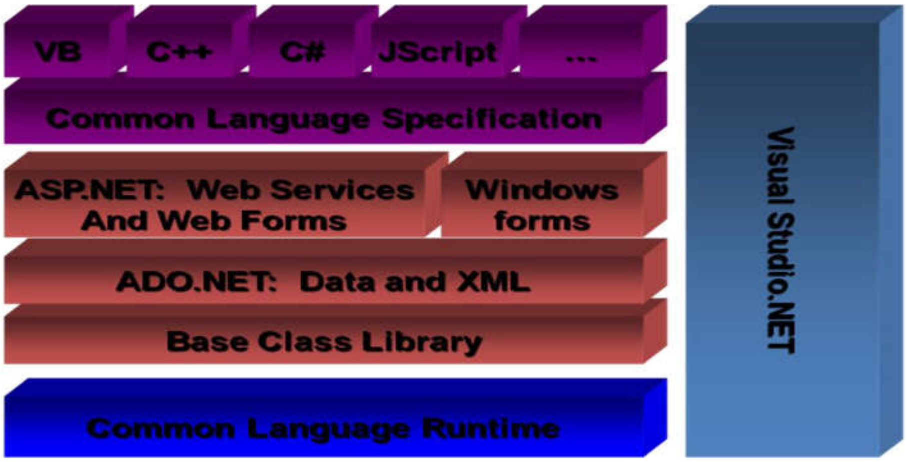
Operating System: WINDOWS 98/XP/Vista/7

C#

C# is a simple, modern, object oriented, and type-safe programming language derived from C and C++. C# (pronounced “C sharp”) is firmly planted in the C and C++ family tree of languages, and will immediately be familiar to C and C++ programmers. C# aims to combine the high productivity of Visual Basic and the raw power of C++.

C#(.Net):

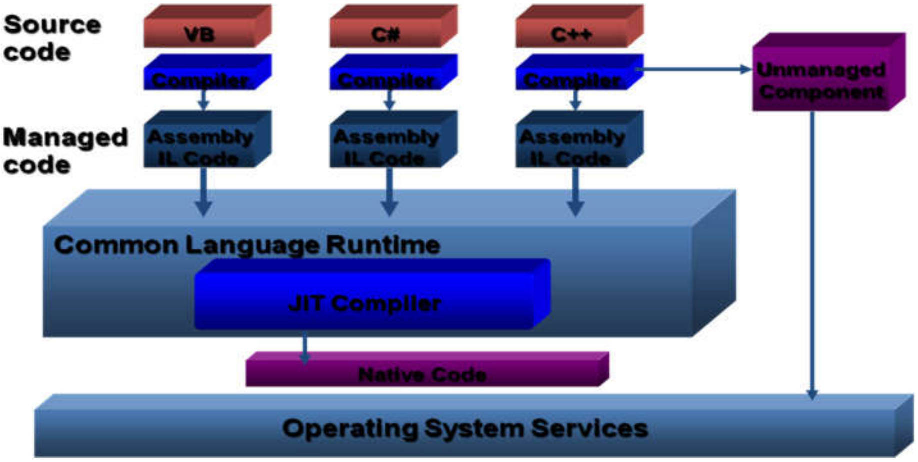
The .NET infrastructure comprises the .NET Framework, Microsoft Visual Studio.NET, the .NET Enterprise Servers, and Microsoft Windows.NET. The .NET infrastructure refers to all the technologies that make up the new environment for creating and running robust, scalable, distributed applications.



Runtime Execution Model of C# .Net:

C# supports 44 languages and the running procedure is common for all as shown in figure 6.3. The source codes are compiled through individual compilers and converted into Managed code (Assembly Intermediate language code). These assembly IL codes are compiled by a compiler named Just In Time Compiler (JIT complier) to convert the code into Native Code.

Managed code consist of IL code and Meta data.IL code is CPU independent machine language and Meta data is a structured information and describes programming constructs.



Net Framework:
The .Net framework is:-

1. A software development environment
2. A runtime engine for Managed Code.
3. A platform designed for Internet-Distributed software.

Microsoft Visual Studio .Net

A Integrated Development Environment
1. To develop managed and unmanaged applications
2. Supports C#, C++, Visual Basic, JavaScript etc.

Operating System: Windows 7

The operating system used for the development of the project is Windows 7 and like any other operating system Windows 7 has it‟scertain features due to which the 7 is used as a development platform for our system. The features of Windows 7 which are beneficial for us are:

he GUI (graphical user interface) of Windows 7 is very good as this is one of the latest O.S of Microsoft and specially designed for this purpose. It provides a more pluggable look to the user interface of the proposed system. Windows 7 is more reliable than any other version of Windows so the development of a system on Windows 7 is more secure as problems like hanging of Windows during file transfer can be easily incurred in the system.

Hardware Details

This section describes the minimum hardware requirements for Hardware:
CPU: Pentium based system.
RAM: A minimum of 2GB of RAM is recommended.
HDD: A minimum of 10 GB is recommended.

5. COST-BENEFIT ANALYSIS AND SYSTEM EVALUATION

Cost-Benefit Analysis

The most common way of carrying out an economic assessment of a proposed system is by comparing the expected costs of development and operation of the system with the benefits of having it in place. Assesment is based on the question whether the estimated costs are exceeded by the estimated income and other benefits.

The standard way of evaluating the economic benefits of any project is to carry out a cost benefit analysis, consisting of two steps

1) Identifying and estimating all of the costs and benefits of carrying out the project and operating the system.
2) Expressing these costs and benefits in common units.

CBAM- Cost Benefit Analysis Method-as exemplified by the cubes labeled P,A,S, and M, in the figure below (representing Performance, Availability, Security and Modifiability respectively).The CBAM guides system engineers and other stakeholders to determine the costs and benefits associated with architectural decisions that result in the system‟s qualities. Given this information, the stakeholders can then reflect upon and choose among the potential architectural decisions. For example, our the stakeholders could choose to invest their finite resources in some other quality attribute-perhaps that believing that higher performance will have a better Rate Of Interest.

A typical software project comprises of following expenditure heads:

(a) Manpower Cost:
It comprises of the effort put in the project regarding the power involved in the development of the project. It is calculated in person months by calculating the time spent by each member of the development team.

(b) Hardware Cost:
The hardware cost includes the cost incurred in the purchase of the hardware configuration to provide the system a suitable platform to perform its execution. The platform includes the processor, memory etc.

(c) Software Cost:
The software cost includes the cost incurred in the purchase of suitable software required for the development of the proposed system like operating system etc.

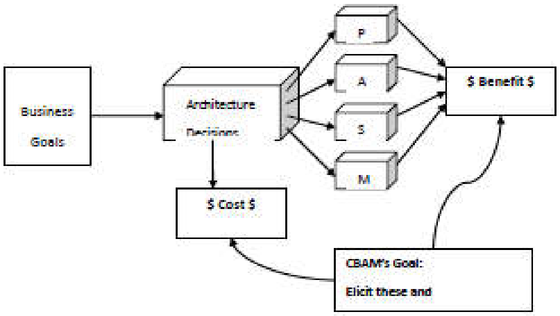


Figure 7 Cost Benefit Analysis

The major benefits of the Cost-Benefit Analysis are

(A) Direct Benefits:

These accrue directly from the operation of the proposed system. Example reduction in salary bills through new computerized system.

(B) Indirect Benefits:

These are generally secondary benefits such as increased accuracy through a friendlier user interface so that errors are minimized and cost reduced.

Technical costs of our software are negligible as this project is based on most popular technologies and any other special software or hardware is not required by our project. Hence the benefits from the project are far more than the costs incurred.

6. LIMITATIONS, FUTURE ENHANCEMENTS AND SUMMARY

Limitations

Since, every system has some limitations so our proposed system is also not untouchable in this regard. This system has following limitation:

□Proper check on some button/textbox is absent.

Future Enhancement

This software can be integrated as a full scale software to be used in various department as a complete image analysis and enhancement software. Various features can be added to it such as adding vertical and horizontal scroll-bar so that the zoomed image which goes out of the size of the preview area of the screen can be traversed, shrinking of image can also be added to it so that large image can be viewed in the provided screen size, features like adjusting brightness, sharpness, hue factor can also be added to it. On enhancement part, enhancement based on RBG factor can be included and facility of resize can be added so that image can be saved in a desired size as required by the user. Sector wise enhancement can be added so that the software can enhance specific sector of image as desired by the user.

SUMMARY

Since images are extensively used everywhere today, this application can be used in all those domains where image details become important like in the cricket replays that are used to give a particular decision, this application can be of extensive help through which we can zoom a particular image in a customized fashion and can also enhance it to have a better clarity. Moreover, it can also be used in crime investigations to look through the evidences in detail. The biggest advantage of this application is its speed & accuracy.

After we thoroughly understand the working concept behind GUI development & Image API, we will proceed towards the coding in a modular phase that we have finalized during the design stage of the application.

We will test every module after its development to validate & verify whether it is fulfilling all the necessary & expected requirements. We will continue this practice till all the modules are finally developed. After the successful completion & testing of all the modules individually, we will stitch them together to verify whether they are working properly in cohesion and is the inter-communication occurring accurately, as desired.

ACKNOWLEDGMENT

I take this opportunity to express my sincere thanks and deep gratitude to all those people who extended their wholehearted co-operation and have helped me in completing this project successfully. The satisfactions that accompany successful completion of any work would be incomplete without the mention of the people who made it possible.

REFERENCES

[1]. Azriel Rosenfeld, Picture Processing by Computer, New York: Academic Press, 1969

[2]. "Space Technology Hall of Fame:Inducted Technologies/1994" . Space Foundation. 1994. Archived from the original on 4 July 2011. Retrieved 7 January 2010.

[3]. Bhat, Pravin, et al. "Gradientshop: A gradient-domain optimization framework for image and video filtering." ACM Transactions on Graphics 29.2 (2010): 10.

[4]. Gonzalez, Rafael (2008). 'Digital Image Processing, 3rd'. Pearson Hall. ISBN 9780131687288.

[5]. Gonzalez, Rafael (2008). 'Digital Image Processing, 3rd'. Pearson Hall. ISBN 9780131687288.

[6]. A Brief, Early History of Computer Graphics in Film Archived 17 July 2012 at the Wayback Machine., Larry Yaeger, 16 August 2002 (last update), retrieved 24 March 2010

[7]. Solomon, C.J.; Breckon, T.P. (2010). Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab. Wiley-Blackwell. doi:10.1002/9780470689776. ISBN 0470844736.

[8]. Wilhelm Burger; Mark J. Burge (2007). Digital Image Processing: An Algorithmic Approach Using Java. Springer. ISBN 978-1-84628-379-6.

[9]. R. Fisher; K Dawson-Howe; A. Fitzgibbon; C. Robertson; E. Trucco (2005). Dictionary of Computer Vision and Image Processing. John Wiley. ISBN 978-0-470-01526-1.

[10]. Rafael C. Gonzalez; Richard E. Woods; Steven L. Eddins (2004). Digital Image Processing using MATLAB. Pearson Education. ISBN 978-81-7758-898-9.

[11]. Tim Morris (2004). Computer Vision and Image Processing. Palgrave Macmillan. ISBN 978-0-333-99451-1.

[12]. Milan Sonka; Vaclav Hlavac; Roger Boyle (1999). Image Processing, Analysis, and Machine Vision. PWS Publishing. ISBN 978-0-534-95393-5.