

P-1745

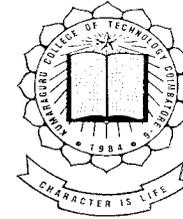


IMage PROcessing SoftWARE

By

M.Sangeetha

Reg. No 71203621043

of

Kumaraguru College of Technology

Coimbatore

(Affiliated to Anna University)

A PROJECT REPORT

Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements

for the award of the degree



Bayonet Information Technology Solutions

Run through the Future

112, Jubilee Complex, ChinnaSwamy Naidu Road, New Sidhapudhur, Coimbatore-44
Phone : 93642-33173, 94437-09376 E-mail : info.bayonet@gmail.com

Bayonet are pioneers in dealing with custom softwares. We provide innovative solutions to meet the technology needs of corporate industries. We specialize in business automation, e-commerce, embedded and research.

Bayonet Information Technology Solutions

PROJECT COMPLETION CERTIFICATE

This is to certificate that **M.SANGEETHA, Reg no: 71203621043**, final year M.C.A. Kumaraguru College of Technology, Coimbatore, has completed his project in this organization. He did the project entitled "**IMAGE PROCESSING SOFTWARE**" in Mainframe form January 2006 to June 2006 at our organization in partial fulfillment of the requirements for the award of M.C.A. Anna University. His work was timely from the company's perception and useful to the company.

As part of the company's policy we don't let out any sort of coding or pseudo code out of the company's premises in printed or in electronic media.

BONAFIDE CERTIFICATE

Certified that this project report titled

**IMage PROcessing softWARE
(IMPROWARE)**

Is the bonafide work of

Ms. M. Sangeetha (Reg. No: 71203621043)

who carried out the research under my supervision. Certified further that to the best of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. Anand Kishor P
21/6/06.
PROJECT GUIDE

Dr. Anand Kishor P
HEAD OF THE DEPARTMENT

We examined the candidate with university Registration Number. 71203621043

ABSTRACT

The main Objective of this project **IMPROWARE** - the Image Processing System is to modify and prepare the pixel values of a digitized image to produce a form that is more suitable for the subsequent operations with a generic model. This system replaces the existing system which has certain drawbacks compared to the vastly changing programming environment.

Major areas of this system focus on two main branches of image processing such as image enhancement and image restoration. It attempts to integrate the various image processing techniques.

Image enhancement attempts to improve the quality of the image or to emphasize particular aspects with the image. Such an objective usually implies a degree of subjective judgment about the resulting quality and will depend on the operation and the application in question.

The aim of image restoration is to recover the original image after it has been degraded by known effects such as geometric degradation within a mathematical model of the degradation is required so that restorative action can be taken.

ACKNOWLEDGEMENT

I wish to acknowledge my sincere gratitude and heartfelt thanks to our beloved principal, **Dr. Joseph V. Thanikal** and our former principal, **Dr. K. K. Padmanaban**, for having given me the adequate support and opportunity for completing this project work successfully.

I express my sincere thanks to **Dr. Gururajan**, professor and Head of the Department of Computer Applications, who with his careful supervision has ensured me in attaining perfection of work.

I extend my sincere thanks to **Ms. V. Geetha**, Project Coordinator, Department of Computer Applications for regarding us all the timely helps throughout the project.

I regard my heartfelt thanks and everlasting gratitude to my Project guide **Ms. P. Parameswari**, Senior Lecturer, Department of Computer Applications for her uplifting ideas and inspiring guidance, which have been very helpful in refining up the project.

I would like convey my sincere thanks to **Mr. Krishna Kumar**, Managing Director, Bayonet Solutions, Coimbatore, for his support, guidance and experience from which I have greatly benefited.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO
	ABSTRACT	iii
	ACKNOWLEDGEMENT	iv
1.	INTRODUCTION	
	1.1 OVERVIEW OF THE PROJECT	1
	1.2 ORGANIZATION PROFILE	2
2.	SYSTEM REQUIREMENT AND SPECIFICATION	
	2.1 HARDWARE REQUIREMENTS	3
	2.2 SOFTWARE REQUIREMENTS	3
	2.3 SOFTWARE OVERVIEW	4
3.	SYSTEM ANALYSIS	
	3.1 EXISTING SYSTEM	5
	3.2 PROPOSED SYSTEM	5

5.	DATA FLOW DIAGRAMS	16
6.	SYSTEM TESTING AND IMPLEMENTATION	
6.1	SYSTEM TESTING	17
6.1.1	WHITE BOX TESTING	18
6.1.2	BLACK BOX TESTING	18
6.1.3	INTEGRATION TESTING	18
6.1.4	TOP-DOWN INTEGRATION TESTING	19
6.1.5	BOTTOM-UP INTEGRATION TESTING	19
6.1.6	VALIDATION TESTING	19
6.1.7	SYSTEM TESTING	20
6.1.8	USER ACCEPTANCE TESTING	20
6.2	SYSTEM IMPLEMENTATION	
6.2.1	IMPLEMENTATION PROCESS	21
7.	CONCLUSION AND FUTURE ENHANCEMENT	
7.1	CONCLUSION	22
7.2	FUTURE ENHANCEMENTS	22

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF PROJECT

1.1.1 IMPROWARE – Image Processing Software

Improware involves processing or altering an existing image in a desired manner. It is the process of doing some graphical operations on images. Image processing is used for improving the pictorial information for human interpretation. It attempts to integrate and automate the various concepts of image processing.

The major modules in the project are:

- ♣ File
- ♣ Edge detection
- ♣ Filters
- ♣ Transformations
- ♣ Color control
- ♣ Photo utilities

1.1.2 Project Scope

- Performs file operations like opening and saving images.

1.2 ORGANIZATION PROFILE

Bayonet Solutions, Coimbatore is one of the fastest growing organizations. The organization offers wide range of solutions in application development, Web designing, Finance, Banking and so on. Though it is a recently started company, it has a wide range of clients from small sectors to good industries. They maintain a very good working relationship with the clients and the clients prefer Bayonet Solutions for their projects.

The main advantage of the Bayonet solutions is that they cater all types of the needs of clients. It is one of the client based organization which is sure to reach great heights in the future.

CHAPTER 2

SYSTEM REQUIREMENT AND SPECIFICATION

2.1 HARDWARE REQUIREMENTS

Processor	:	Intel Pentium III 800 MHz
Primary Memory (RAM)	:	256 MB
Secondary Memory (Hard disk)	:	40 GB
Monitor	:	COLOR, 15inch
Display card	:	SVGA
Mouse	:	Logitech
Keyboard	:	Standard 101 keys

2.2 SOFTWARE REQUIREMENTS

Operating System	:	Windows XP and above
Language	:	Java

2.3 SOFTWARE OVERVIEW

The java technology is actually a group of technologies.

- It is the language for developing the codes necessary for applications.
- It consists of the architecture for running the application that has been developed.
- It also consists of the tools necessary to build, compile, run the applications in combinations.

The java technology provides a portable, high performance and robust run time environment over heterogeneous networks. One of the most highly preferred aspect of java is the security features of the run time environment. Since java is a “*write once run anywhere*” language, they can be run in any system as long as that system implements the Java Virtual Machine. The language is designed to adapt to an evolving environment and are used even across networks. The java tools enabled programmers and end users flexibility to produce and use.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing system had the following drawbacks:

- Most of the current systems have highly advanced concepts such that the user cannot handle easily.
- The existing systems are not platform independent so cannot be used anywhere.
- Edge detection algorithms are not fully detected in any of the image processing software.
- The existing system contains only one workspace so the user cannot understand the difference between the original image and the processed image.

3.2 PROPOSED SYSTEM

The proposed system is developed using **JAVA**. The system has the following features:

- The proposed system provides enhanced functionality and efficient process design.
- It is designed in such a way that it is easily understood by the user.

- The system is divided into two workspaces so that the user can understand the difference between the original and the processed image.
- The functionalities of already existing related softwares are also added in order to ensure that it is easily understood and upgraded by the user.

CHAPTER 4

SYSTEM DESIGN

4.1 INPUT DESIGN

The input design is to get the user inputs. We convert the user inputs to the format which is recognized by the computer. But the Input screen must be designed in such a way that the user feels comfortable with the operations without the violating the input validation. The input design must also ensure that the data entry is logical and error free.

4.2 OUTPUT DESIGN

The output generated by the system must be clearly communicated to the user such that the users realize that the requirements specified by them has been clearly followed and executed. The results of the processing should be communicated to the users in a clean form. Effective output design will improve the clarity and performance of output. Sample Screens are shown in appendix.

4.3 ARCHITECTURE DESIGN

Architectural design is concerned with refining the conceptual design of the system identifying internal processing functions, decomposing high level functions into

4.3.1 Module Overview

IMPROWARE consists of eight modules:

- ♣ File
- ♣ Edge detection
- ♣ Filters
- ♣ Transformations
- ♣ Color control
- ♣ Photo utilities

File

The File module helps us to perform the file utilities. The sub modules are

- ⇒ New
- ⇒ Open
- ⇒ Save
- ⇒ Exit

Edge detection

Detecting edges is the basic operation of image processing. The edges of the item in the image hold much of the information in the image. The edges tell you

The Kirsh, Prewitt, Sobel and Quick masks are directional edge detectors. This means that each of the eight masks detects the edge in each direction. Given a pixel there are eight directions you can travel to neighboring pixel (above, below, left, right). Therefore there are eight possible directions for an edge the directional edge detectors can detect an edge in only one of the eight directions. If we want to detect only left to right edges, then we use only one of the eight masks. If we want to detect all the edges, we need to perform convolution over the image eight times using the each of the eight masks.

Kirsch Operator Masks

Kirsch – 0

```

5  5  5
-3  0 -3
-3 -3 -3

```

Kirsch – 1

```

-3  5  5
-3  0  5
-3 -3 -3

```

Kirsch – 2

```

-3 -3  5
-3  0  5
-3 -3  5

```

Kirsch – 3

```

-3 -3 -3
-3  0  5
-3  5  5

```

Kirsch – 4

```

-3 -3 -3
-3  0 -3
 5  5  5

```

Kirsch – 5

```

-3 -3 -3
 5  0 -3
 5  5 -3

```

Prewitt – Operator Masks

Prewitt – 0

1 1 1

1 -2 1

-1 -1 -1

Prewitt – 1

1 1 1

1 -2 1

1 -1 -1

Prewitt – 2

1 1 -1

-1 -2 -1

1 1 -1

Prewitt – 3

1 -1 -1

1 -2 -1

1 1 1

Prewitt – 4

-1 -1 -1

1 -2 1

1 1 1

Prewitt – 5

-1 -1 1

-1 -2 1

1 1 1

Prewitt - 6

-1 1 1

-1 -2 1

-1 1 1

Prewitt – 7

1 1 1

-1 -2 1

-1 -1 1

Sobel – Operator Masks

Sobel – 0

$$\begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$$

Sobel – 1

$$\begin{matrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{matrix}$$

Sobel – 2

$$\begin{matrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{matrix}$$

Sobel – 3

$$\begin{matrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{matrix}$$

Sobel – 4

$$\begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}$$

Sobel – 5

$$\begin{matrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{matrix}$$

Sobel – 6

$$\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

Sobel – 7

$$\begin{matrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{matrix}$$

Quick – Operator Masks

The Quick mask is so named because it can detect edges in all eight directions. It performs edge-detection using the single 3×3 Quick mask.

$$\begin{matrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{matrix}$$

Filters

Filtering is a common concept. When we adjust the bass and treble on series, we are filtering our certain audio frequencies and amplifying others. High pass filters pass high frequencies and stop low frequencies. Low pass filters stop high frequencies and pass low frequencies. In the same manner we can filter the spatial frequencies in image. A high pass filter can amplify or “Pass” frequent changes in gray level. A high pass filter will pass, amplify or enhance the edge. Low pass filters smoothens and often try to remove sharp edge.

⇒ *High pass filters* - High pass filter amplify or enhance an image. Image sharpening can be achieved in the frequency domain by a high pass filtering process.

There are three masks used for high pass filter.

High pass filter masks

High pass filter – 1

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

High pass filter – 2

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

High pass filter – 3

$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

The first mask amplifies all edges. In the second mask the image is enhanced more with tiny highlights. The third mask has little effect on low frequencies and a greater effect on areas with relatively high frequency.

⇒ *Low pass filters* – Low pass filters stop high frequencies and pass low frequencies. A low pass filtering smoothes out sharp transitions in gray levels and removes noise. Five convolution masks have been used. The image segment changes the gray level once, but with a sharp transition. Hence the blurring is achieved in the frequency domain by attenuating a specified of a high frequency component in the transformation.

Low pass filter – 1

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Low pass filter – 2

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Low pass filter – 3

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Transformations:

⇒ *Translation*: A translation is applied to an image by repositioning it along a straight line path from one co-ordinate location to another.

⇒ *Scaling*: Scaling is useful for saving disk space. Fitting a large image into a small display. A scaling transformation alters the size of the image.

⇒ *Shearing*: A transformation that distorts the shape of the image such that transformed shape appears as if an image is composed of internal layers that has been caused to slide over each other is called a shear.

⇒ *Rotation* : The image is rotated to a specified angle and the location of the pixels are computed using the formula,

$$x = x \cos () + y \sin ()$$

$$y = -x \sin () + y \cos ()$$

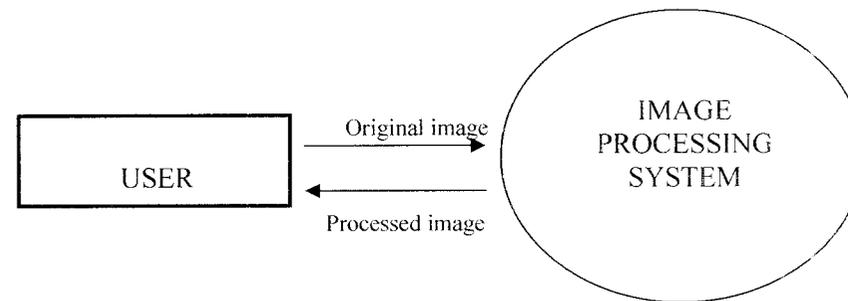
where () is the angle specified by the user.

Photo utilities:

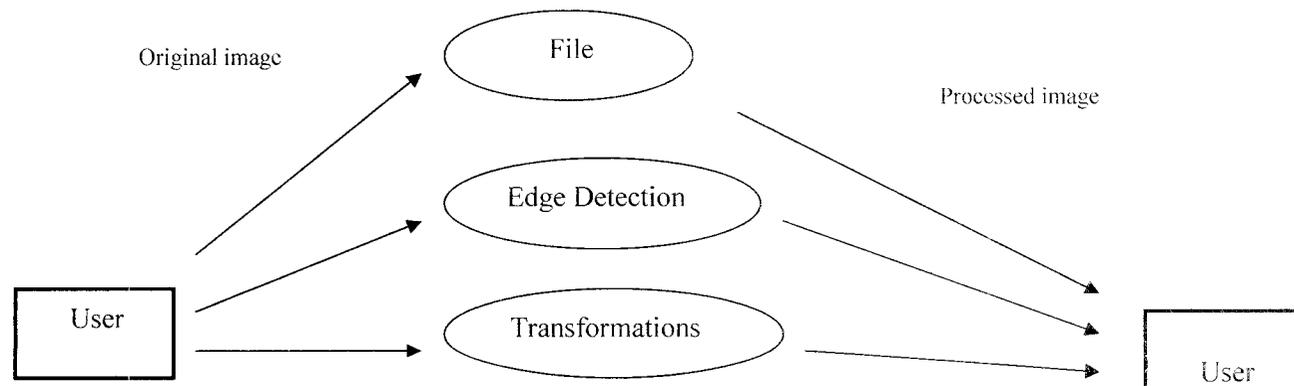
- ⇒ *Image load*: We load the bitmap image by specifying the path and displaying at the specified coordinates on the client.
- ⇒ *Embossing*: It is processing of plotting the difference between a pixel and a pixel about and to the left of it, this differences adding 128 to make the whole image appear gray.
- ⇒ *Engraving*: We can create engraving image by taking difference between a pixel and the pixel to its lower right and adding 128 to the result.
- ⇒ *Sweeping*: It gives the illusion of the image, which is in motion. In this we move the pixel from the lower right to upper left.
- ⇒ *Gray scale image*: It is the process where we convert color image to black and white image. We consider each of the pixel and get its RGB values.
- ⇒ *Negative*: It means that each pixel is logically complimented to give negative effect black pixel to white and white pixels to black.
- ⇒ *Mirror image*: The mirror image of the bit map (where the position of the pixels is reversed) is obtained both horizontally and vertical y.
- ⇒ *Blurring*: We can blur the image by averaging the pixels. here we just average each pixel with the next pixel to the right.

CHAPTER 5 DATA FLOW DIAGRAM

CONTEXT LEVEL DIAGRAM



Level 1



CHAPTER 6

SYSTEM TESTING AND IMPLEMENTATION

6.1 SYSTEM TESTING

The main objective of the system testing is to execute a program with the intention of finding errors. Usually testing is done to validate the user input and see if the user requirements are satisfied. We can also test the data flow by giving the actual data.

A good test case is one that has a probability of finding an as-yet-undiscovered error. Successful test is one that uncovers an as-yet-undiscovered error. A series of tests were performed.

The various types of testing are

- White box testing.
- Black box testing.
- Integration testing.
- Top-down integration.
- Bottom-up integration.
- Validation testing.
- System testing & User acceptance testing.

6.1.1 WHITE BOX TESTING

White box testing is essentially path tests. The simplest approach is to ensure that every statement is executed at least once. We can also exercise each condition for each decision statement at least once. White box testing gives the highest possible error yield of all the testing techniques. Using white box testing, the software engineer can derive test case that guarantee that all independent paths within a module have been exercised at least once and exercise all logical decisions.

6.1.2 BLACK BOX TESTING

Black box testing is designed to exercise the program to its external specifications. It focuses on the functional requirements of the software. Since the testers are not likely to be biased by the knowledge of program design, it will likely provide tests that resemble the user's environment. If sufficiently detailed requirements are not available to design the functional tests, it is also likely that the requirements were inadequate for the program design as well. When errors are found a closer examination is required. It also identifies the interface errors and errors in data structures.

6.1.3 INTEGRATION TESTING

The proper approach to integration depends on both the kind of system being used and the nature of the development project. The initial problem is to establish a framework on which to run these various elements. These initial integration tests are generally a combination of black and white box tests to ensure that the relevant system

6.1.4 TOP-DOWN INTEGRATION TESTING.

Top-down integration is essentially a prototyping method. The initial tests establish a basic skeleton from the top and each new module adds capability. The functions of lower level modules that are not initially present must be simulated by program stubs. Certain logical conditions such as error handling or special checking cannot be done until most of the system is being integrated.

6.1.5 BOTTOM-UP INTEGRATION TESTING.

In Bottom-up integration each module is individually tested and they are all assembled and run. If the system executes, the requirement of the stubs is eliminated. The need for very sophisticated drivers is needed for very high quality modules. Only less effort is needed for module integration.

6.1.6 VALIDATION TESTING.

Validation testing is a final series of software tests. Validation tests succeed when software functions in a manner that can be reasonably expected by the customer. Software validation is achieved through a series of black-box tests that demonstrate conformity with requirements. After each validation test case has been conducted, one of two possible conditions exists (1) The function or performance characteristics conform to specification and are expected or (2) a deviation from specification is undiscovered and a deficiency list is created.

6.1.7 SYSTEM TESTING.

The system testing can be done only when the objectives are clearly specified. Even if the objectives are clear, they may not accurately reflect the end users' needs. The purpose of the system test is to find those cases in which the system does not work as intended, regardless of the specifications. If the system does not meet the user's real need then it will be a disadvantage. The most critical problems should be stressed most thoroughly during tests that include performance, recovery and human factor issues.

Compatibility test intends to expose those areas where the system has improper incompatibilities.

The recovery test determines the behavior of the system after the occurrence of an error or an abnormal condition. Regression testing is a system test that forces the software to fail in a variety of ways and verifies that recovery is properly performed.

Security testing attempts to verify that protection mechanisms built into a system will, in fact, protect it from improper penetration.

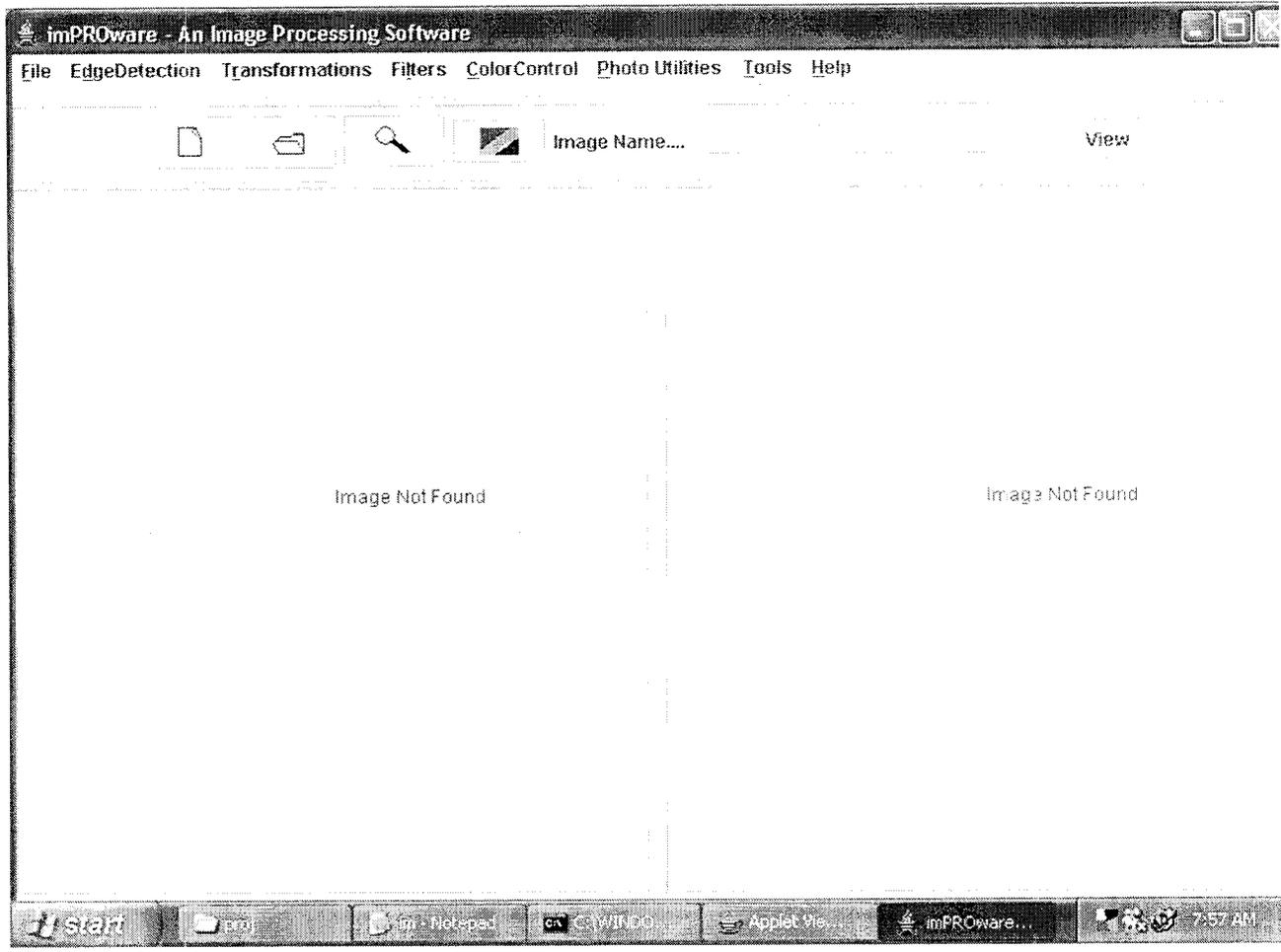
6.1.8 USER ACCEPTANCE TESTING.

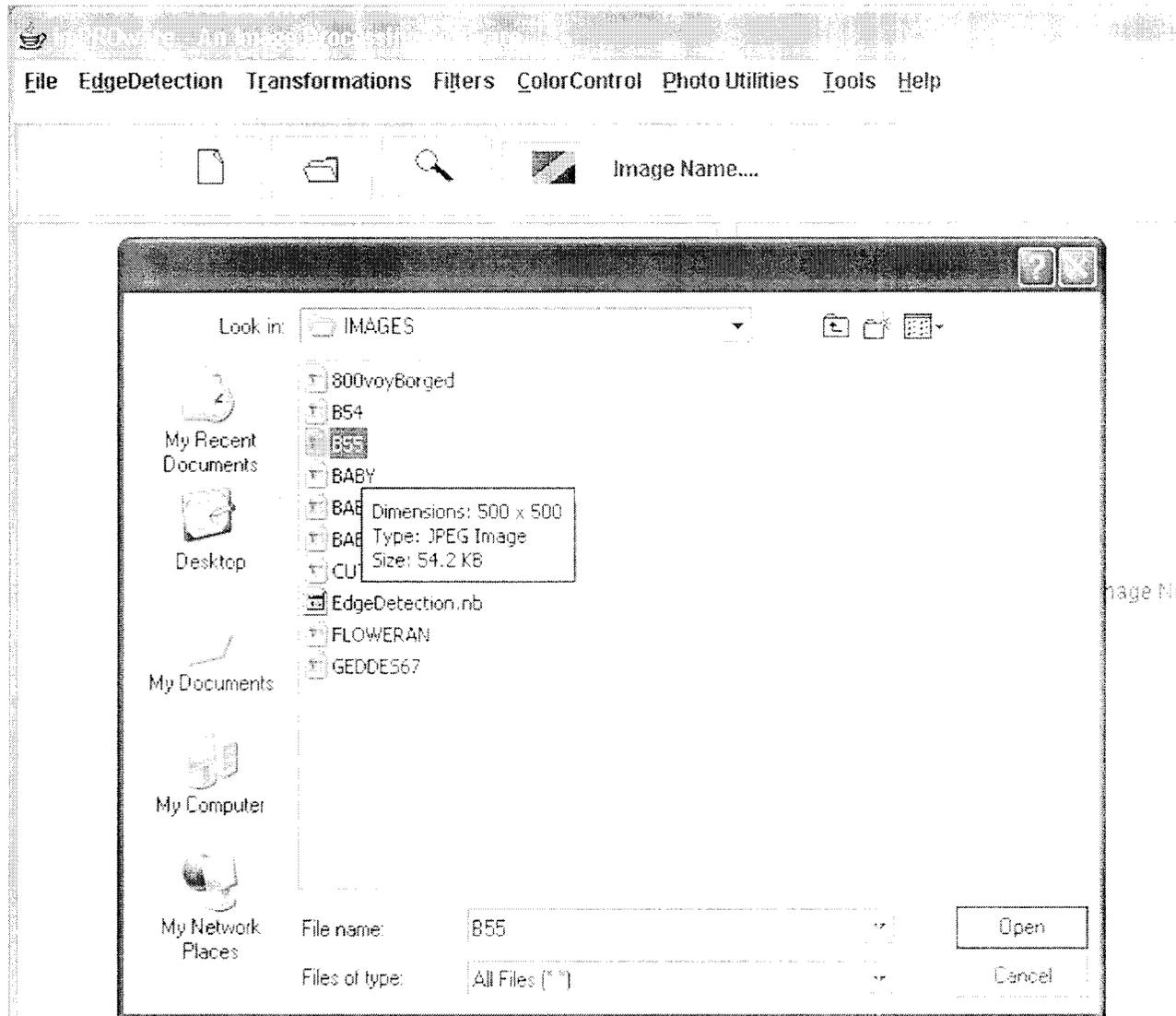
A user acceptance test has the object of telling the user on the validity and reliability of the system. It verifies that the system's procedures operate to system specifications and the integrity of vital data is maintained. Performance of an acceptance test is actually the user's show. User motivation and knowledge are critical for the

APPENDICES

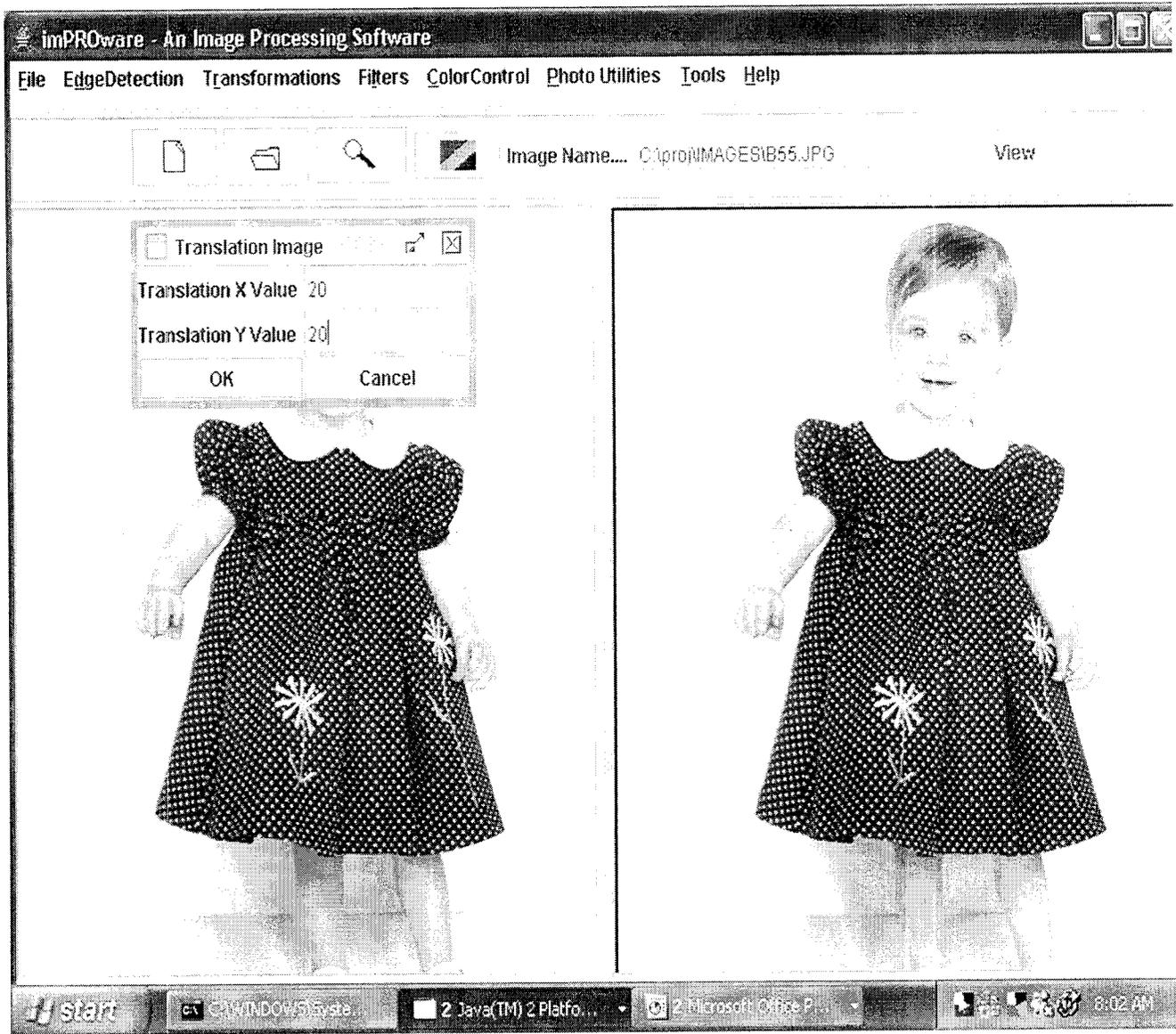
APPENDIX 1

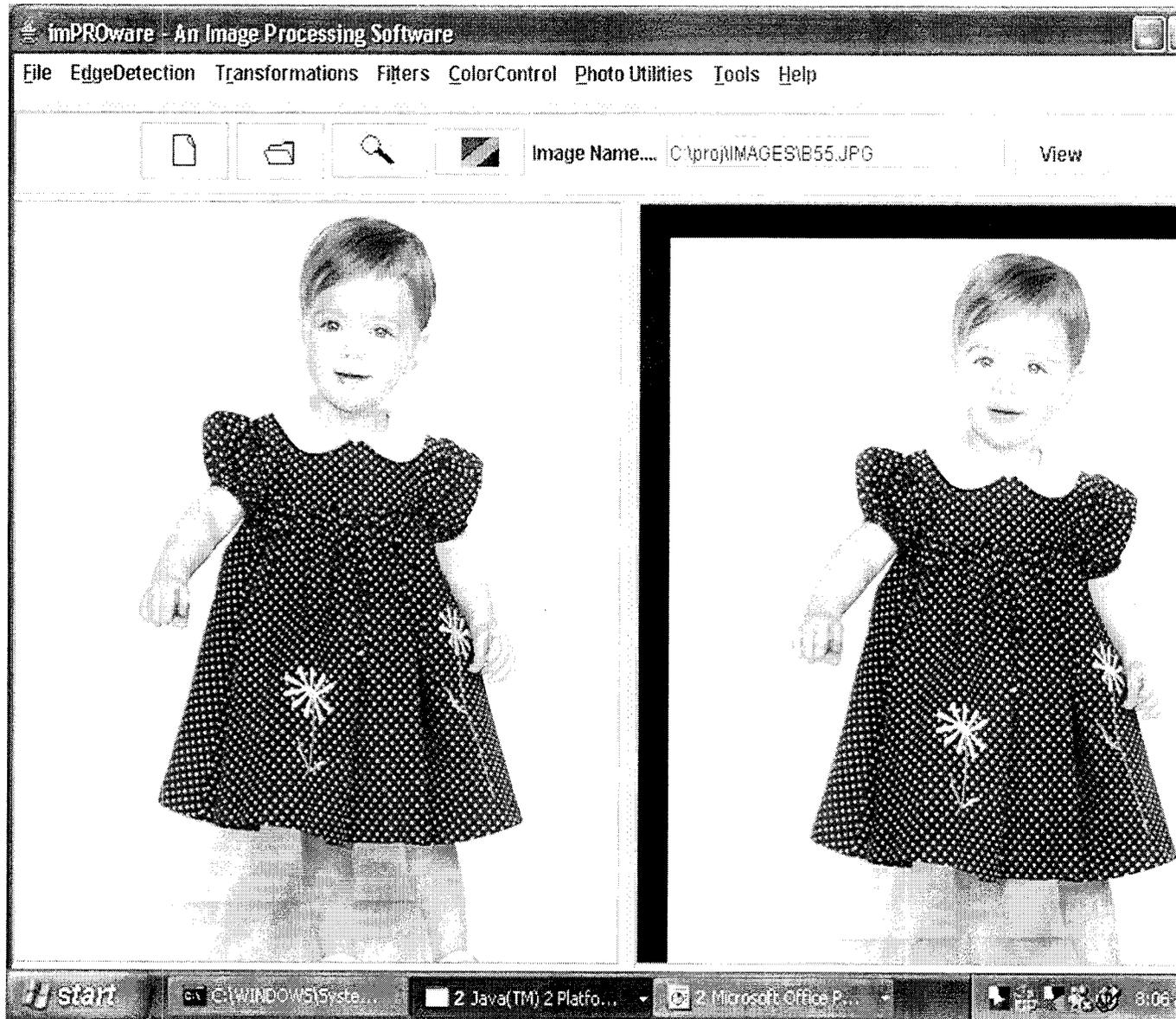


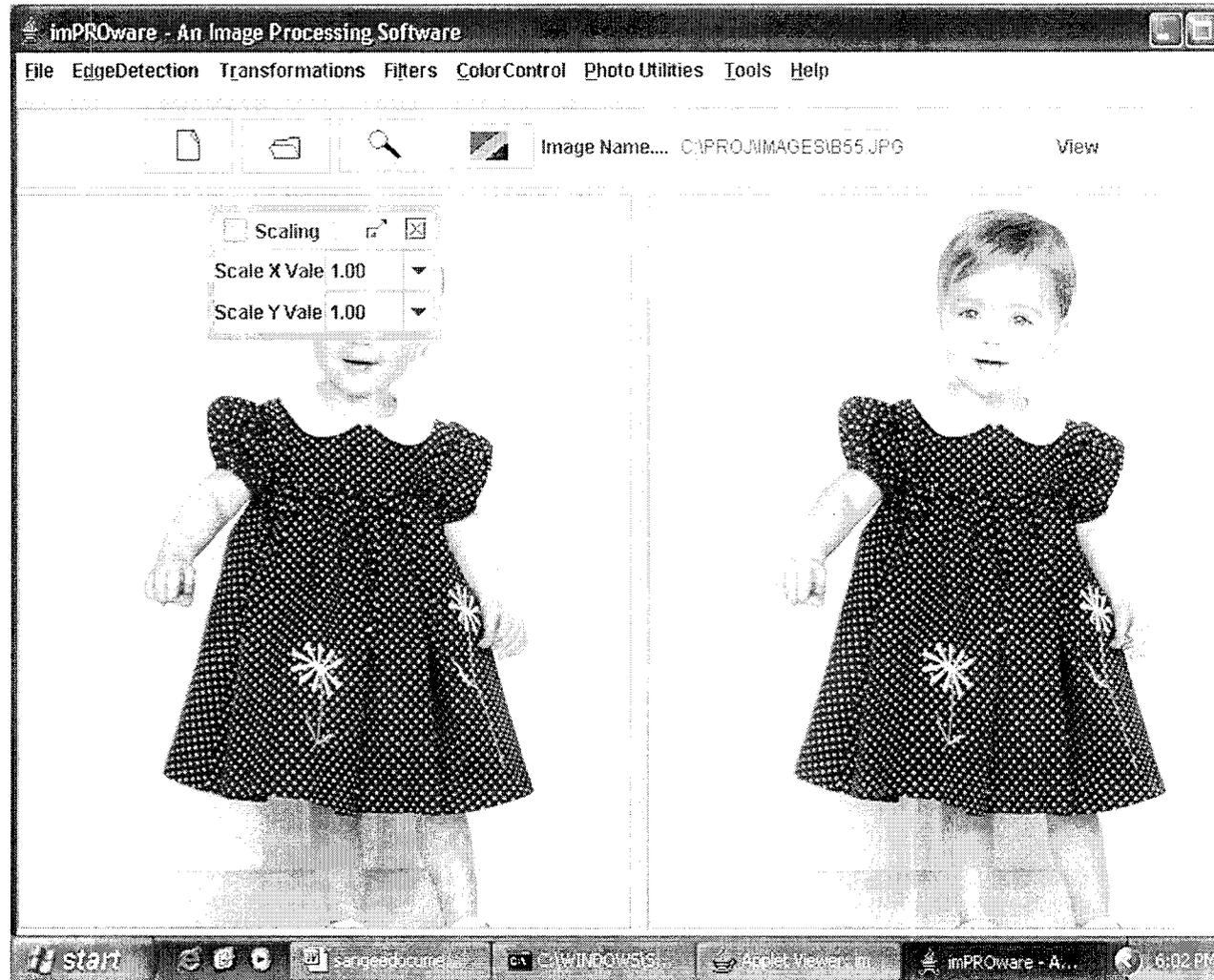








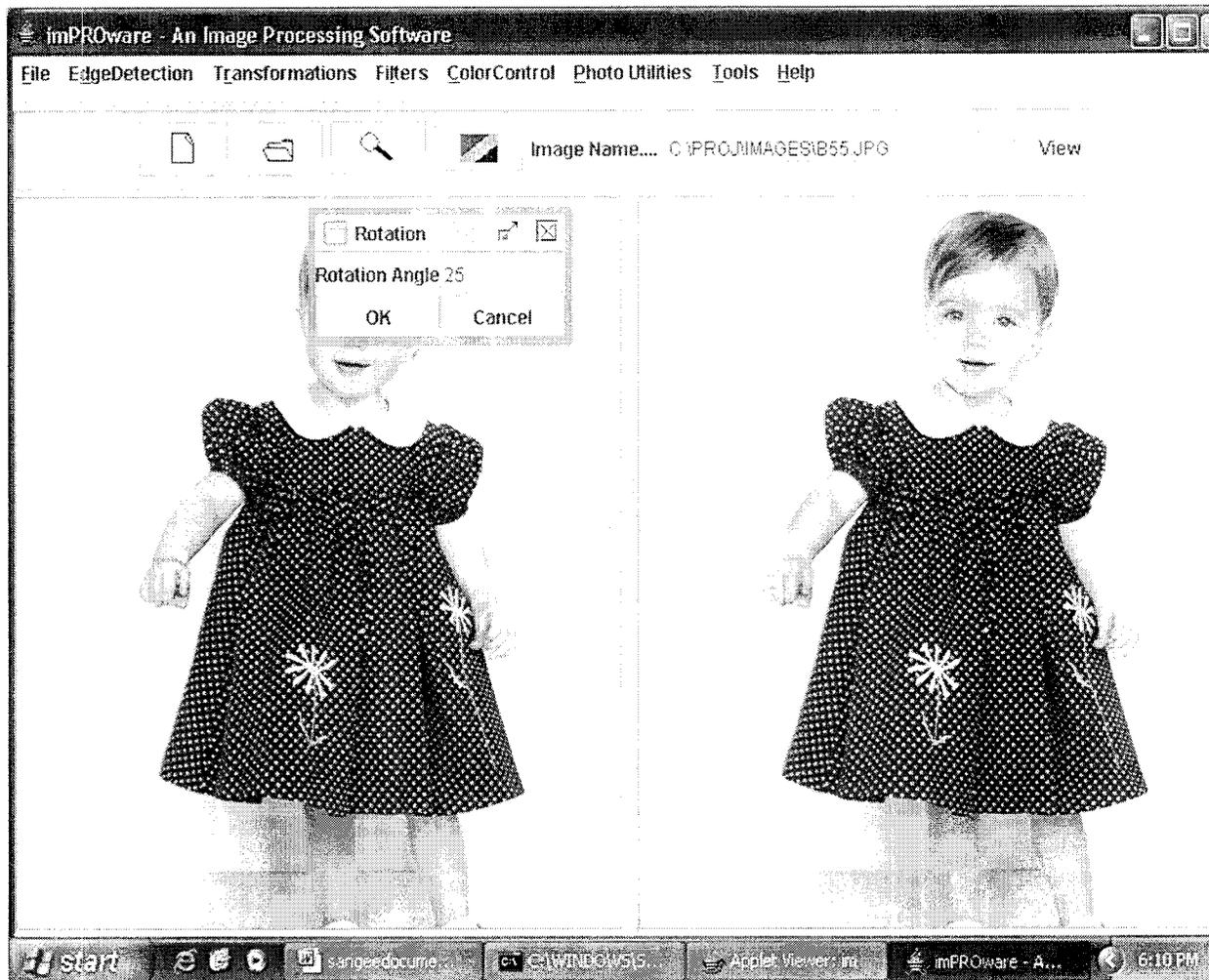


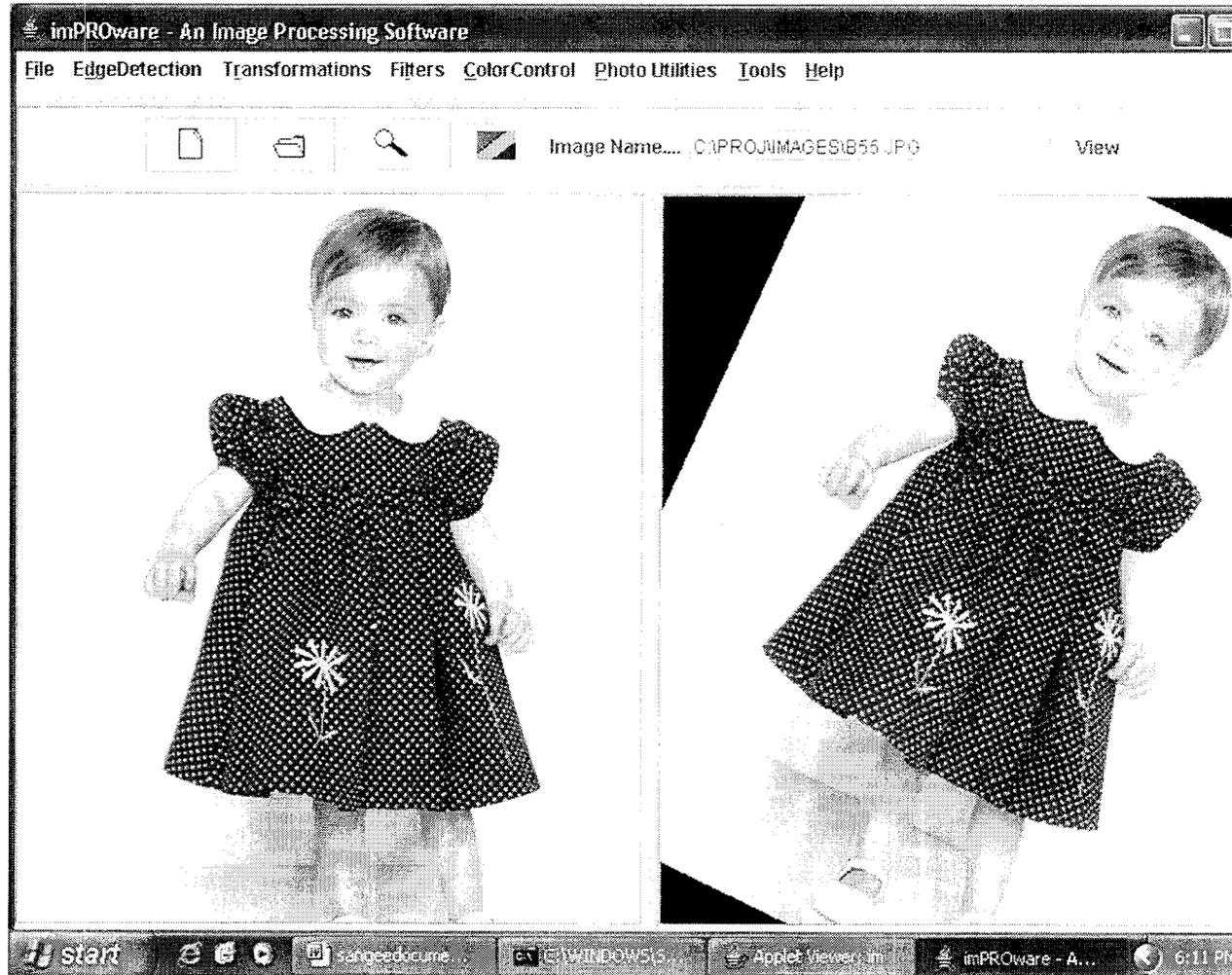




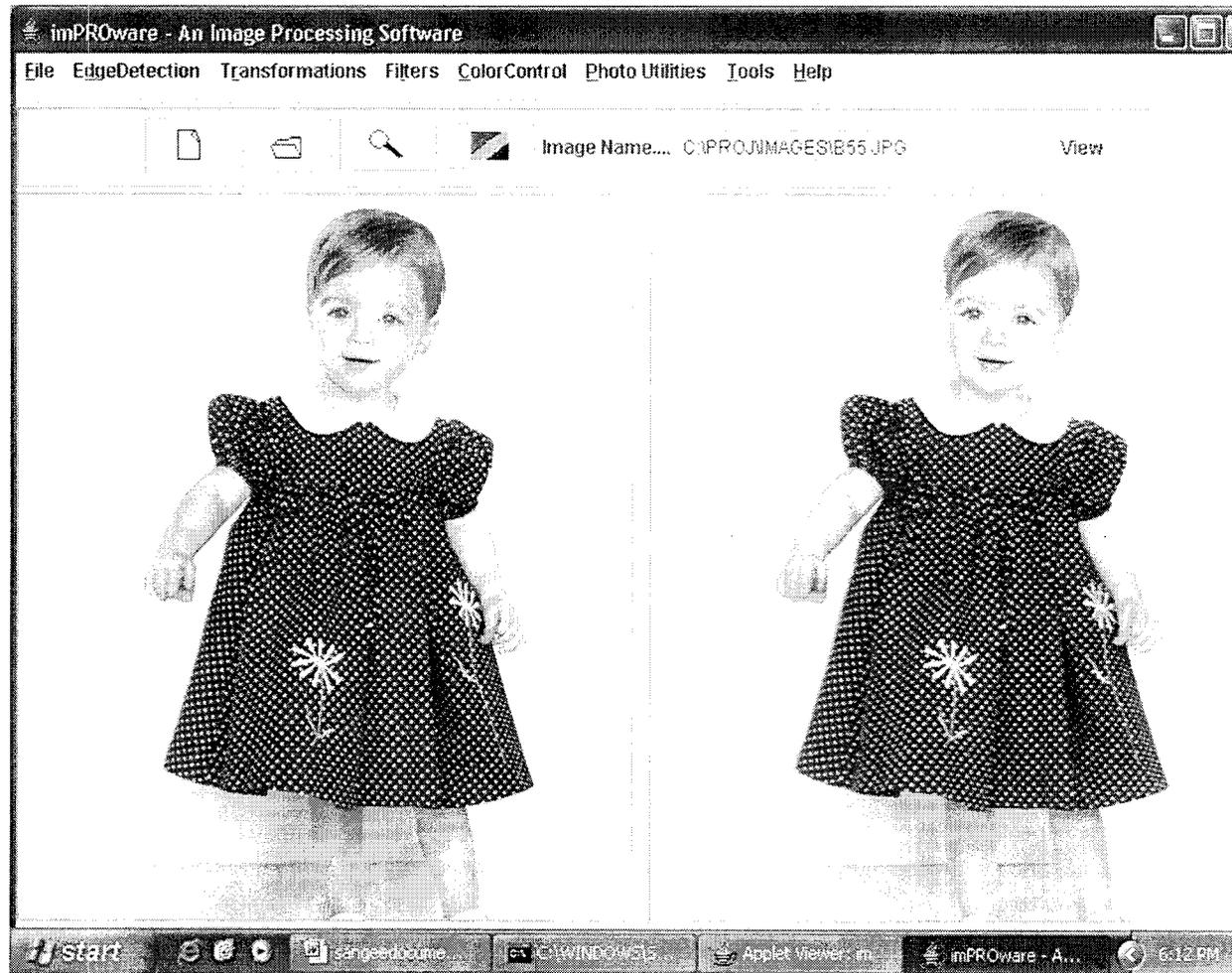








Rotation Output Page



Low pass filter – 1

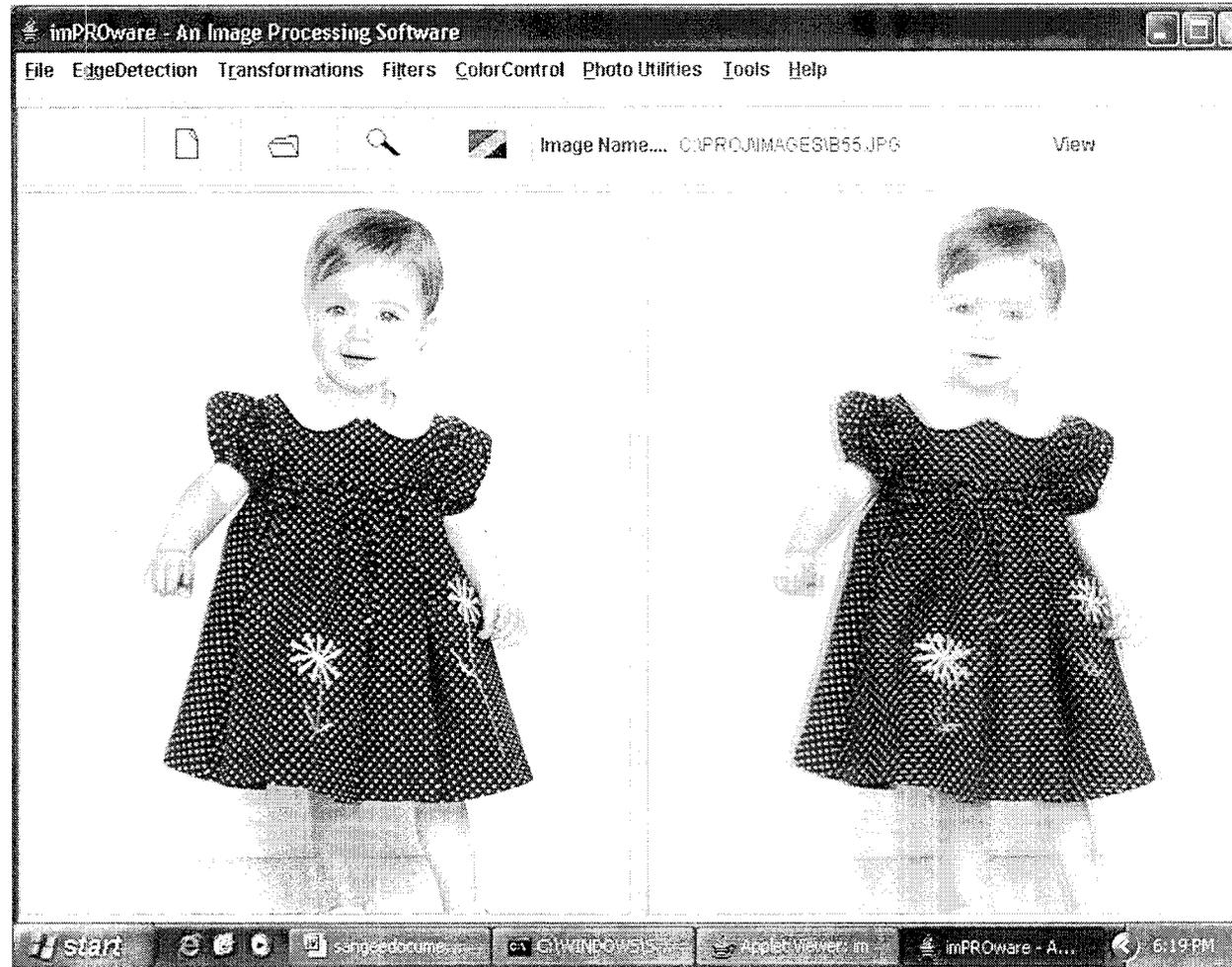




Low pass filter – 3



Low pass filter - 4



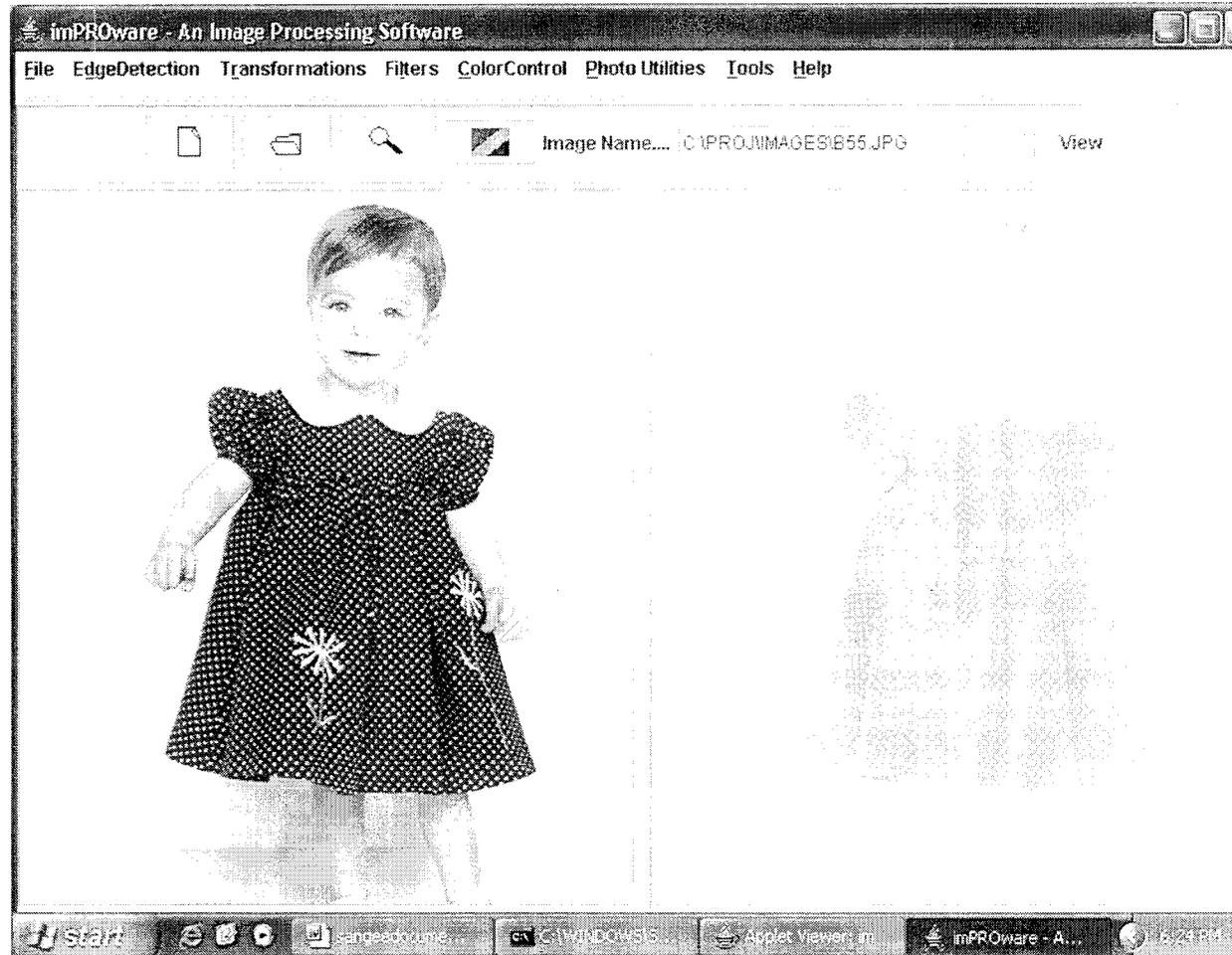




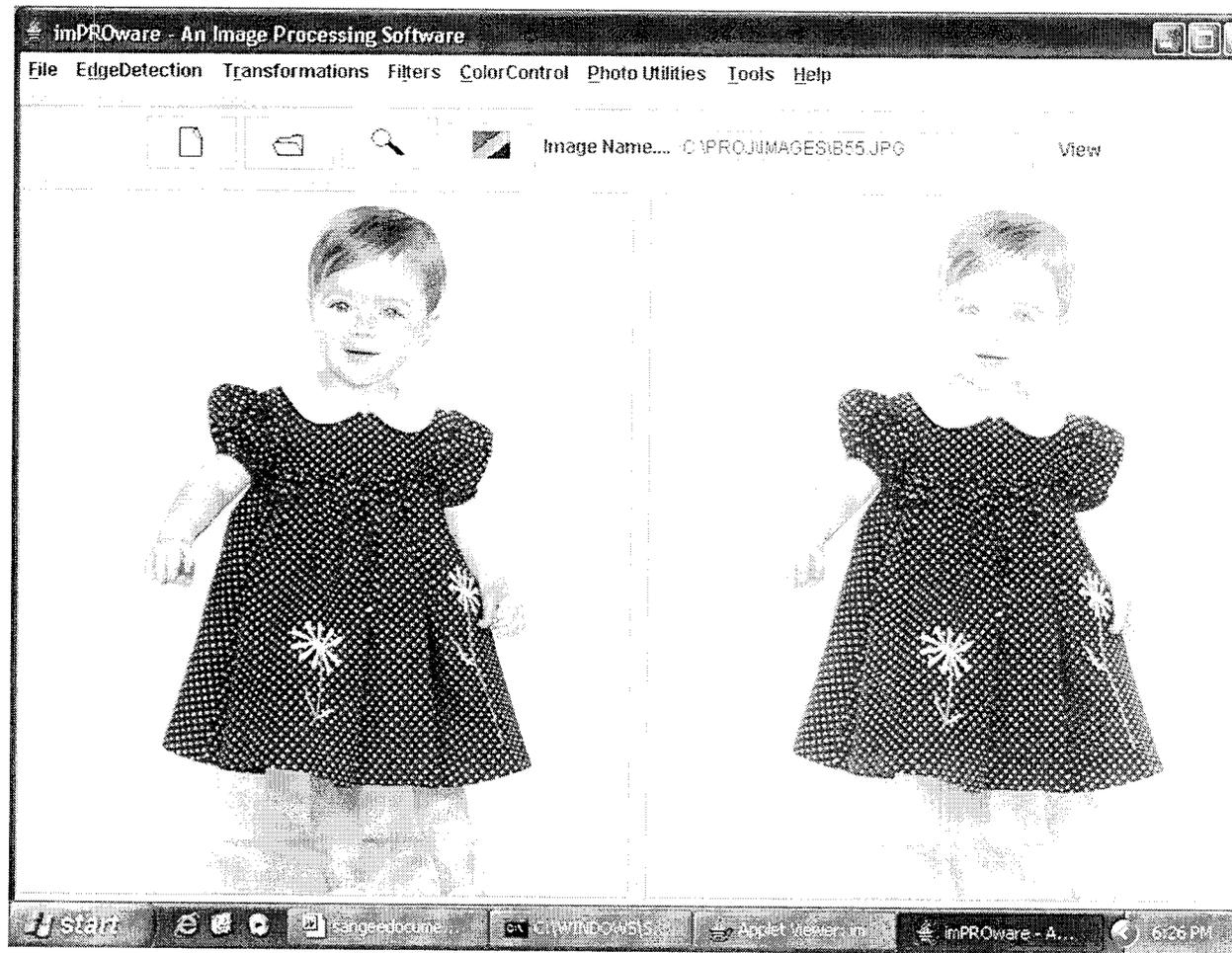
High Pass filter - 2



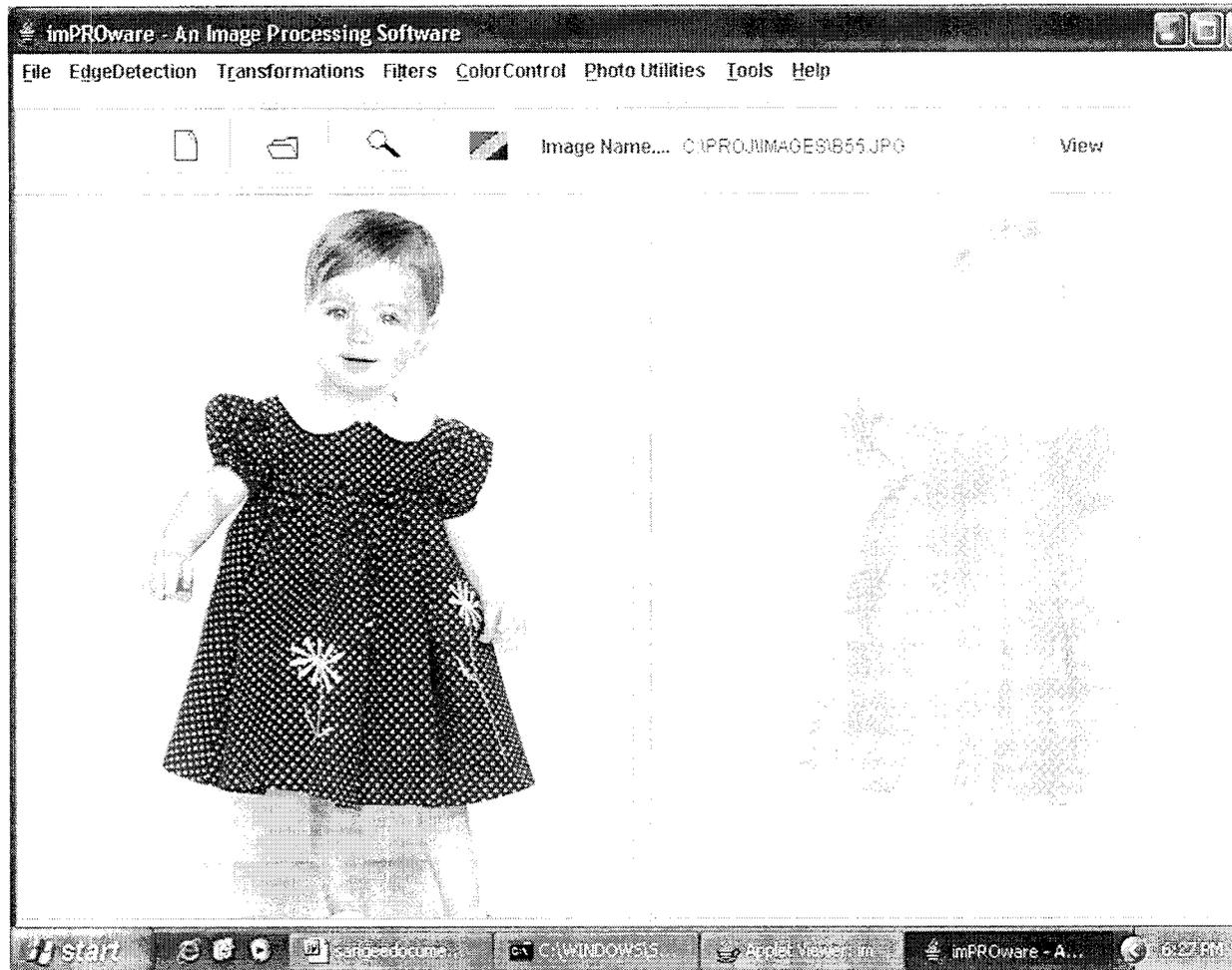
High pass filter - 3



Color Control – Changing RGB values



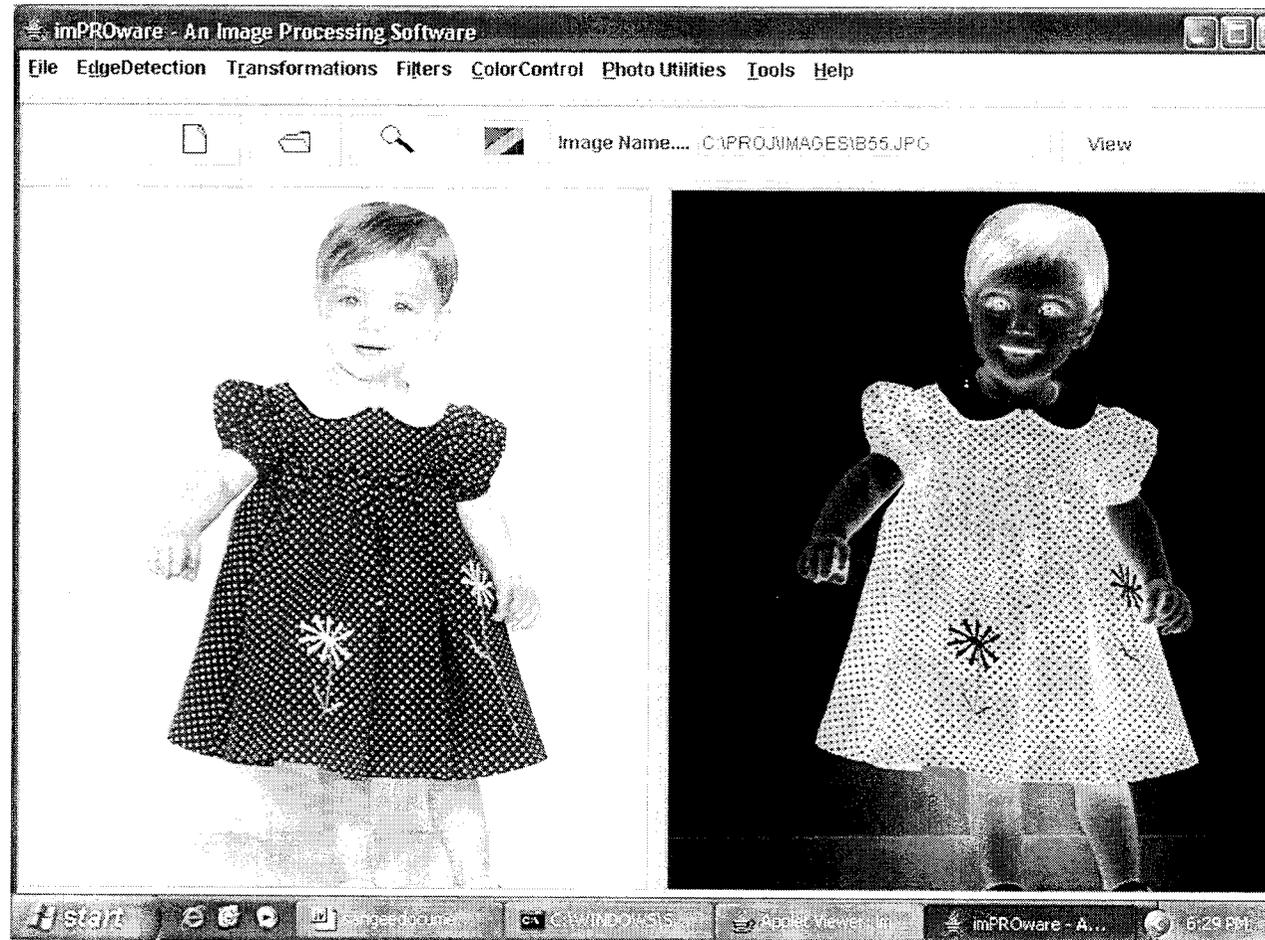
Color Control – Changing the red value



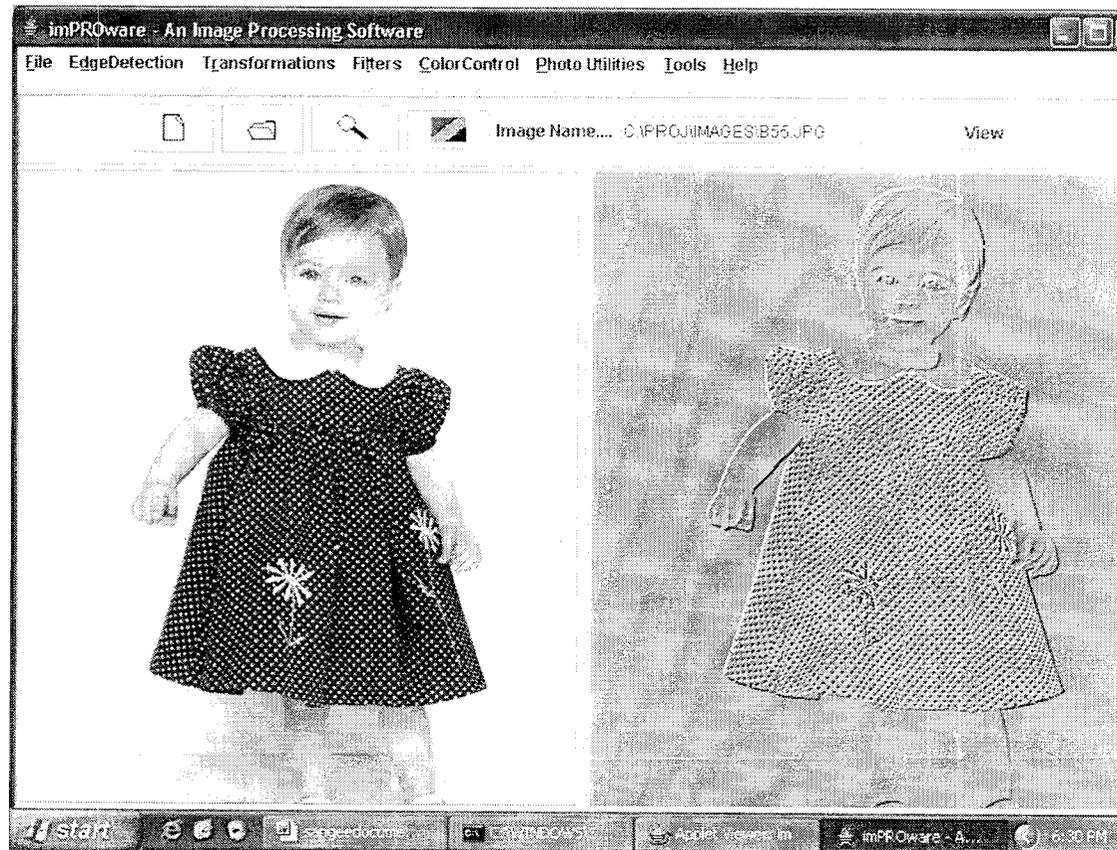
Color Control – Changing green value



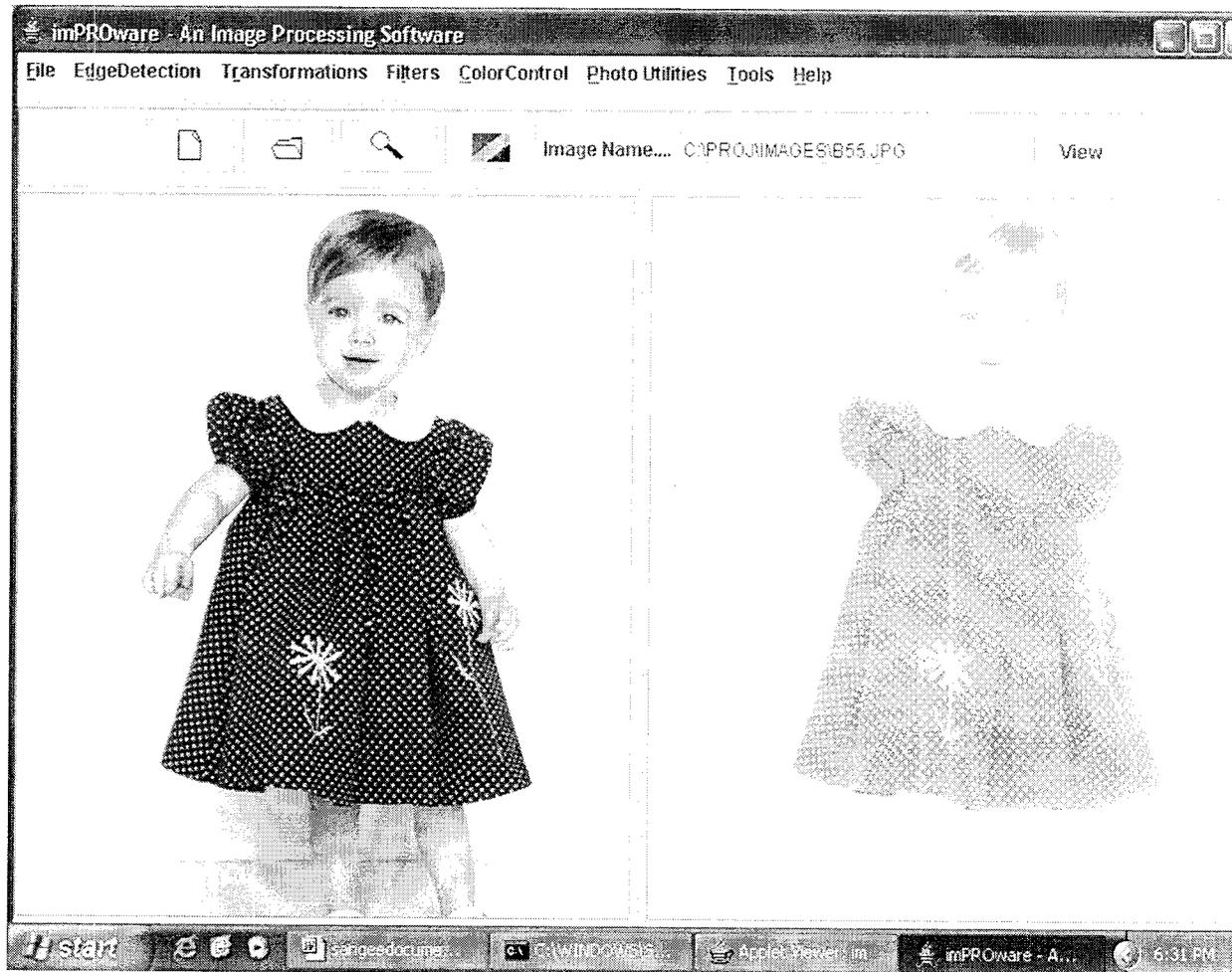
Color Control – Changing blue value

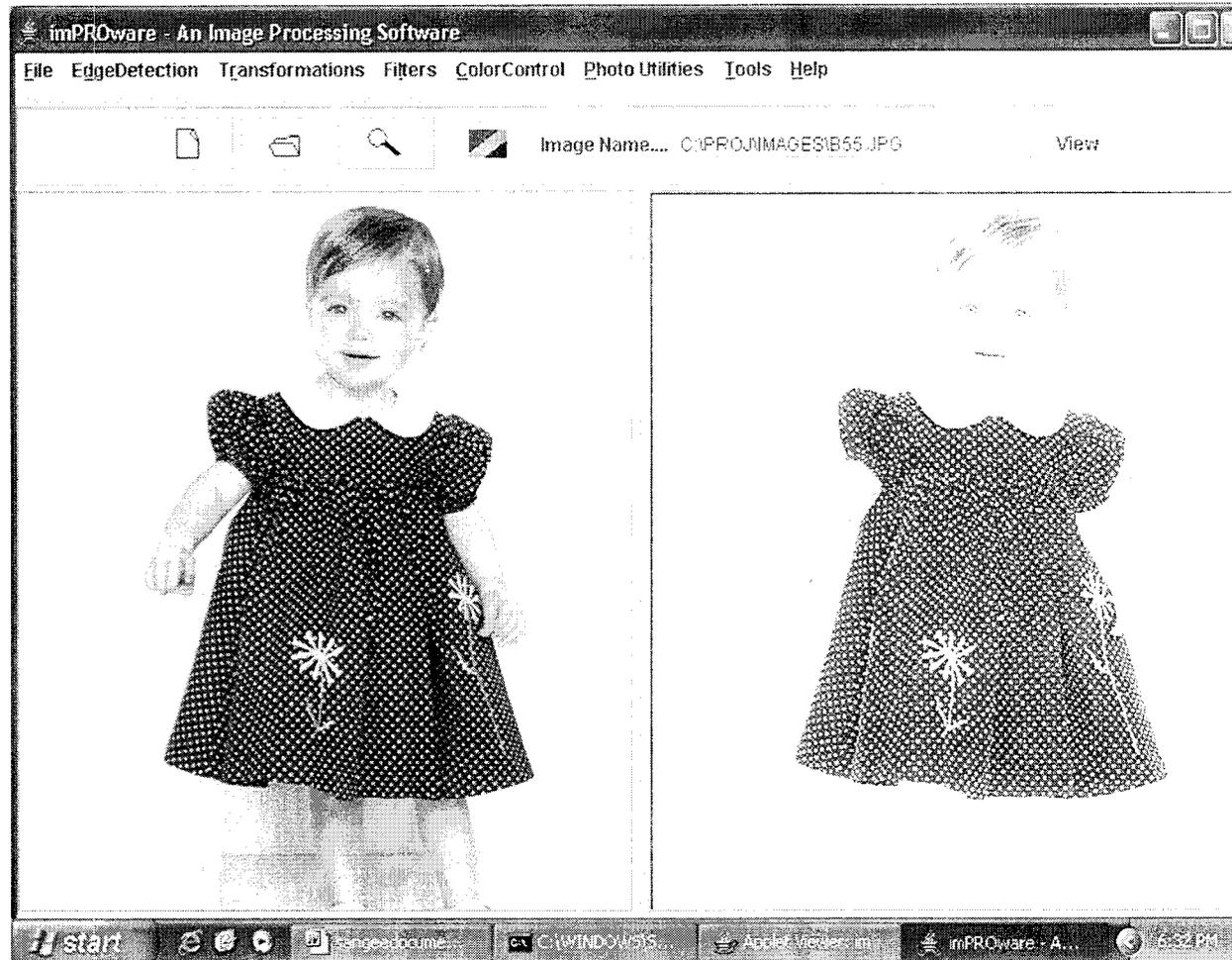


Negative image



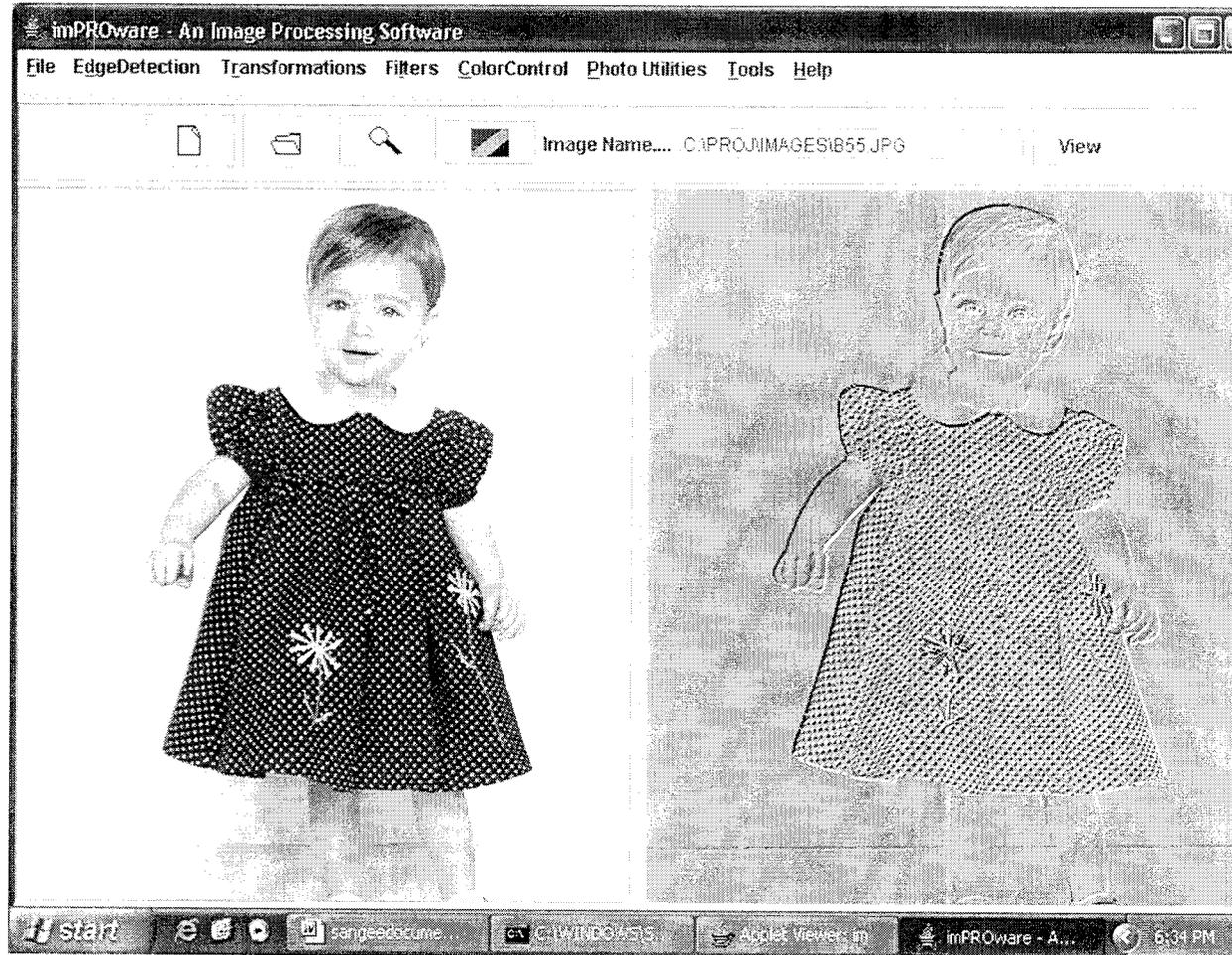
Embossing an image



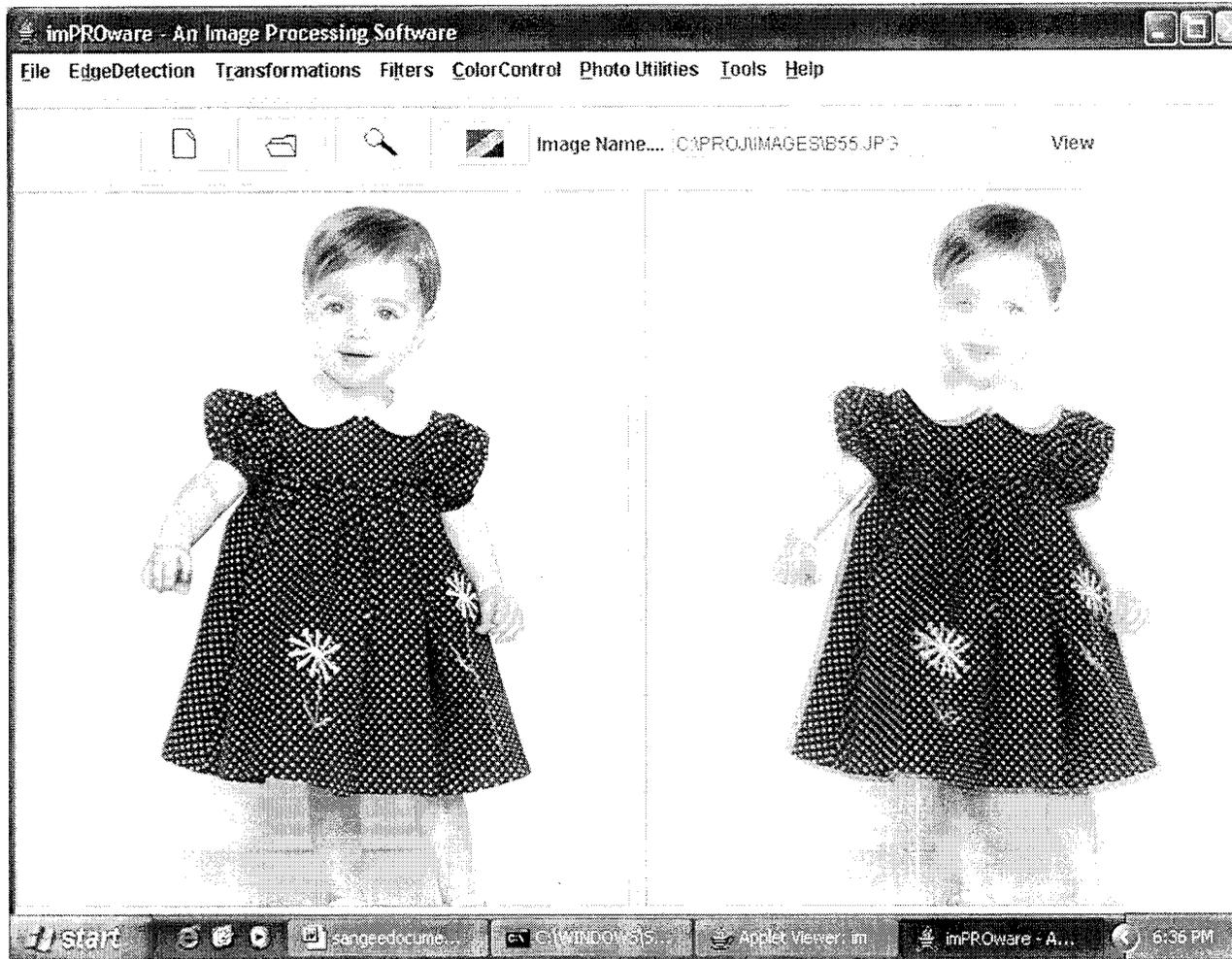




Blurring of an image



Engraving an image



Sweeping



Gray Scale image



Mirror Image

BIBLIOGRAPHY

Text Books

1. Herbert Schildt, “ **JAVA 2 Complete Reference** “, 5th Edition, Tata McGraw Hill Publishing Company
2. Steve Holzmer, “**JAVA 2 Black Book**”, Dreamtech Press
2. Hearn and Baker, “**Interactive Computer Graphics**”, Tata McGraw Hill Publishing Company
3. Roger S Pressman, “**Software Engineering**”, 1997, Tata McGraw-Hill Publishing Company.

Websites

1. <http://www.mathworks.com>
2. <http://www.exforsys.com>
3. www.divx.com
4. www.wikipedia.com