VIRTUAL TRIAL ROOM

The Augmented Reality Technique

By

Abhinav Biswas

Soumalya Dutta

Of

B.Tech

Computer Science & Engineering

Under the guidance of

Mr. Sudarshan Nandy

Computer Science & Engineering Department

JIS College of Engineering

Block-A, Phase-III, Kalyani, Nadia, Pin-741235

West Bengal, India.

JIS College of Engineering

Block-A, Phase-III, Kalyani, Nadia 741235
Phone: 2582 2137, Tele-fax: 2582 2865
Email: info@jiscollege.org

CERTIFICATE

To Whom It May Concern

This is to certify that the Project entitled "**Virtual Trial Room**" is a bona-fide work done by **Abhinav Biswas** [CSE/08/07, 08123001068] and **Soumalya Dutta** [CSE/08/38, 08123001117] of B.Tech, Computer Science & Engineering in fulfilment of Final Year project developed under the guidance of Prof. Sudarshan Nandy, Department of Computer Science & Engineering, JIS College of Engineering.

………………………………	………………………………
Dr. A. Guha	**Prof. Urmibrata Banerjee**
(Director)	(Principal)
JIS College of Engineering	JIS College of Engineering
………………………………	………………………………
Mrs.Pranati Rakhshit	**Mr.Sudarshan Nandy**
(HOD of CSE Dept.)	(Project Guide)
JIS College of Engineering	JIS College of Engineering

ACKNOWLEDGEMENT

A project work is not an obligation done by one person towards a predetermined specific goal. Rather, it is the coming together of many elements, some direct and some indirect, towards the planning and implementation of a long drawn effort & above all the support of its members, what we call as team work.

We wish to express our sincere gratitude to Prof. Sudarshan Nandy, Department of Computer Science & Engineering, JIS College of Engineering for providing us the opportunity to proceed with this project .We sincerely thank him for his guidance and encouragement in carrying out the project work.

We express our deep sense of gratitude to all faculty members and other laboratory staff of our college who rendered their help during the project work session towards the completion of our project.

CONTENTS

VTR

1. INTRODUCTION

An application that:

- Portrays an augmented view of the user
  - in a dynamic real-time Live Video mode
  - with virtual superimposed clothes
- Implements a virtual mirror
  - with interactive digital options onscreen
- Immediately allows the user to see
  - if a clothing's color/style suits his/her body
  - before he can actually decide to buy it

Let's take an overview:

- User stands in front of the digital screen
  - Various onscreen options get displayed
  - Like Category (Men, Women, Kids).
- Selects option using simple Hand Motion
  - Then the augmented image with virtual clothes is shown.
- Adjusts the placement of the cloth item
  - With onscreen options like Zoom In/Out, Up/Down etc.
- Finally take a snap, save or print it.

❖Also now the user can decide whether the cloth is worth a buy.

So, the basic objective of the overall system is to introduce an intuitive way at the various retail stores to allow the end-user to try new cloth items virtually using the power of Augmented Reality.

It's fun to use, fast & effective and has a potential to increase prospective customers.

2. LIST OF REQUIREMENTS

1

Hardware Requirements:

- CPU Type & Speed
  - Intel Core 2 Duo 1.8 GHz or equivalent
- Hard Disk Space
  - 2 GB
- Memory
  - 1 GB
- Graphics
  - Any 32MB DirectX 11 compatible graphics system
- Webcam
  - 640X480 resolution or higher
- Printer
  - Optional (recommended)

Software Requirements:

- Operating System
  - Windows 7 or above
- Java Runtime Environment
  - JRE 1.5 or above
- Microsoft Visual C++ Redistributable_x86
  - 1 GB
- OpenCV Library Package
  - Super-pack 2.3.1 or above

3. SYSTEM SYNOPSIS

S. No.	Particular	Description
1.	Language/Platform	Java, jdk 1.5
2.	IDE	Eclipse 3.6, Helios
3.	Native Image Library	OpenCV-2.3.1 Super-Pack
4.	OpenCV Wrapper	JavaCV

The details of the particulars are described in the following sections.

- Why Java??
- Eclipse – The Wonder Tool
- Magic of OpenCV
- JavaCV – The Wrapper

3.1 Why Java??

➢ Platform Independent and Portable

- Java program can be easily moved from one computer system to another.
- Changes and upgrades in the operating system & system resources will not force any changes in java programs.

➢ Object Oriented

- Java is a true object oriented language. Almost everything in java is an object.

➢ Multi Threaded & Interactive:

- Java supports multi-threaded programs. The Java Runtime comes with tools that support multiple processes synchronization.
- Construction of smoothly running interactive systems is easy yet powerful.

➢ Dynamic & Extensible:

- Java is capable of dynamically linking new class libraries, methods and objects.
- Java program can access native functions written in C/C++ using various wrappers.

VTR

3.2 Eclipse – The Wonder Tool

Eclipse is a multi-language software development tool comprising an Integrated Development Environment (IDE) and an extensible plug-in system. It is written mostly in Java. It can be used to develop applications in Java and, by means of various plug-ins, other programming languages including Ada, C, C++, COBOL, Haskell, Perl, PHP, Python, R, Ruby (including Ruby on Rails framework). The development environment includes the Eclipse Java development tools (JDT) for Java, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others, by default.

3.3 Magic of OpenCV

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real time computer vision, developed by Intel and now supported by Willow Garage. It is free for use under the open source BSD license. The library is cross-platform. It focuses mainly on real-time image processing. If the library finds Intel's Integrated Performance Primitives on the system, it uses proprietary optimized routines to accelerate itself.

Features of OpenCV:

- 2D and 3D feature toolkits
- Egomotion estimation
- Facial recognition system
- Gesture recognition
- Human–computer Interaction (HCI)
- Motion understanding
- Object identification
- Segmentation and Recognition
- Stereopsis Stereo vision: depth perception from 2 cameras
- Structure from motion (SFM)
- Motion tracking, etc.

3.4 JavaCV – The Wrapper

JavaCV provides wrappers to commonly used image-processing libraries in the field of computer vision viz. OpenCV, FFmpeg, OpenKinect, ARToolKitPlus, etc. The classes found under the com.googlecode.javacv.cpp package namespace expose their complete APIs. Moreover, utility classes make their functionality easier to use on the Java platform, including Android.

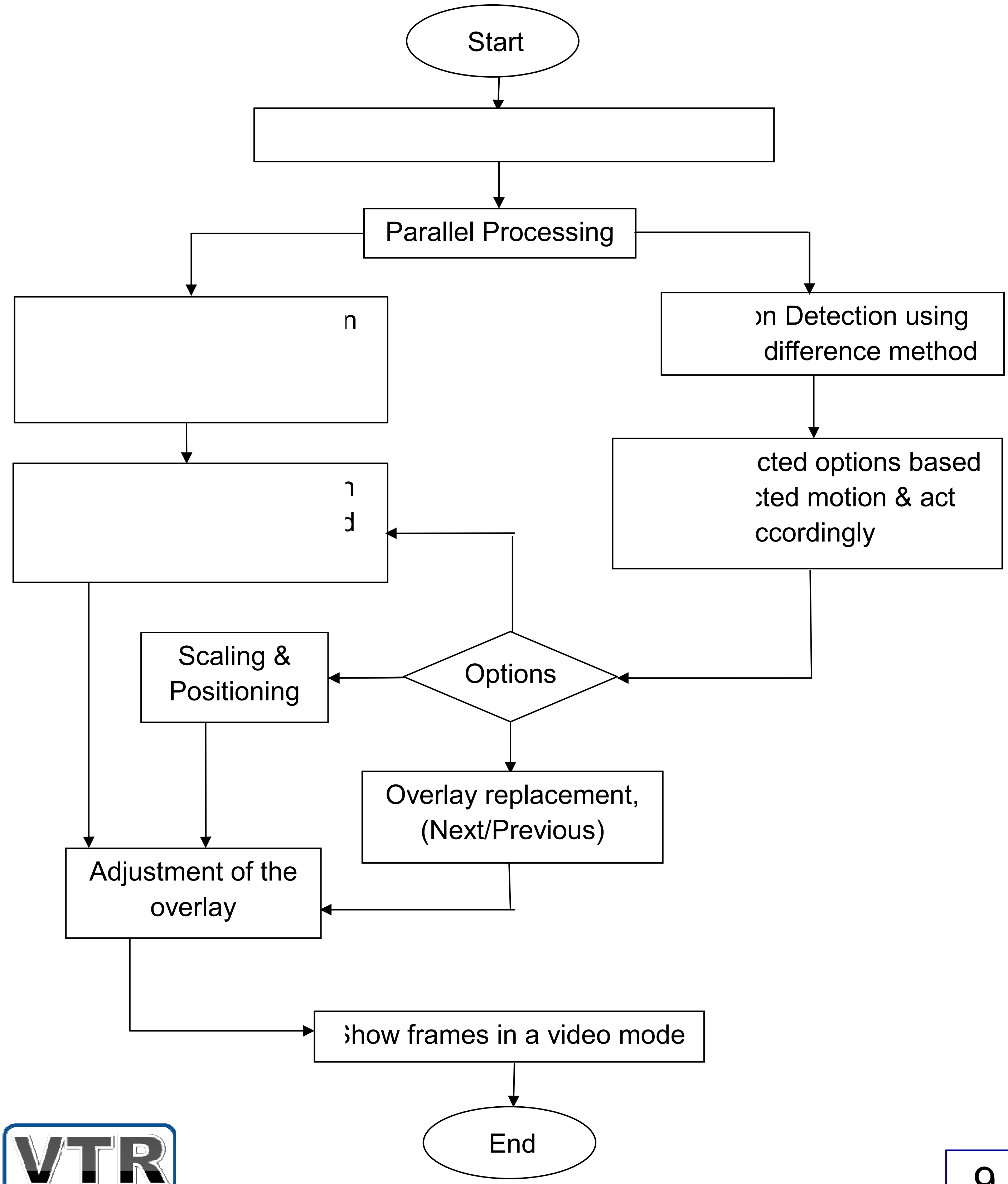
JavaCV also comes with hardware accelerated full-screen image display (CanvasFrame & GLCanvasFrame), easy-to-use methods to execute code in parallel on

multiple cores (Parallel processing), user-friendly geometric and color calibration of cameras and projectors (GeometricCalibrator, ProCamColorCalibrator ProCamGeometricCalibrator,), detection and matching of feature points (ObjectFinder), a set of classes that implement direct image alignment of projector-camera systems (mainly GNImageAligner, ProjectiveTransformer, ProjectiveColorTransformer, ProCamTransformer, and ReflectanceInitializer), as well as miscellaneous functionality in the JavaCV class. Some of these classes also have an OpenCL and OpenGL counterpart, their names ending with CL, i.e.: JavaCVCL, etc. except for GLCanvasFrame.

4. SYSTEM METHODOLOGY

In this section let's get deep into the working of the actual system and understand the whole process work-flow.

4.1 Step by Step System Design



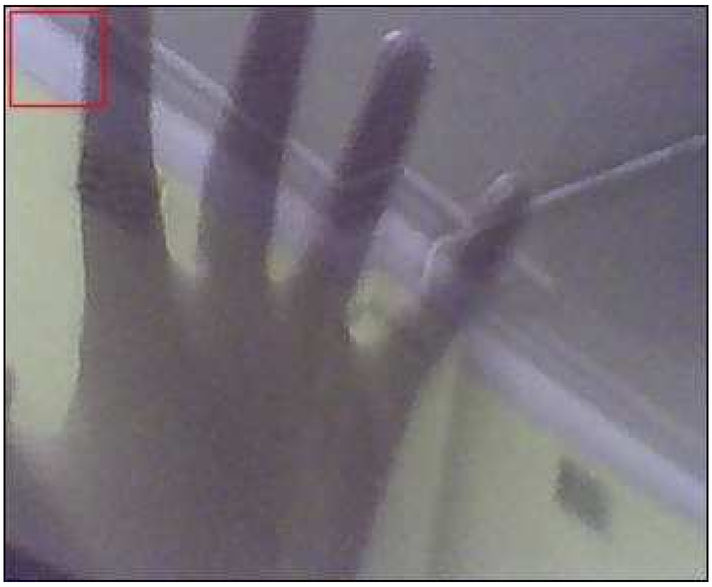
Start
Parallel Processing
n
ɔn Detection using difference method
ן
ɖ
cted options based ːted motion & act ccordingly
Scaling & Positioning
Options
Overlay replacement, (Next/Previous)
Adjustment of the overlay
ʖhow frames in a video mode
End

5. ANALYSIS OF CODE

5.1 Grab video frame & Flip it

- Our first aim was to detect video frames from the webcam. We used OpenCVFrameGrabber class to perform this task.
- Thereafter cvFlip() method is used to perform lateral shift operation on the image frame to convert it to a digital mirror.

Output:



(It's capturing)

5.2 Implement Face Detection

- **Face Detection:**
  - It is nothing but to detect face(s) in an image and mark its position on the image. Here we used haarcascade classifiers for face detection.

- **What is Haarcascade classifier??**
  - It is an XML file which contains trained data of the samples.

Let's illustrate this, how to create the HaarCascade classifier for face detection. It consists of the following steps:

- Create description file of positive samples
- Create description file of negative samples
- Pack the positive samples into a vector file
- Train the classifier
- Convert the trained cascade into a cascade file
- Use the xml to detect Object

✓Taking a Large No Samples :

- First of all, we need a large number of images of our object. All the positive and negative images, which can run into thousands.

VTR

- Here we have used ffmpeg to perform this operation in the following way by executing this command:

```
ffmpeg -i Video.mpg image-No%d.bmp
```

- There after we put the positive and negative samples in another directory and create the description file for both types of samples to train the classifier.

✓Creating the Description file :

- After the successful execution of objectmarker code, we get the description file for both kind of samples with the following attributes:
    - The description file is just a text file, with each line corresponding to each image. The fields in a line of the positive description file are: the image name, followed by the number of objects to be detected in the image, which is followed by the (x, y) coordinates of the location of the object in the image. Some images may contain more than one objects.
    - Description file for the negative samples is created by listing the contents of negative samples folder and redirecting the output to a text file by executing this command:

```
ls  > negative.txt
```

✓Packing the positive samples into a vector file :

- All the positive images in the description file are packed into a .vec file. It is created using the createsamples utility provided with opencv package. Its synopsis is:

```
opencv-createsamples -info positive.txt -vec
vecfile.vec -w 30 -h 35
```

✓Training the classifier:

- It's the most time consuming phase of Haarcascade creation. It is performed by executing the following command:

```
opencv-haartraining -data haar -vec vecfile.vec  -
bg negative.txt -nstages 30 -mem 2000 -mode all
-w 30 -h 35
```

- It puts the data in the haar directory.
- Now, it's the time to generate the haarcascade Xml file which will be used for object detection.

✓Converting the trained cascade into a cascade file :

- Once the training is over, we are left with a folder full of training data (named haar in our case). The next step is to convert the data in that directory to an xml file. It is done using the convert_cascade program given in the opencv samples directory.

- It is performed by executing the following command :

```
$convert_cascade –size="30x35"   data
haarcascade_face.xml
```

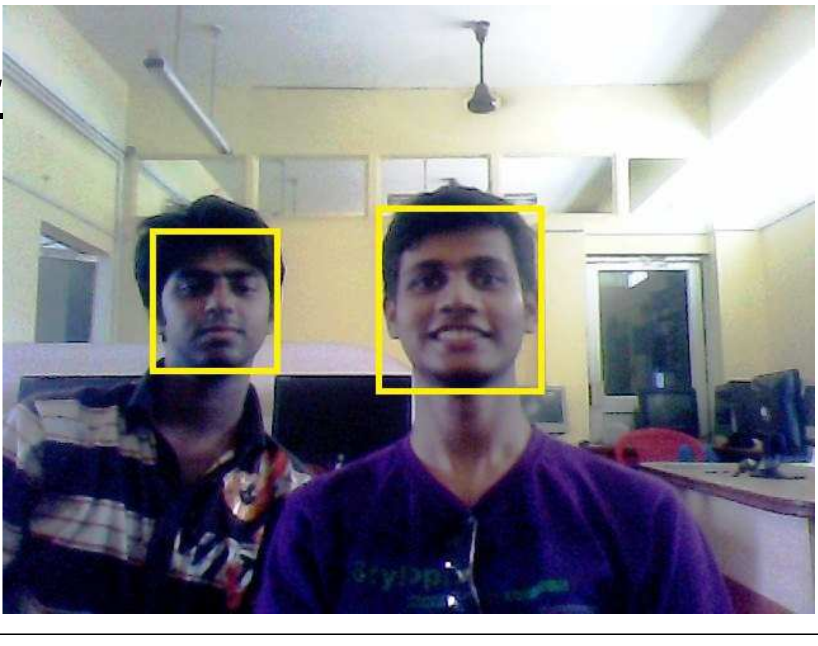
- Now, it's the time to detect face with this generated haarcascade xml file.

✓Face Detection with haarcascade file :

- Face Detection is performed with the help of haarcascade file by using the following piece of code:

```
CvHaarClassifierCascade cascade = new
CvHaarClassifierCascade(
cvLoad(CASCADE_FILE));
 CvSeq faces = cvHaarDetectObjects(grayImage,
cascade, storage, 1.1, 1, 0);
//We iterate over the discovered faces and draw yellow
rectangles around them.
for (int i = 0;i< faces.total; i++) {
        CvRect r = new CvRect(cvGetSeqElem(faces, i));
        cvRectangle(originalImage, cvPoint(r.x, r.y),
        cvPoint(r.x + r.width, r.y + r.height),
       CvScalar.YELLOW, 1, CV_AA, 0);
}
```

Output:



5.3 Super impose virtual Cloth item:

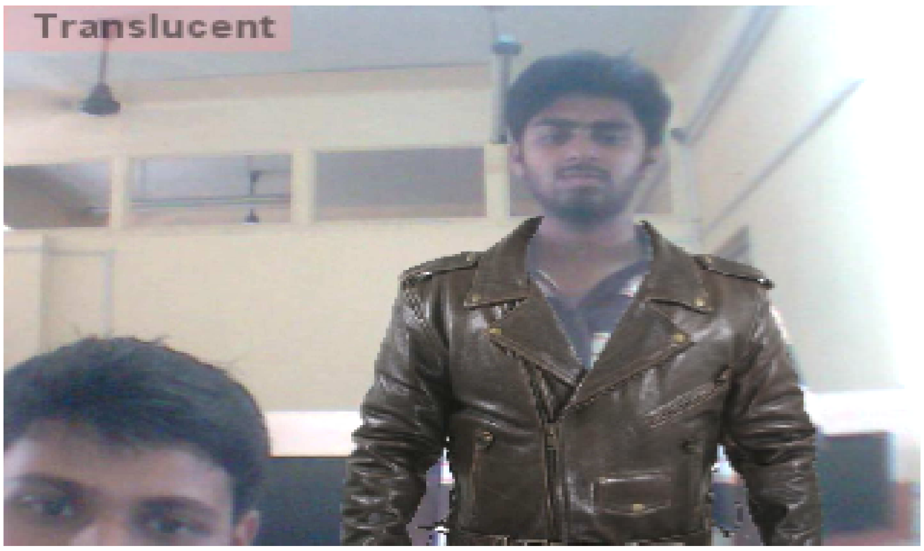
- After successful face detection, it's the time to superimpose virtual clothes on the frames. It's performed by importing java.awt.* packages in the following way:

```
Graphics g=overlaidImage.getGraphics();
Graphics2d g2d=(Graphics2D)g;
g2d.drawImage(grabbedImage,0,0,width,height,null);
g2d.drawImage(cloth_sample_object,x,y,w,h,null);
```

where,

x->Calculated Horizontal Position
y->Calculated Vertical Position

- **Output :**



5.4 Scaling & Positioning:

✓Scaling :

Here Scaling operation is performed to resize & scale the cloth image using the following

Mathematical concept:

A scaling transformation alters the size of the object.

X' = Sx.X

Y' = Sy.Y

where,

(X, Y) -> Co-ordinate of Vertex initially

Sx, Sy -> Scaling Factors

(X', Y') -> (Transformed Coordinate)

$$\begin{pmatrix} X' \\ Y' \end{pmatrix} = \begin{pmatrix} Sx & 0 \\ 0 & Sy \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix}$$

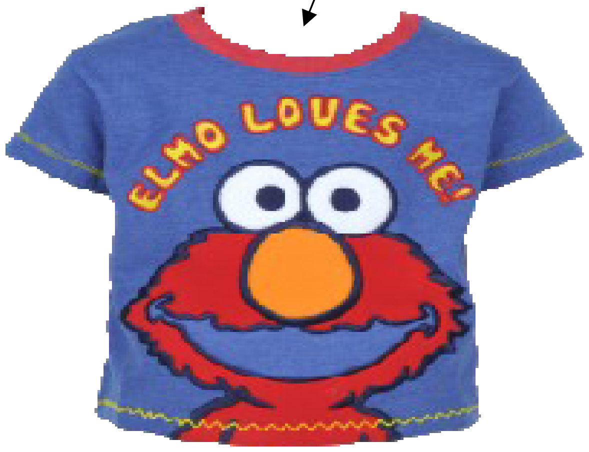
- **<u>Output :</u>**

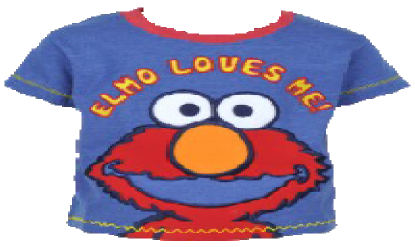
ELMO LOVES ME!

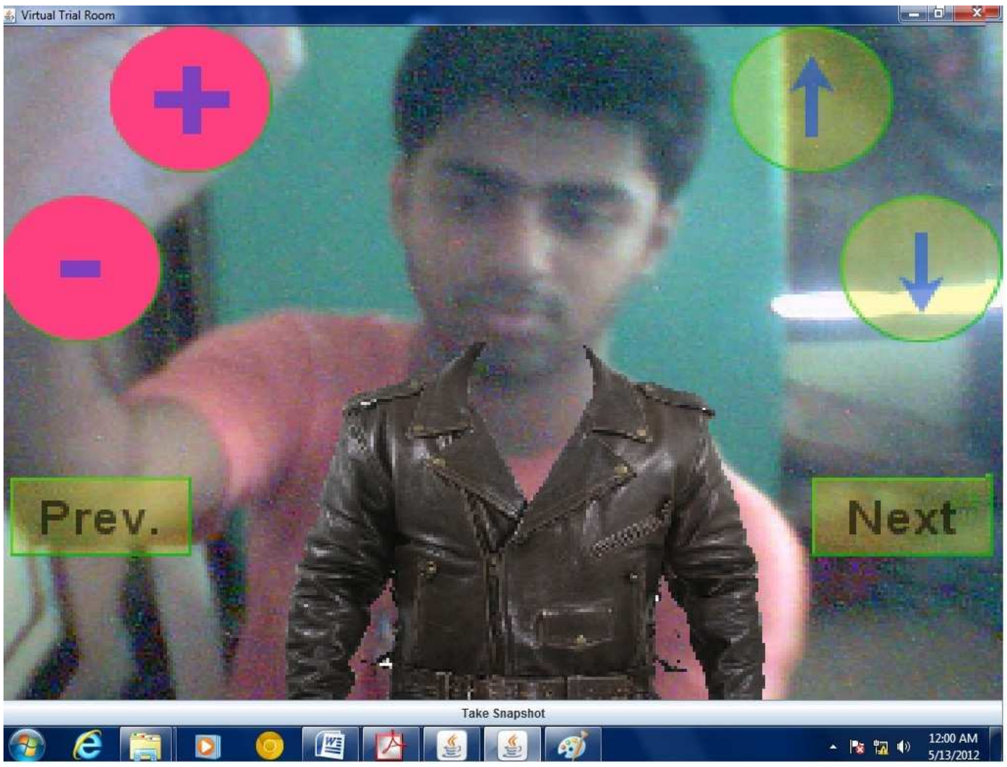
Initial image

(Positive Scaling)

(Negative Scaling)

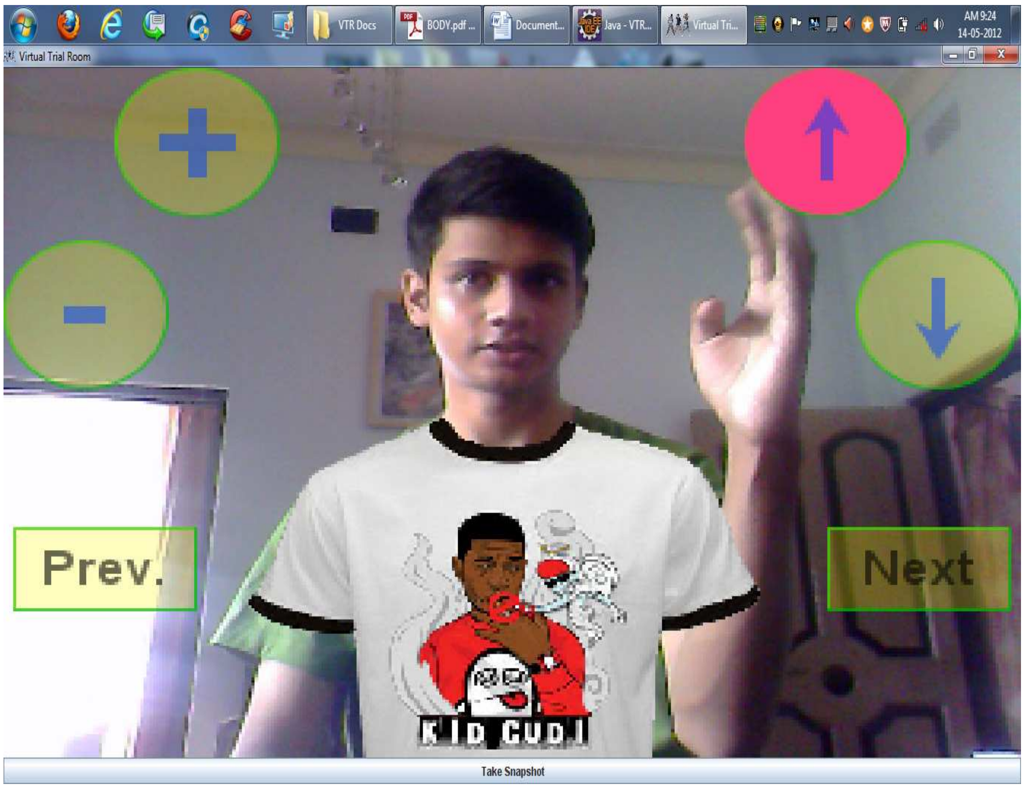






Zoom (+/-)

✓ Positioning :



↑ ↓ Up/Down(↑/↓)

5.5 Motion Detection

The user is provided onscreen options. The selections are actually performed by detecting motion in the circled/rectangle segment of the screen.

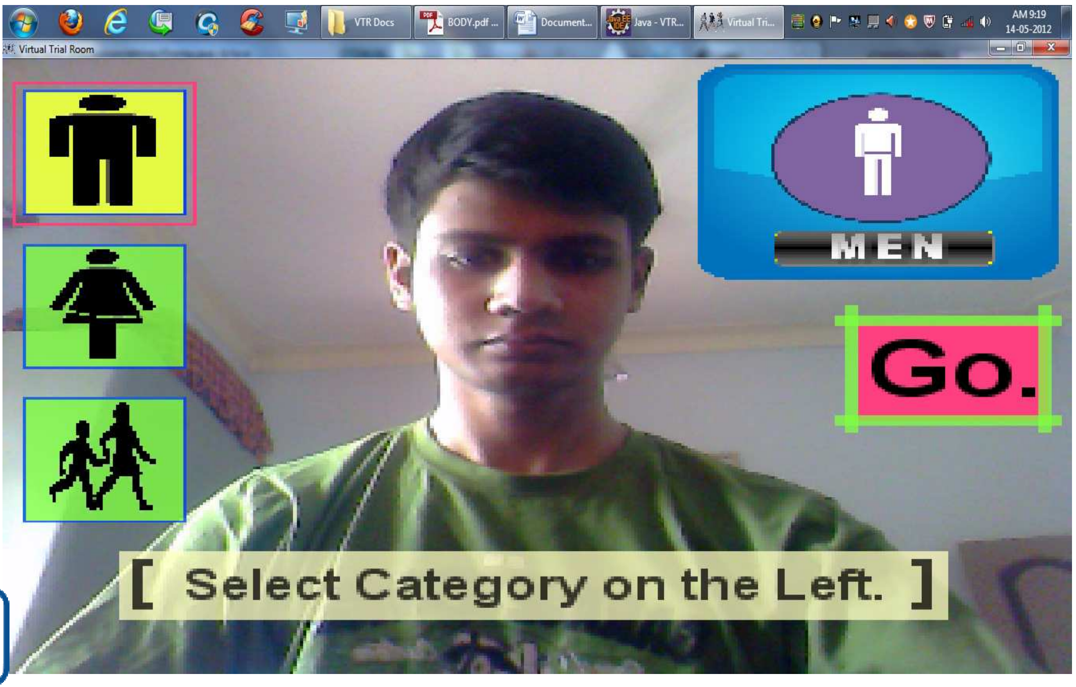
Motion Detection using OpenCV

The basic idea of detecting motion in video frames is by finding out the Absolute difference between two consecutive frames. It is performed using JavaCV in the following way:

```
cvAbsDiff(currentframe, prevframe, diff);
// do some threshold for wipe away useless details
cvThreshold(diff, diff,Threshold_Value,255,
    CV_THRESH_BINARY);
```

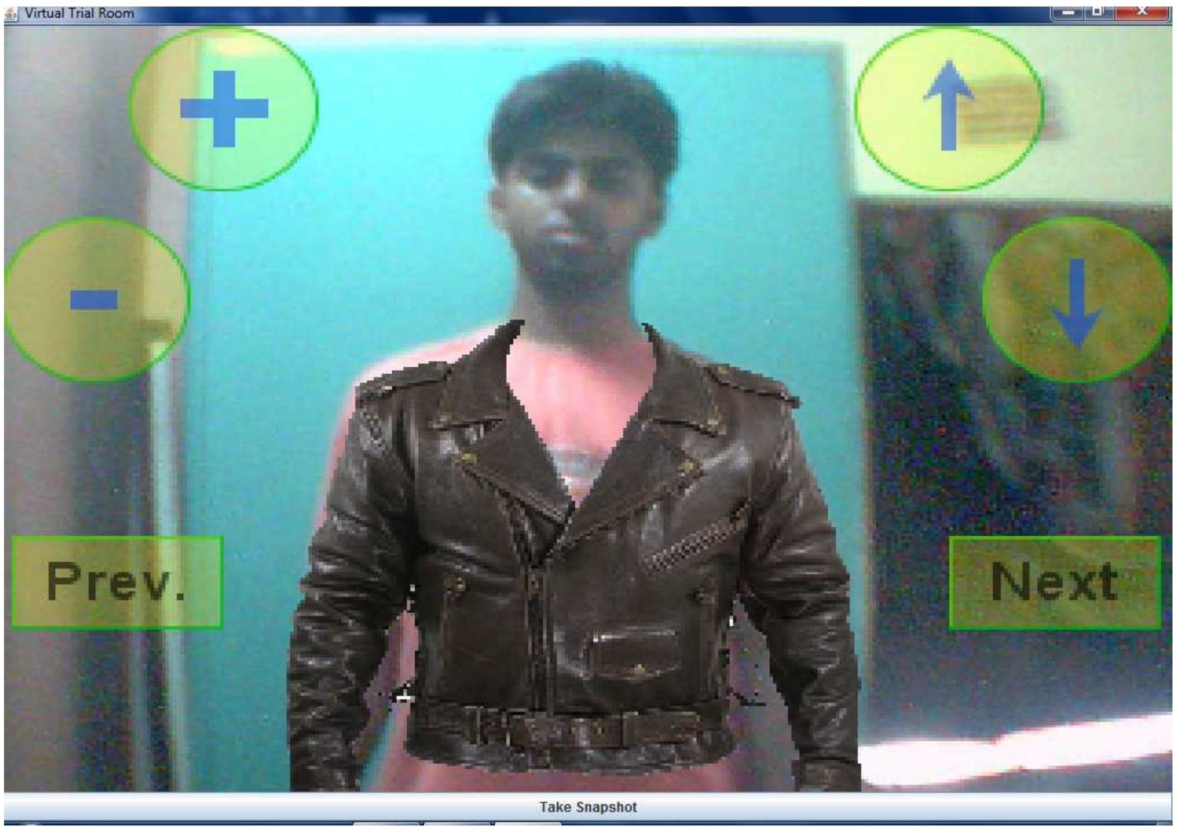
where,

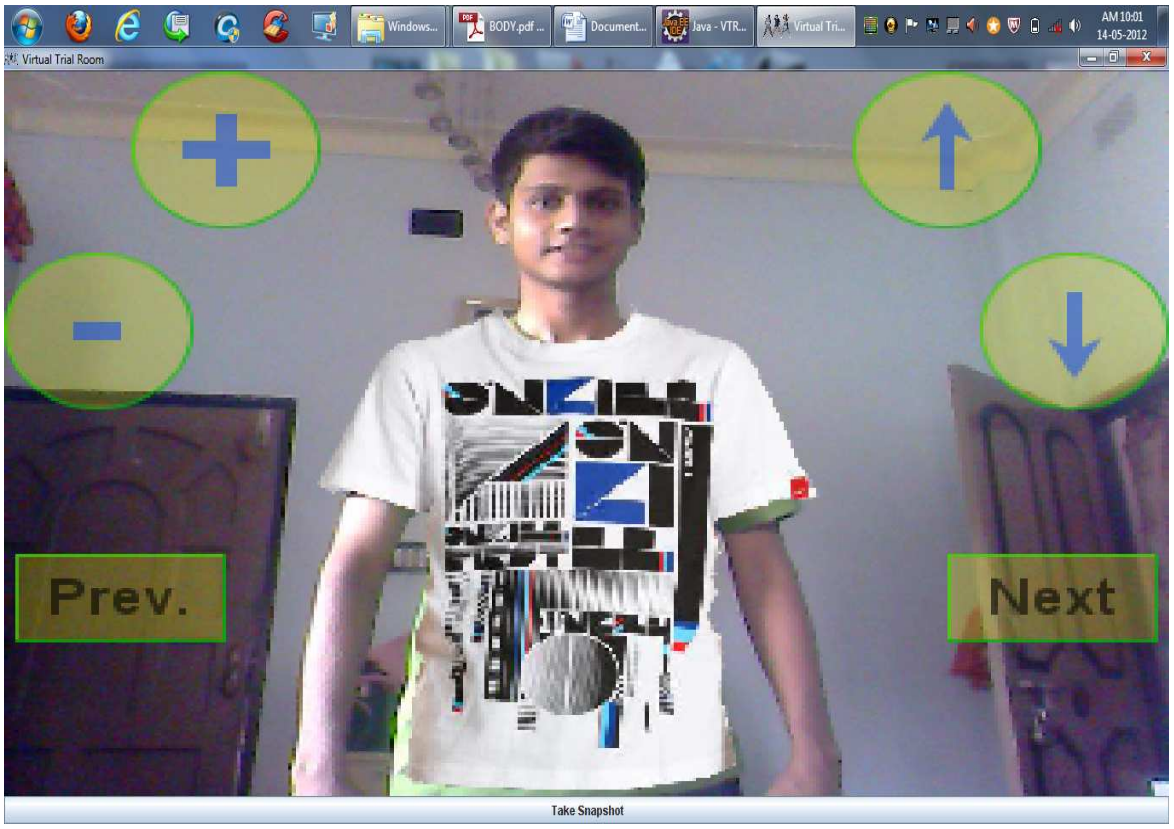
- Threshold_Value -> Amount of Threshold Motion.
- CV_THRESH_BINARY -> Convert whole Image to binary to observe the result properly.



VTR

5.6 Final Output





VTR

6. FUTURE SCOPE

This application has a wide scope on implementation in the Retail Industry. We are currently working on it to append it to a Web application for online shopping websites to provide Online Trial Room support.

A bunch of new features can be added like,

- one-click share on Facebook.com
- virtual mirror for eye-wear (sunglasses) & necklace
- Automated smile detection for easy snapshot.
- Auto-suggestion of cloth items, etc.

So, in short, imagination is the limit. We can exploit this intuitive technology in various ways in different fields.

7. CONCLUSION

Virtual Trial Room is an image processing based Augmented Reality (AR) software developed for the retail industry to give a virtual support of trial room. It portrays an augmented view of the user in a dynamic real-time Live Video mode with virtual superimposed clothes, by implementing a virtual mirror with interactive digital options onscreen. It immediately allows the user to see if the clothing's color/style suits his/her body before he can actually decide to buy it

Though the concept may seem simple & intuitive but actually it's a power of various complex algorithms running on the backend with parallel processing. It has been developed by blending various concepts like Face Detection, Image Scaling, Motion Detection etc. with a plethora of technologies like Java Swing, OpenCV etc.

8. BIBLIOGRAPHY

- Learning OpenCV – O'Reilly
  - Gary Bradski
- Head First Java – O'Reilly
  - Cathe Siera
- JavaCV – Google Project Hosting
  - http://code.google.com/p/javacv
- StackOverflow
  - http://www.stackoverflow.com