

IMAGE AUGEMENTATION

Submitted in partial fulfillment of the requirements
for the degree of
BACHELOR OF TECHNOLOGY
(CSE & IT)

Under the Supervision of
Mr. Punit Gupta
by

Nitish Marwah(101344)



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY
WAKNAGHAT, SOLAN
HIMACHAL PRADESH
INDIA
2011

Certificate

This is to certify that project report entitled “IMAGE AUGEMENTATION”, submitted by NITISH MARWAH (101344) in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date:

24-12-2014

Signature in full of Supervisor:

Name : Mr. PUNIT GUPTA

Designation: Asstt. Prof of CS/IT Department

DECLARATION

I hereby declare that the project work entitled “Image Augmentation” submitted to the JUIT, is a record of an original work done by me under the guidance of Mr. Punit Gupta, Faculty member, JUIT, and this project work has not performed the basis for the award of any Degree associateship and similar project if any.

NITISH MARWAH

[101344]

ACKNOWLEDGEMENT

The project entitled as Image augmentation. The main purpose is to make a game based on the concept of image augmentation in Java. The project is the result of my endurance in best of ways for these couple of months.

My project guide Mr. Punit Gupta, an Asstt. Professor for Department of Computer Science & IT at Jaypee University of Information Technology, has indeed guided me to proceed with this project in best of his capability and enlightened me with his knowledge and wisdom. He has given me the push that I needed to go forward and do more. So I take this opportunity to thank him for his lending help and cooperation. May he shower such wisdom and guidance to future individual.

Date: 16th Dec, 2014

Name of the student: Nitish Marwah (101344)

CONTENT

| | |
|-------------------------------|----|
| 1.1 IMAGE AUGMENTATION | 6 |
| 1.2 USE OF OPEN CV IN JAVA | 7 |
| 1.3 HOW TO USE OPENCV IN JAVA | 13 |
| 2 REVIEW STUDY | |
| 2.1 MEDICAL | 21 |
| 2.2 MANUFACTURING AND REPAIR | 25 |
| 2.3 ANNOTATION | 26 |
| 2.4 ROBOT PATH | 27 |
| 2.5 ENTERTAINMENT | 27 |
| 2.6 MILITARY AIRCRAFT | 28 |
| 3.1 PROBLEM STATEMENT | 29 |
| 4.1 PROPOSED SOLUTION | 30 |
| 4.2 REVIEW STUDY :OSMO | 32 |
| 5 HOW TO USE OPENCV IN JAVA | 35 |

LIST OF FIGURES

| S. NO. | Table Description | Page No. |
|---------------|-------------------------------------------------|-----------------|
| 1. | Image Enhancement | 15 |
| 2. | Showing Gray Scale Intensity Graph | 16 |
| 3. | Showing Variation In Noise With S | 17 |
| 4. | Tic Tac Toe | 21 |
| 5. | Virtual Foetus Inside Womb Of Pregnant Women | 24 |
| 6. | Mockup Of Breast Tumour Biopsy | 24 |
| 7. | External View Of Columbia Printer | 25 |
| 8. | Boeing Prototype Wire Assembly | 26 |
| 9. | Virtual Lines Display Geometry As Seen In Orbit | 27 |
| 10. | Robot Use | 28 |
| 11. | Tangram | 32 |
| 12. | Newton | 33 |
| 13. | Words | 33 |
| 14. | OpenCv In Java | 36 |
| 15. | OpenCv In Java | 36 |
| 16. | OpenCv In Java | 36 |

ABSTRACT

Augmented reality (AR) is a live direct or indirect view of a physical, real-world environment whose elements are augmented (or supplemented) by computer-generated sensory input such as sound, video, graphics or GPS data. It is related to a more general concept called mediated reality, in which a view of reality is modified (possibly even diminished rather than augmented) by a computer.

Merchlar's mobile game Get On Target uses a trigger image as fiduciary marker

Augmented reality allows gamers to experience digital game play in a real world environment. In the last 10 years there has been a lot of improvements of technology, resulting in better movement detection and the possibility for the Wii to exist, but also direct detection of the players movements.

CHAPTER 1

1. INTRODUCTION

1.1 What is OpenCV?

This might be the 'basic' question that comes first to your mind. Well, it stands for '**Open Source Computer Vision Library**' initiated by some enthusiast coders in '1999' to incorporate Image Processing into a wide variety of coding languages. It has C++, C, and Python interfaces running on Windows, Linux, Android and Mac.

Before, I get into the use of OpenCV; Let's get familiar with the same. As, by now you must have understood that it is a '**library**', so to use the functions available in the library you would need a compiler.

1.2 What is OPENCV Used for?

-HUMAN COMPUTER INTERACTION (HIC)

-OBJECT IDENTIFICATION

-OBJECT RECOGNITION

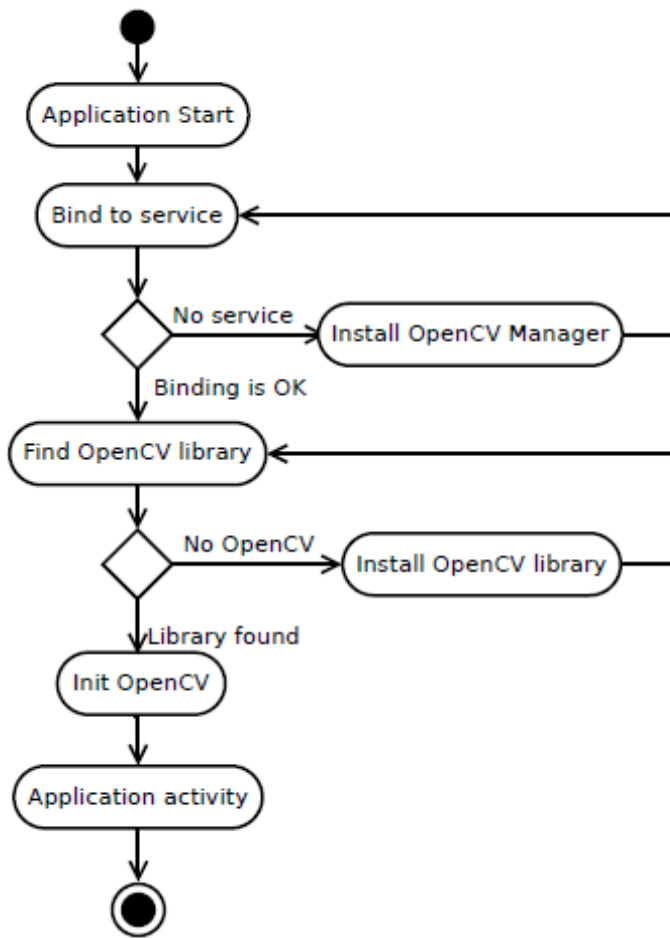
-FACE RECOGNITION

-GESTURE RECOGNITION

-MOTION TRACKING

-IMAGE PROCESSING

-MOBILE



OPEN CV LOADER MECHANISM

To start off, you need to install '**OpenCV**' and a '**compiler**' and then establish a linkage between the two (i.e. Compiler is able to use the functions available with the library)

Configuring Eclipse

First, obtain a fresh release of OpenCV from download page and extract it under a simple location like C:\OpenCV-X.X.X\ . I am using version 2.4.6, but the steps are more or less the same for other versions.

Now, we will define OpenCV as a user library in Eclipse, so we can reuse the configuration for any project.

Launch Eclipse and select Window → Preferences from the menu.

Navigate under Java → Build Path → User Libraries and click New....

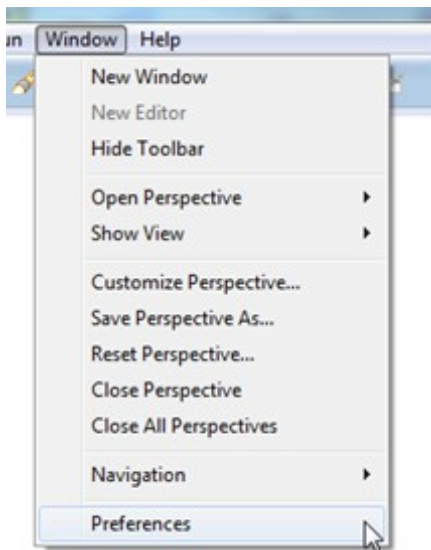


FIGURE 14

Enter a name, e.g. OpenCV-2.4.6, for your new library.

Now select your new user library and click Add External JARs....

Browse through C:\OpenCV-2.4.6\build\java\ and select opencv-246.jar.

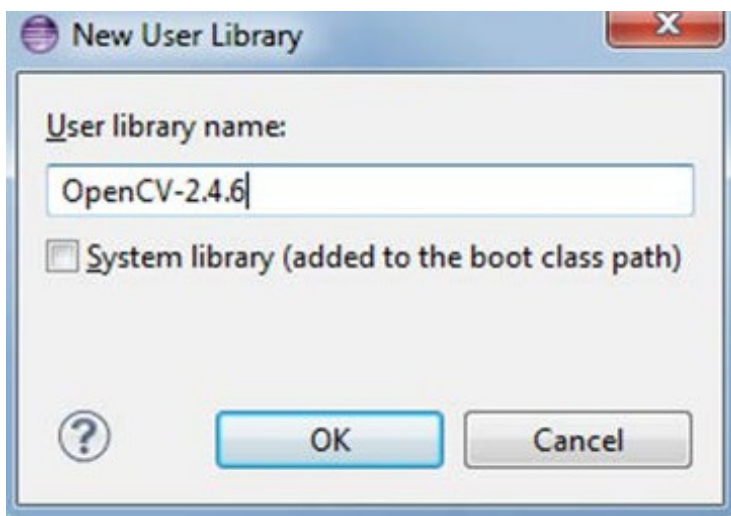


FIGURE 15

After adding the jar, extend the opencv-246.jar and select Native library location and press Edit....

Select External Folder... and browse to select the folder

C:\OpenCV-2.4.6\build\java\x64.

If you have a 32-bit system you need to select the x86 folder instead of x64.

Your user library configuration should look like this:

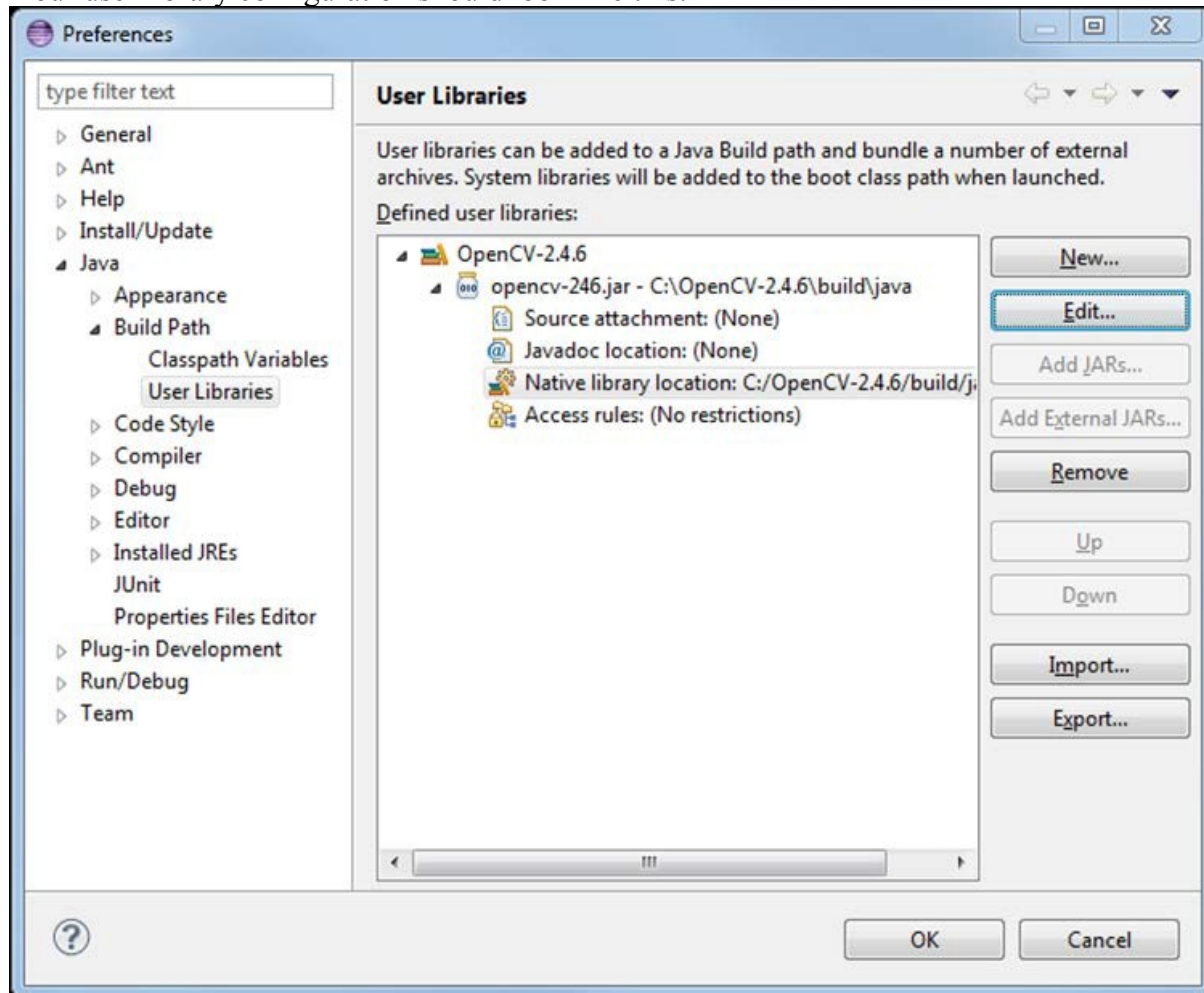


FIGURE 16

Testing the configuration on a new Java project

Now start creating a new Java project.

On the Java Settings step, under Libraries tab, select Add Library... and select User Library -> OpenCV-2.4.6, then click Finish.

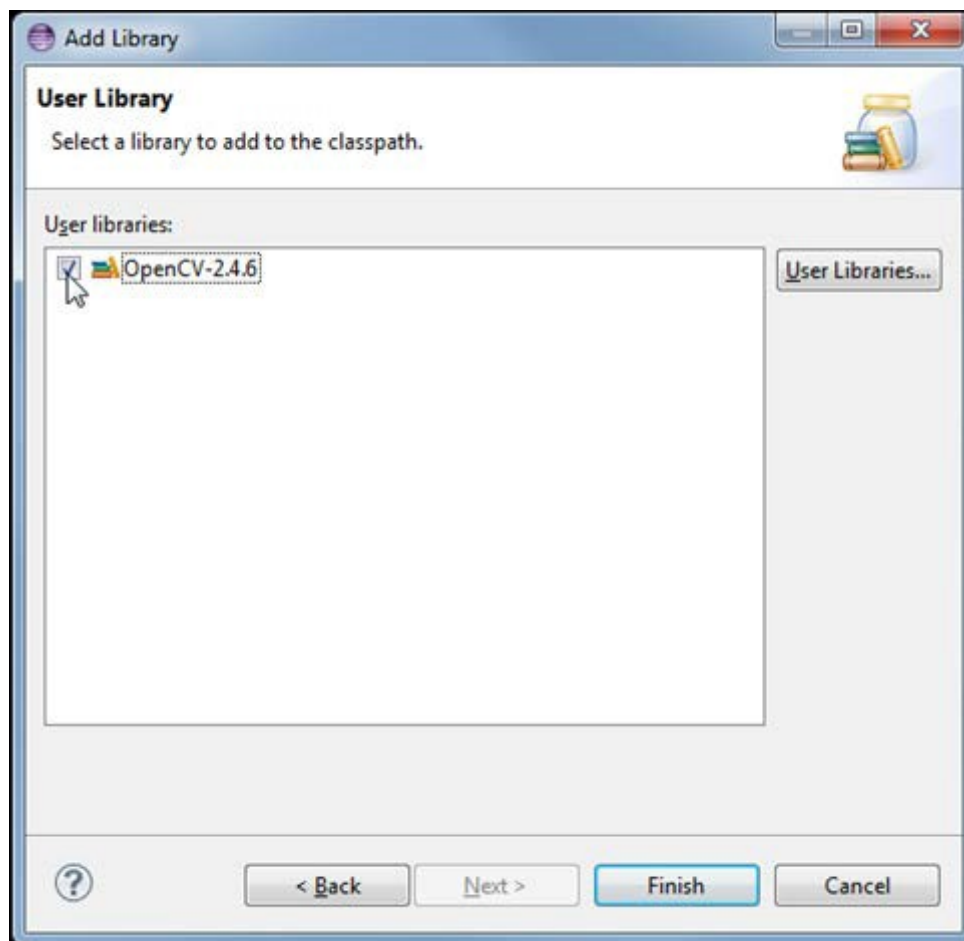


FIGURE 17

Library should look like this:

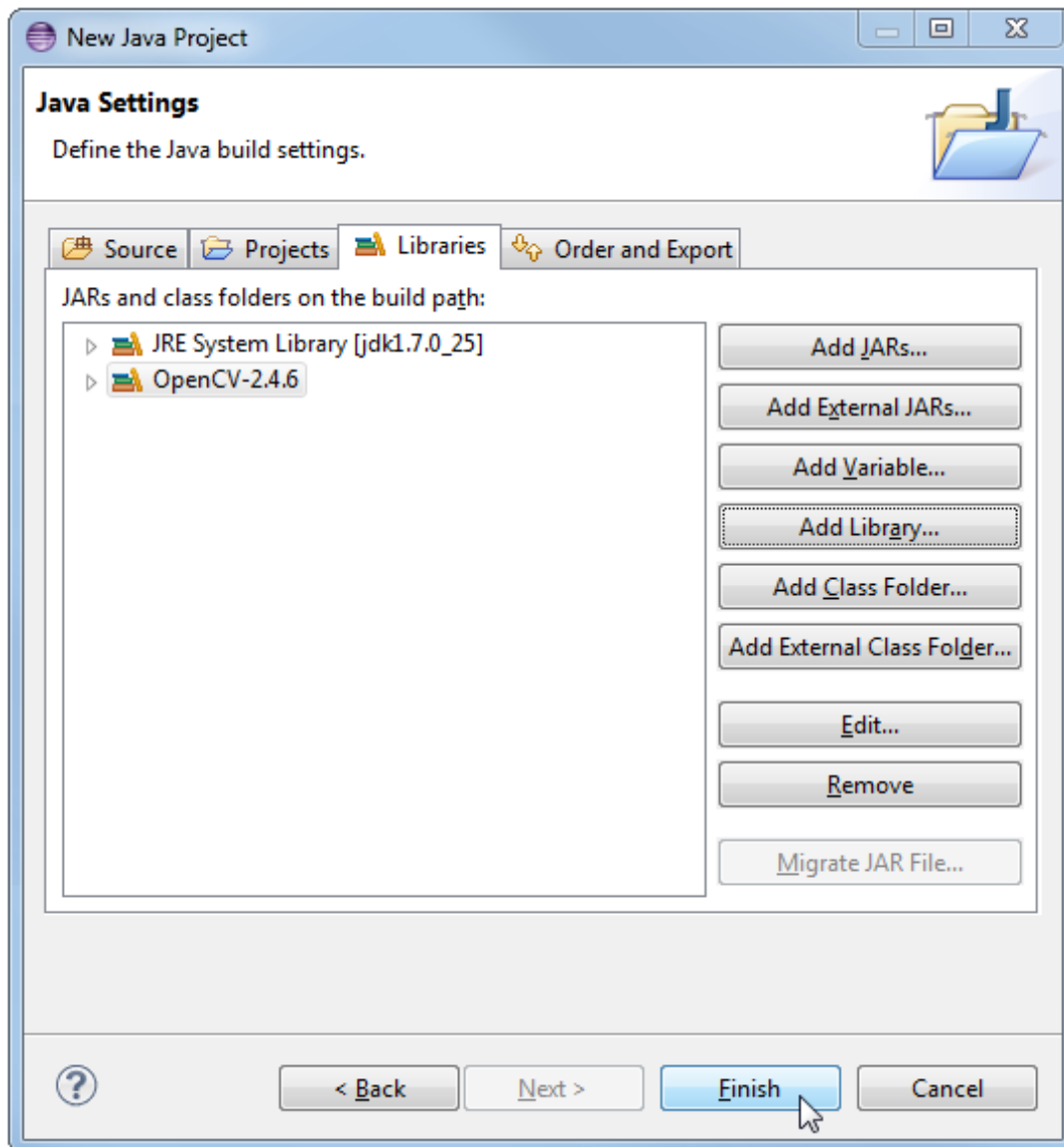


FIGURE 18

Import Function for image show

Create new java Class file, name as “FuncImage”, and copy code into it. This step is for Import the image show function, using JFrame (like Windows Form)

```
package demo;
import java.awt.FlowLayout;
import java.awt.image.BufferedImage;
import java.awt.image.DataBufferByte;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
import org.opencv.core.Mat;
public class FuncImage {
public static void bufferedImageShow(Mat mat) {
FuncImage.bufferedImageShow(mat, null);
}
public static void bufferedImageShow(Mat mat, String title) {
BufferedImage bufferedImage = FuncImage.mat2BufferedImage(mat);
ImageIcon icon = new ImageIcon(bufferedImage);
JFrame frame = new JFrame(title);
frame.setLayout(new FlowLayout());
frame.setSize(bufferedImage.getWidth(null) + 30,
bufferedImage.getHeight(null) + 50);
JLabel lbl = new JLabel();
lbl.setIcon(icon);
frame.add(lbl);
frame.setLocationRelativeTo(null);
frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
public static BufferedImage mat2BufferedImage(Mat mat) {
int type = BufferedImage.TYPE_BYTE_GRAY;
if (mat.channels() > 1)
type = BufferedImage.TYPE_3BYTE_BGR;
int bufferSize = mat.channels() * mat.cols() * mat.rows();
byte[] b = new byte[bufferSize];
mat.get(0, 0, b);
BufferedImage bufferedImage = new BufferedImage(mat.cols(), mat.rows(), type);
final byte[] targetPixels = ((DataBufferByte)bufferedImage.getRaster()
.getDataBuffer()).getData();
System.arraycopy(b, 0, targetPixels, 0, b.length);
return bufferedImage;
}
}
```

Capture Image from Camera

Create new java Class file, name as "Demo_04_camera", and copy code into it.

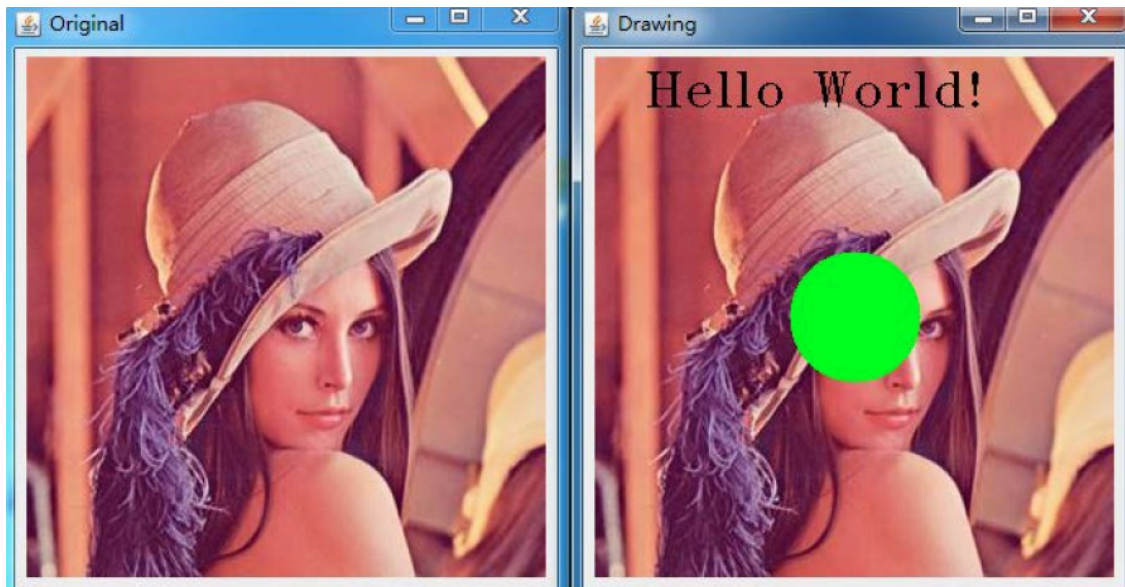
```
package demo;
import org.opencv.core.Core;
import org.opencv.core.Mat;
import org.opencv.core.MatOfRect;
import org.opencv.core.Point;
import org.opencv.core.Rect;
import org.opencv.core.Scalar;
import org.opencv.highgui.VideoCapture;
import org.opencv.objdetect.CascadeClassifier;
public class Demo_04_camera {
public static void main(String[] args) {
System.loadLibrary(Core.NATIVE_LIBRARY_NAME);
VideoCapture cap = new VideoCapture(0);
if (!cap.isOpened())
System.out.println("not Open");
else
System.out.println("fond webcam:" + cap.toString());
Mat matFrame = new Mat();
cap.retrieve(matFrame); // First frame maybe empty frame
cap.retrieve(matFrame);
FuncImage.bufferedImageShow(matFrame, "From Camera");
cap.release();
}
}.
```

11

Draw Shape

Create new java Class file, name as "Demo_06_drawing", and copy code into it.

```
package demo;
import org.opencv.core.Core;
import org.opencv.core.Mat;
import org.opencv.core.Point;
import org.opencv.core.Scalar;
import org.opencv.highgui.Highgui;
public class Demo_06_drawing {
public static void main(String[] args) {
System.loadLibrary(Core.NATIVE_LIBRARY_NAME);
Mat mat = Highgui.imread("files/lena.jpg");
FuncImage.bufferedImageShow(mat, "Original");
Core.circle(mat, new Point(mat.width() * 0.5, mat.height() * 0.5), 40,
new Scalar(0, 255, 0), Core.FILLED);
Core.putText(mat, "Hello World!", new Point(30, 30), 100, 1, new Scalar(
0, 0, 0));
FuncImage.bufferedImageShow(mat, "Drawing");
}
}
```



1.1 Problem Statement:

Problem:

In current scenario students have forgotten to play outdoor games and focus more on PC games. Our aim is to play a PC game using real objects like dice, alphabets etc. to design a multi user game. Computer use by both mouse and keyboard is possible and nowadays people are moving away from hardware games.

Solution:

To design a game which recognizes alphabets and dice count to automatically run the game and change the chance of the next user so that multiple users can play at the same time. This will enable users to play game using augmented reality which will make it a more reality based solution for gaming and applications.

CHAPTER-2

2.1 APPLICATIONS OF AUGMENTED REALITY:

Augmented reality other than being used for this project has also many other applications.

2.1

Medical

Doctors could use Augmented Reality as a visualization and training aid for surgery. It may be possible to collect 3-D datasets of a patient in real time, using non-invasive sensors like Magnetic Resonance Imaging (MRI), Computed Tomography scans (CT), or ultrasound imaging. These datasets could then be rendered and combined in real time with a view of the real patient. In effect, this would give a doctor "X-ray vision" inside a patient. This would be very useful during minimally-invasive surgery, which reduces the trauma of an operation by using small incisions or no incisions at all. A problem with minimally-invasive techniques is that they reduce the doctor's ability to see inside the patient, making surgery more difficult. AR technology could provide an internal view without the need for larger incisions.

AR might also be helpful for general medical visualization tasks in the surgical room. Surgeons can detect some features with the naked eye that they cannot see in MRI or CT scans, and vice-versa. AR would give surgeons access to both types of data simultaneously. This might also guide precision tasks, such as displaying where to drill a hole into the skull for brain surgery or where to perform a needle biopsy of a tiny tumor. The information from the non-invasive sensors would be directly displayed on the patient, showing exactly where to perform the operation.

AR might also be useful for training purposes. Virtual instructions could remind a novice surgeon of the required steps, without the need to look away from a patient to consult a manual. Virtual objects could also identify organs and specify locations to avoid disturbing.

Several projects are exploring this application area. At UNC Chapel Hill, a research group has conducted trial runs of scanning the womb of a pregnant woman with an ultrasound sensor, generating a 3-D representation of the fetus inside the womb and displaying that in a see-through HMD (Figure 5). The goal is to endow the doctor with the ability to see the moving, kicking fetus lying inside the womb, with the hope that this one day may become a "3-D stethoscope".

. More recent efforts have focused on a needle biopsy of a breast tumor.

Figure 3 shows a mockup of a breast biopsy operation, where the virtual objects identify the location of the tumor and guide the needle to its target. Other groups at the MIT AI Lab, General Electric and elsewhere are investigating displaying MRI or CT data, directly registered onto the patient.



FIGURE 5:Virtual foetus inside womb of pregnant woman.

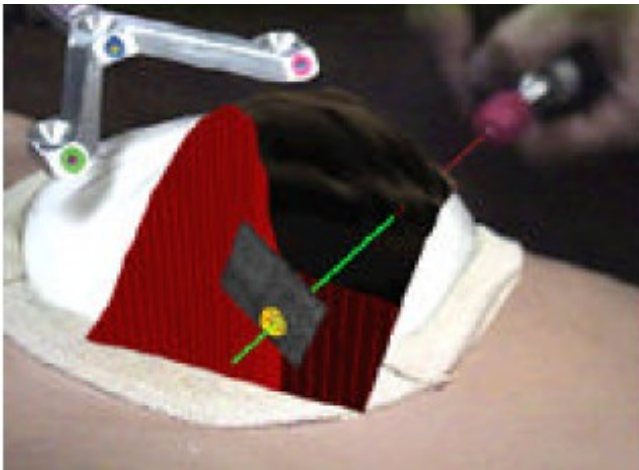


FIGURE 6:Mock up of breast tumour biopsy.

2.2 Manufacturing and repair

Another category of Augmented Reality applications is the assembly, maintenance, and repair of complex machinery. Instructions might be easier to understand if they were available, not as manuals with text and pictures, but rather as 3-D drawings superimposed upon the actual equipment, showing step-by-step the tasks that need to be done and how to do them. These superimposed 3-D drawings can be animated, making the directions even more explicit.

Several research projects have demonstrated prototypes in this area. Steve Feiner's group at Columbia built a laser printer maintenance application , shown in Figures 7. Figure 7 shows an external view.

. A group at Boeing is developing AR technology to guide a technician in building a wiring harness that forms part of an airplane's electrical system. Storing these instructions in electronic form will save space and reduce costs. Currently,

technicians use large physical layout boards to construct such harnesses, and Boeing requires several warehouses to store all these boards. Such space might be emptied for other use if this application proves successful .

Boeing is using a Technology Reinvestment Program (TRP) grant to investigate putting this technology onto the factory floor. Figure 7 shows an external view of Adam Janin using a prototype AR system to build a wire bundle. Eventually, AR might be used for any complicated machinery, such as automobile engines .



FIGURE 7: External view of Columbia Printer.



FIGURE 8: Boeing prototype wire assembly.

2.3 Annotation and visualization

AR could be used to annotate objects and environments with public or private information. Applications using public information assume the availability of public databases to draw upon. For example, a hand-held display could provide information about the contents of library shelves as the user walks around the library

At the European Computer-Industry Research Centre (ECRC), a user can point at parts of an engine model and the AR system displays the name of the part that is being pointed shows this, where the user points at the exhaust manifold on an engine model and the label "exhaust manifold" appears.

Alternately, these annotations might be private notes attached to specific objects. Researchers at Columbia demonstrated this with the notion of attaching windows from a standard user interface onto specific locations in the world, or attached to specific objects as reminders. He wears a tracking device, so the computer knows his location. As the student moves around, the label follows his location, providing the AR user with a reminder of what he needs to talk to the student about.

AR might aid general visualization tasks as well. An architect with a see-through HMD might be able to look out a window and see how a proposed new skyscraper would change her view. If a database containing information about a building's structure was available, AR might give architects "X-ray vision" inside a building, showing where the pipes, electric lines, and structural supports are inside the walls. Researchers at the University of Toronto have built a system called Augmented Reality through Graphic Overlays on Stereovideo (ARGOS) which among other things is used to make images easier to understand during difficult viewing conditions. The lines make it easier to see the geometry of the shuttle bay. Similarly, virtual lines and objects could aid navigation and scene understanding during poor visibility conditions, such as underwater or in fog.

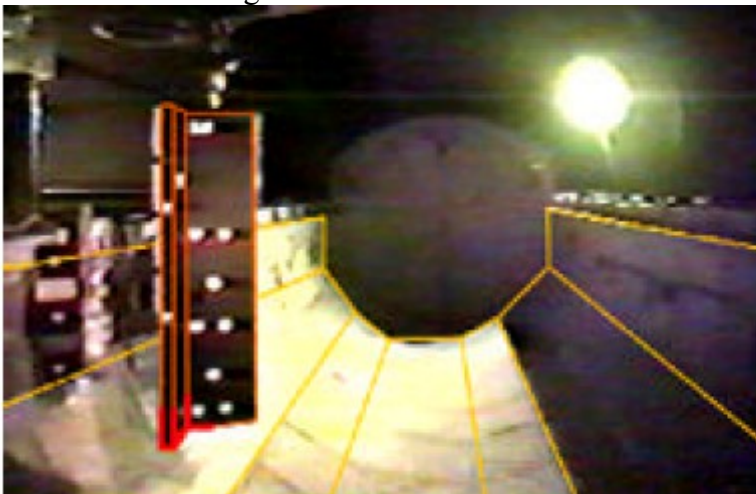


FIGURE 9: Virtual lines display geometry of bay as seen in orbit.

monoscopic interfaces Others have also used registered overlays with telepresence system.

Face Detect

Create new java Class file, name as "Demo_faceDetect", and copy code into it.

```
package demo;
import javax.swing.JOptionPane;
import org.opencv.core.Core;
import org.opencv.core.Mat;
import org.opencv.core.MatOfRect;
```

```

import org.opencv.core.Point;
import org.opencv.core.Rect;
import org.opencv.core.Scalar;
import org.opencv.highgui.Highgui;
import org.opencv.objdetect.CascadeClassifier;
public class Demo_faceDetect {
public static void main(String[] args) {
System.loadLibrary(Core.NATIVE_LIBRARY_NAME);
CascadeClassifier faceDetector = new CascadeClassifier(
"files/lbpcascade_frontalface.xml");
Mat mat = Highgui.imread("files/test.jpg");
MatOfRect faceDetections = new MatOfRect();
faceDetector.detectMultiScale(mat, faceDetection)

for (Rect rect : faceDetections.toArray()) {
Core.rectangle(mat, new Point(rect.x, rect.y), new Point(rect.x
+ rect.width, rect.y + rect.height), new Scalar(0, 255, 0));
}
FuncImage.bufferedImageShow(mat, "faceDetection");
JOptionPane.showMessageDialog(null,
"Detected " + faceDetections.toArray().length + " faces");
}
}

```

The XML File “lbpcascade_frontalface.xml”, you can find it from OpenCV Folder.
C:\OpenCV-2.4.6\sources\samples\android\face-detection\res\raw
Then Copy it to files folder in your Project.

20

2.2 REVIEW STUDY:OSMO

Osmo is best described as an iPad accessory that brings real-world gameplay to the tablet. The device itself consists of a simple white iPad stand that props up the tablet in portrait mode, and a small red clip that covers up the iPad’s camera. Once in place, an internal mirror redirects the camera viewpoint to an area on the table in front of the iPad that roughly is the size of an open magazine. This area of Osmo is used for gameplay. At launch, Osmo comes with two packaged games with extra accessories. One of them is Tangram, the traditional Chinese puzzle game consisting of seven objects that can be combined to create

The goal of this Osmo game is to redirect balls that are falling on the iPad screen to hit other objects on the screen. Except you don’t do the redirecting on the screen, but on your piece of paper. Draw a line, and the iPad app automatically imports it into the game. Draw another one, and you might have a funnel for those falling balls. Now add a physical object — a key, a pen, a toy, your own fingers, or any object, really — and Osmo automatically registers its outlines and adds them to the game as well. It’s augmented reality meets scribbling, and it’s a whole lot of fun.

GAMES ON OSMO:

TANGRAM:

Arrange tangible puzzle pieces into matching on-screen shapes. Play with a friend or challenge yourself to more advanced levels as your handy-work lights up with each victory



FIGURE 11

FUTURE WORK

We are currently evaluating our technique in the overall framework of a transpacific teleconferencing project, including tests of user acceptance for different methods of reinstating homo graphies. We are working towards further improving the tracking robustness, considering frame-to-frame constraints on feature patterns in the scene. We also aim to bring up the frame rate by both increased computational efficiency and use of hardware acceleration, e.g. a GPU implementation of the warping.

REFERENCES

- [1] T. Arici, S. Dikbas, and Y. Altunbasak, “A histogram modification framework and its application for image contrast enhancement,” *IEEE Trans. Image Process.*, vol. 18, no. 9, pp. 1921-1935, Sep. 2009.
- [2] K.Sreedhar and B.Panlal , Enhancement of images using morphological transformation|| *IJCS&IT*, vol. 4, no. 1,2012.
- [3] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, New Jersey, 2008.
- [4] I.R. Terol-Villalobos, —Morphological image enhancement and segmentation with analysis,|| P. W. Hawkes, Ed.New York: Academic, 2005, pp. 207–273.
- [5] Muhammad shahzed and shiraz latif, —efficient image enhancement Techniques,|| *Jl&CT*, vol. 3, pp. no.1, 2009.
- [6] Scott E. Umbauugh , *Computer Vision and Image Processing*, PH , New Jersey 1998, pp209.
- [7] Rafael C. Gonzalez, and Richard E. Woods, “*Digital Image Processing*”, 2nd edition, Prentice Hall, 2002
- [8] R. Mantiuk, S. Daly and L. Kerofsky, “Display adaptive tone mapping”,*ACM Trans. Graphics*, vol.27, no. 3, pp. 681-690, 2008.

