P-3113

THE APPROACH OF REAL TIME MONITORING BASED ON BACKGROUND SUBTRACTION

A PROJECT REPORT

Submitted by

SHANKAR.P	71206104044
SURESH.M	71206104050

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY ,COIMBATORE

ANNA UNIVERSITY:: CHENNAI 600 025

APRIL 2010

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"THE APPROACH OF REAL TIME MONITORING BASED ON BACKGROUND SUBTRACTION"** is the bonafide work of "**SHANKAR.P** and **SURESH.M**" who carried out the project work under my supervision.

[illegible]

SIGNATURE

Prof. S.THANGASAMY, Ph.D

HEAD OF THE DEPARTMENT

Computer Science and Engineering

Kumaraguru College of Technology,

Coimbatore-641 006.

[illegible]

SIGNATURE

Ms.S.RAJINI,M.S

SUPERVISOR

SENIOR LECTURER,CSE Department

Kumaraguru College of Technology,

Coimbatore-641 006.

Submitted for the University Examination held on 16-4-10

[illegible]

Internal Examiner

[illegible]

External Examiner

DECLARATION

We

SHANKAR.P	**71206104044**
SURESH.M	**71206104050**

Here by declare that the project entitled " **THE APPROACH OF REAL TIME MONITORING BASED ON BACKGROUND SUBTRACTION**" is a record of original work done by us and to the best of our knowledge, a similar work has not been submitted to Anna University or any Institution, for fulfillment of the requirement of the course study.

The report is submitted in partial fulfillment for the award of the degree of Bachelor of Computer Science and Engineering of Anna University ,Chennai.

Place:Coimbatore
Date: 16.4.10.

[illegible]
[SHANKAR.P]

[illegible]
[SURESH.M]

ACKNOWLEDGEMENT

We extend our sincere thanks to our Principal **Dr.S.RAMACHANDRAN,Ph.D.**,for being a constant source of inspiration and providing us with necessary facility to work on this project.

We would like to make a special acknowledgement and thanks to **Dr.S.THANGASAMY,Ph.D.**,Dean,Professor and Head of Department of Computer Science and Engineering, for his support and encouragement throughout the project.

We express deep gratitude to our guide, **Ms.S.RAJINI,**M.S, Department of Computer Science and Engineering for her supervision, enduring patience, active involvement and guidance.

We would also like to thank our Project Coordinator, **Ms.P.DEVAKI,**M.S Associate Professor, Department of Computer Science and Engineering, for her support during the course of our project.

We would like to convey our honest thanks to all members of staff of the Department .

ABSTRACT

Real-time monitoring is an important research content of safety domain and real-time control, and the study on its related key technology has great significance. In this project, an algorithm of real-time monitoring is proposed, which is based on background subtraction. By using the frame image obtained from real-time monitor or video, the images are divided into moving object area and still area, which is used for updating the background. Then the moving area is compared with background to capture the moving object and its position, which is used to recognize the security of the monitored real-time system. Experimental results demonstrate that the algorithm can quickly and exactly detect the moving objects and give the alarm.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The development of monitoring is a major trend in future, it is an important research content of security field, Recently, monitoring system has became commonplace. So, study on its related technology has great significance.

Extracting the moving objects, in particular, are interesting and important in monitoring systems, because it can contribute not only to theoretical insights but also to practical application. Computing the moving objects of the monitoring system could be applied to a wide variety of problems, including criminal identification, real time control system, traffic control, industry and civil monitoring, etc..

Unfortunately, extracting the moving objects quickly and exactly from a real time stream image or monitoring video is quite difficult, because of the following reasons:

1.1 Complexity of objects:

All kinds of moving objects can be detected by the object detection process, and they have different sizes and features ,which is quite difficult to be processed later.

1.2 Complexity of the environment:

The monitoring systems are all ways inevitably exposed in a real time and variety of environment, including the dithering of the background, or in the brightness of the environment, or in the chromatism between the foreground color and background color.

1.3 Complexity of the device:

All of the devices including input devices and processing devices make the image signal susceptible to noise, which makes the later processing complex.

Therefore, moving objects detection is the key technology of the monitoring. Because of the simple principle and the simple implementation, the image difference method is widely used for detecting the moving objects in the monitoring system. The algorithms of moving objects can be divided into the following categories:

1.4 Background subtraction:

Background subtraction is a commonly used class of techniques for segmenting out objects; it involves comparing an observed image with an estimate of the image if it contained no objects of interest.

1.5 Frame difference:

Frame difference is to look at the difference between two consecutive frames to detect a cut, and a large number of difference metrics has been defined to estimate frame differences. The straightforward approach is to measure the differences between the particular pixels consecutive frames, but this approach is very sensitive to object motion, camera motion, brightness changes and noise.

CHAPTER 2
LITERATURE REVIEW

2.1 EXISTING SYSTEM

- In the existing design they used Unimodel distribution.
- In this method, they compared individual pixel intensity values for subtraction so it has low complexity but will not update the background.
- They used Mobile phones to capture the Video.
- But while using Mobile, we need an user near the mobile to initiate sending of each and every SMS.
- This cannot handle moving backgrounds

2.2 PROPOSED SYSTEM

- In the proposed design we are using Background Subtraction for segmenting of the objects.
- Next we will find out the frame difference ,then SMS alerts are sent to the User mobile.
- We can reduce the complexity of the moving objects , environment and device.
- It has good performance and efficiency.

CHAPTER 3

REQUIREMENTS SPECIFICATION

3.1 HARDWARE REQUIREMENTS

Processor	:	PentiumIII
Hard Disk	:	20GB
RAM	:	128MB
Monitor	:	VGA Color

3.2 SOFTWARE REQUIREMENTS

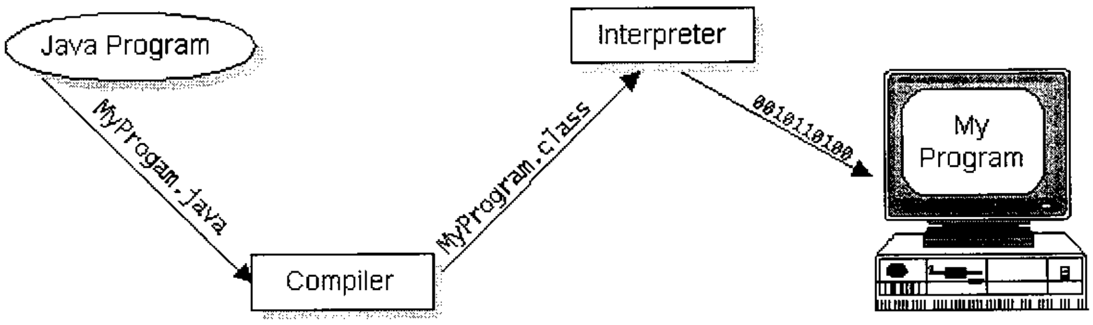
Language	:	Java, Swings
Build Tool	:	Apache Ant 1.7.0

3.3 SOFTWARE DESCRIPTION

3.3.1 JAVA

It is a platform independent programming language that extends its features wide over the network. Java2 version introduces a new component called "Swing" – is a set of classes that provides more power & flexible components than are possible with AWT. It is a light weight package, as they are not implemented by platform-specific code. Related classes are contained in javax.swing and its sub packages, such as javax.swing.tree. Components explained in the Swing have more capabilities than those of AWT.

Java is a high-level programming language. Java is also unusual in the case that each Java program is both compiled and interpreted. With a compiler, you translate a Java program into an intermediate language called Java byte codes, the platform-independent codes interpreted by the Java interpreter. With an interpreter, each Java byte code instruction is parsed and run on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.



Java byte codes can be considered as a machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it is a Java development tool or a Web browser that can run Java applets, is an implementation of the Java VM. The Java VM can also be implemented in hardware.

Java byte codes help in making "write once, run anywhere" possible. The Java program can be compiled into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. For example, the same Java program can run on Windows NT, Solaris, and Macintosh.

SWING:

The Swing toolkit includes a rich set of components for building GUIs and adding interactivity to Java applications. Swing includes all the components you expected from a modern toolkit: table controls, list controls, tree controls, buttons, and labels. Swing is far from a simple component toolkit, however. It includes rich undo support, a highly customizable text package, integrated internationalization and accessibility support.

To truly leverage the cross-platform capabilities of the Java platform, Swing supports numerous look and feel, including the ability to create one's own look and feel.. Swing wouldn't be a component toolkit without the basic user interface primitives such as drag and drop, event handling, customizable painting and window management.

Swing is part of the Java Foundation Classes (JFC). The JFC also include other features important to a GUI program, such as the ability to add rich graphics functionality and the ability to create a program that can work in different languages with different input devices. The Swing toolkit includes a rich array of components: from basic components, such as buttons and check boxes, to rich and complex components, such as tables and text.

Even deceptively simple components, such as text fields, offer sophisticated functionality, such as formatted text input or password field behavior. There are file browsers and dialogs to suit most needs, and if not, customization is possible.

3.3.2. APACHE TOMCAT 6.0

Apache Tomcat is a web container, or application server developed at the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the Java Server Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management but can also be configured by editing configuration files that are normally XML-formatted.

CHAPTER 4

PROJECT DESCRIPTION

4.1. PROBLEM DEFINITION

The aim of the project is to Extracting the moving objects quickly and exactly from a real time stream image or monitoring video.

4.2 PROBLEM SOLUTION

The system is divided into 3 modules.

- User Interface
- Gray-Scale Conversion
- BackGround Subtraction

4.2.1 USER INTERFACE

- The user interface is done using Swing.
- Swing is called Light Weight Process .You can develop your own look n feel using swings.
- Swing is a toolkit in java. It is part of Sun Microsoft System JFC i.e Java Foundation Classes an API for providing a GUI for Java programs. Swing includes GUI widgets such as text boxes, buttons, split-panes, and tables.
- Swing supports a pluggable look and feel.This means you can get any supported look and feel on any platform or by using the current platform's graphics interface to achieve consistency through modifications made by additional API calls. This means the application can use any supported look and feel on any platform.

4.2.2 GRAY-SCALE CONVERSION

- In this module we will take a 3 plane image i.e RGB image of 2 different frames.
- This RGB image is converted into the Gray-Scale image by using color theory.
- For human eye color will not be visible , only luminance (brightness) will be visible.
- So from that RGB image, we have to take only the intensity variation. We are going to separate the intensity values(gray values) from the color images.

4.2.3 BACKGROUND SUBTRACTION

- To find the moving object we have to compare the foreground and background frame.
- To compare the 2 frames difference we have to apply BackGround Subtraction algorithm.
- Without object we have to take one frame as a background.
- Then we have to compare the fore ground frame with the background frame.
- The compared result will be checked with the Threshold.
- If the difference is greater than the threshold that point is noted as a moving object.

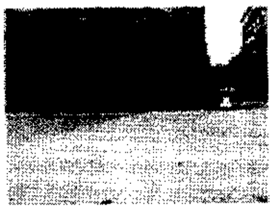


Figure. 2.1 Background image

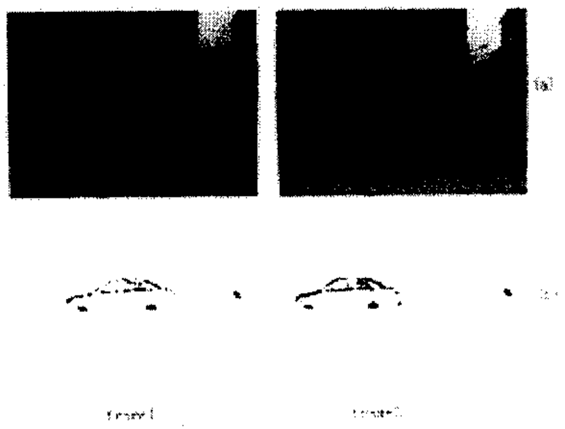


Figure. 2.2 moving object

CHAPTER 5
TESTING AND IMPLEMENTATION

5.1. SOFTWARE TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

5.2. TEST OBJECTIVES

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

5.3. FEATURES TO BE TESTED

- Verify that the entries are in correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

5.4. TYPES OF TESTING

5.4.1. UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produces valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.4.2. INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

5.4.3. FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation , and user manuals.

Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.
Systems/Procedures	: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

5.4.4. SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

5.5 IMPLEMENTATION OF THE ALGORITHM:

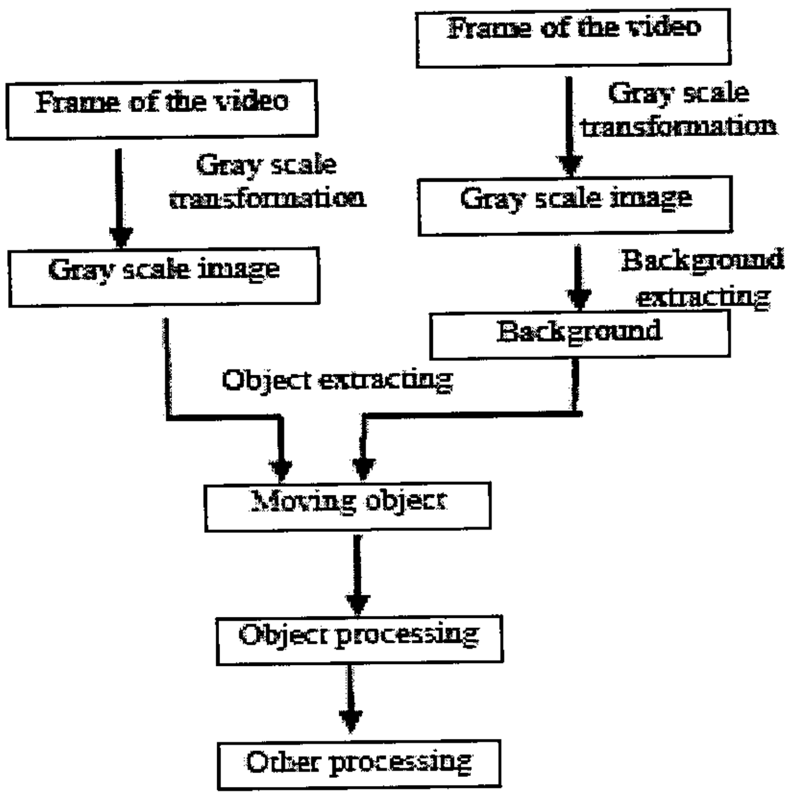


Figure. 5.1 Implementation

The algorithm of the monitoring based on background subtraction can be implemented by the following process:

- First, the images are extracted from a monitor or a video.
- Secondly, the color image must be transformed into gray scale image.
- According to the change of the environment and the algorithm of the background introduced above,the real background image is extracted in the third process, which is ready for the next process.

• According to the approach of the moving object extracting proposed above, the moving objects are extracted from the original image in the fourth process.

• The fifth process is the object processing, which has been obtained from the fourth process, and which is the real moving objects of the system.

• The last process is the other processing of the system. Different system has different function in this process.

5.5.1 EXPERIMENT:

In order to validate the performance of the algorithm of the real time monitoring based on background subtraction, which is discussed above, extensive experiments are conduced on many vehicle images, which are obtained from a real time monitor.

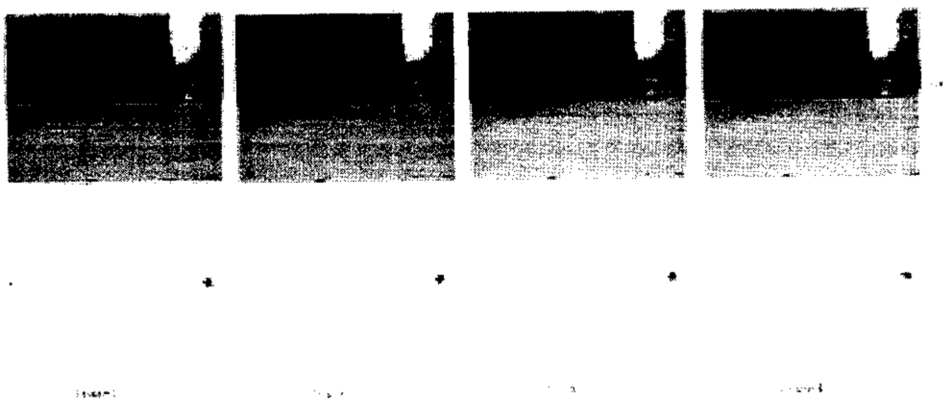


Figure. 5.2 Experiment

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 CONCLUSION:

A technology of background subtraction for real time monitoring system was proposed in this project. The obvious keystone of our work is studying the principle of the background subtraction, discussing the problem of the background, and exploring the base resolve method of the problem. Experiments show that the method has good performance and efficiency.

6.2 FUTURE WORK

The Possibilities of future work exist in the following areas:

- A low-cost intelligent mobile phone-based wireless video surveillance solution using moving object recognition technology developed.
- The subtraction of the current captured image and the background reaches a certain threshold ,a moving object can be considered to be in the current view ,and the mobile phone would automatically notify the central control unit or the user through phone call ,SMS.

APPENDIX 1

CODING

SERVER MANAGER:

```
import  java.io.*;
import  java.util.*;
import  java.net.*;
import  java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;
import java.rmi.*;
import java.rmi.server.*;
import java.sql.*;
import java.util.Properties;
import java.awt.Image;
import java.awt.image.BufferedImage;
import java.awt.*;
import com.gif4j.ImageUtils;
import com.gif4j.GifEncoder;
import com.gif4j.GifDecoder;
import com.gif4j.GifImage;
import com.gif4j.GifTransformer;

public class servermanager extends UnicastRemoteObject implements remote
 {
    static int c=0;
     Connection con,con1;
     Statement st,st1;
      int i,i1;
  ResultSet r1;
  DataBase db=null;
          public servermanager() throws RemoteException
          {
       try{
         System.out.println("server");
                    Naming.rebind("rob",this);
                      } catch(Exception ee){}
          }
                public String writeImageFile(byte[]img,String filen)throws RemoteException
             {
```

```
          String reply="";
          if(c==0)
             {
               close();

}


          try{
     String filepath="Images From Client/"+filen;
                    String imagepath="Server/Images From Client/"+filen;
     FileOutputStream fos=new FileOutputStream(filepath);
                    fos.write(img);
                    fos.flush();
                    fos.close();
                    System.out.println(" File created sucess ");

                    SmsSend sms=new SmsSend();

                    db=new DataBase();
     i=db.executeUpdate("insert into images(filename)values('"+imagepath+"')");
                    System.out.println("i");

     Image image = Toolkit.getDefaultToolkit().createImage(img);
     System.out.println("i1");
        BufferedImage bufferedImage = ImageUtils.toBufferedImage(image);
        System.out.println("i2");
         File fil = new File(filepath);
         System.out.println("i3");
        GifEncoder.encode(bufferedImage, fil);
        System.out.println("i4");

       File gifImageFileToTransform = new File(filepath);
        System.out.println("i5");

        GifImage gifImage = GifDecoder.decode(gifImageFileToTransform);
        System.out.println("i6");

        GifImage resizeGifImage = GifTransformer.resize(gifImage, 160, 120, false);
        System.out.println("i7");

        GifEncoder.encode(resizeGifImage,new File(filepath));
                              //System.out.println("alet");
                   //a a1=new a();
                              //System.out.println("alet1");
          c++;

          }


            catch(Exception ee)
    {
    System.out.println(" Exception server file "+ee);
    }
                return reply;
     }
        public void close()
        {

         try{
        System.out.println("close");

        i1=db.executeUpdate("delete * from images");
       System.out.println("del");
                        }
                        catch(Exception e)
                        {
        System.out.println("error"+e.toString());
                        }

   }
        public static void main(String tt[])
         {
          try{
             servermanager s1=new servermanager();

             }
    catch(Exception ee){}
         }


}
```

LOGIN:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class Login extends JFrame implements ActionListener
{
 JButton SUBMIT,CANCEL;
 JPanel panel;
 JLabel label1,label2;

 final JTextField  text1,text2;
        Login()
        {
          label1 = new JLabel();
                label1.setText("Username:");
                text1 = new JTextField(15);

                label2 = new JLabel();
                label2.setText("Password:");
          text2 = new JPasswordField(15);
                 SUBMIT=new JButton("SUBMIT");
                CANCEL=new JButton("RESET");
          panel=new JPanel(new GridLayout(3,1));
                panel.add(label1);
                panel.add(text1);
                panel.add(label2);
                panel.add(text2);
                panel.add(SUBMIT);
                panel.add(CANCEL);
          add(panel,BorderLayout.CENTER);
         SUBMIT.addActionListener(this);
                CANCEL.addActionListener(this);
                setDefaultCloseOperation(EXIT_ON_CLOSE);
     setTitle("Remote Monitoring-Login Page");
        }
   public void actionPerformed(ActionEvent ae)
        {
                String value1=text1.getText();
                String value2=text2.getText();
```

```
              if(ae.getSource()==SUBMIT)
              {
   if (value1.equals("admin") && value2.equals("admin"))
                        {
                         this.setVisible(false);
               Webcam webcam=new Webcam();

               webcam.setDefaultCloseOperation(EXIT_ON_CLOSE);
    webcam.setSize(300,150);
    webcam.setTitle("Remote Monitoring");
    webcam.setResizable(false);
    webcam.setLocationRelativeTo(null);
    webcam.setVisible(true);
              }
              else{
               text1.setText("");
               text2.setText("");
              System.out.println("enter the valid username and password");
              JOptionPane.showMessageDialog(this,"Incorrect login
      password","Error",JOptionPane.ERROR_MESSAGE);
         }
        }
              else{
       text1.setText("");
   text2.setText("");
  }
}

public static void main(String arg[])
       {
              try
              {
     Login login=new Login();
               login.setSize(400,200);
               login.setLocationRelativeTo(null);
              login.setVisible(true);
              }
       catch(Exception e)
              {JOptionPane.showMessageDialog(null, e.getMessage());
          }

 }

}
```

WEBCAM:

```
import java.awt.*;
import javax.swing.*;
import java.io.Serializable;
import java.awt.event.*;
import java.net.*;
import java.io.*;
import java.util.*;
import java.rmi.server.*;
import java.net.*;
import java.rmi.*;

public class Webcam extends JFrame implements ActionListener
 {
  TestMotionDetection t1;
  long lastPlay = 0;
  int imagem=1;
  int imagec=1;
  int imc=1;
  int v;
  int tRegions=0;
  boolean boo;
  JPanel pan;
  JButton stcbut;
  JButton srtcbut;
  JButton qubut;
  JLabel head;
  Thread th;
   {

     Webcam al=new Webcam();
     al.setDefaultCloseOperation(EXIT_ON_CLOSE);
     al.setSize(300,150);
     al.setTitle("FAST MOTION DETECTION");
     al.setResizable(false);
     al.setLocationRelativeTo(null);
     al.setVisible(true);
   }

  public Webcam()
          {

     pan=new JPanel();
     pan.setLayout(null);
     srtcbut=new JButton("Start CAM");
   stcbut=new JButton("Stop CAM");
   qubut=new JButton("Quit");
   head=new JLabel("REMOTE MONITORING");
   getContentPane().add(pan);
   srtcbut.setBounds(40,40,100,30);
   stcbut.setBounds(170,40,100,30);
   qubut.setBounds(150,150,150,30);

   pan.add(stcbut);
         pan.add(srtcbut);
         pan.add(head);
   stcbut.addActionListener(this);
   srtcbut.addActionListener(this);
   qubut.addActionListener(this);
  }
{
   Vision.setImageSize(320,240);
   Vision.flipHorizontal(true);
   Vision.addRectRegion(1, 0, 0, 320,240);

   Vision.startViewer("Web Cam");
  th.start();
  }
 public void actionPerformed(ActionEvent e)
    {

       if(e.getSource()==stcbut)
                 {
                  System.exit(0);
                  System.out.println("stop");
       }

     if(e.getSource()==srtcbut)
                 {
                 System.out.println("start");
                 t1=new TestMotionDetection("vfw://0");
                 //run1();

      }

                 {
      System.exit(0);
     }
   }

 }
```

TIME STAMP EFFECT:

```
import javax.media.*;
import javax.media.format.*;
import java.awt.*;
import com.sun.image.codec.jpeg.JPEGCodec;
import com.sun.image.codec.jpeg.JPEGImageEncoder;
import java.awt.image.BufferedImage;
import java.io.*;
import java.util.*;
import java.rmi.server.*;
import java.net.*;
import java.rmi.*;


public class TimeStampEffect implements Effect {

   Format inputFormat;
   Format outputFormat;
   Format[] inputFormats;
   Format[] outputFormats;
   java.text.SimpleDateFormat sdf;
   static int d=0;

   public TimeStampEffect() {

      sdf = new java.text.SimpleDateFormat("hh:mm:ss MM/dd/yy");
      inputFormats = new Format[] {
         new RGBFormat(null,
                  Format.NOT_SPECIFIED,
                  Format.byteArray,
                  Format.NOT_SPECIFIED,
                  24,
                  3, 2, 1,
                  3, Format.NOT_SPECIFIED,
                  Format.TRUE,
                  Format.NOT_SPECIFIED)
      };

      outputFormats = new Format[] {
         new RGBFormat(null,
                  Format.NOT_SPECIFIED,
                  Format.byteArray,
                  Format.NOT_SPECIFIED,
                  24,
                  3, 2, 1,
                3, Format.NOT_SPECIFIED,
                Format.TRUE,
                Format.NOT_SPECIFIED)
    };

}

public Format[] getSupportedInputFormats() {
      return inputFormats;
}

public Format [] getSupportedOutputFormats(Format input) {
   if (input == null) {
      return outputFormats;

   }

   if (matches(input, inputFormats) != null) {

      return new Format[] { outputFormats[0].intersects(input) };

   } else {
      return new Format[0];
   }
}

public Format setInputFormat(Format input) {
      inputFormat = input;
      return input;
}

public Format setOutputFormat(Format output) {

   if (output == null || matches(output, outputFormats) == null)
      return null;
   RGBFormat incoming = (RGBFormat) output;

   Dimension size = incoming.getSize();
   int maxDataLength = incoming.getMaxDataLength();
   int lineStride = incoming.getLineStride();
   float frameRate = incoming.getFrameRate();
   int flipped = incoming.getFlipped();
   int endian = incoming.getEndian();

   if (size == null)
      return null;
```

```
  if (maxDataLength < size.width * size.height * 3)
    maxDataLength = size.width * size.height * 3;
  if (lineStride < size.width * 3)
    lineStride = size.width * 3;
  if (flipped != Format.FALSE)
    flipped = Format.FALSE;

  outputFormat = outputFormats[0].intersects(new RGBFormat(size,
                                    maxDataLength,
                                    null,
                                    frameRate,
                                    Format.NOT_SPECIFIED,
                                    Format.NOT_SPECIFIED,
                                    Format.NOT_SPECIFIED,
                                    Format.NOT_SPECIFIED,
                                    Format.NOT_SPECIFIED,
                                    lineStride,
                                    Format.NOT_SPECIFIED,
                                    Format.NOT_SPECIFIED));

  return outputFormat;
}


public int process(Buffer inBuffer, Buffer outBuffer) {

   int outputDataLength = ((VideoFormat)outputFormat).getMaxDataLength();
   validateByteArraySize(outBuffer, outputDataLength);

   outBuffer.setLength(outputDataLength);
   outBuffer.setFormat(outputFormat);
   outBuffer.setFlags(inBuffer.getFlags());

   byte [] inData = (byte[]) inBuffer.getData();
   byte [] outData = (byte[]) outBuffer.getData();

   RGBFormat vfIn = (RGBFormat) inBuffer.getFormat();
   Dimension sizeIn = vfIn.getSize();

   int pixStrideIn = vfIn.getPixelStride();
   int lineStrideIn = vfIn.getLineStride();

   if ( outData.length < sizeIn.width*sizeIn.height*3 ) {
      System.out.println("the buffer is not full");
      return BUFFER_PROCESSED_FAILED;
   }
```

```
      Calendar cal = new GregorianCalendar();
      String tt;
      int hour12 = cal.get(Calendar.HOUR);
      int min = cal.get(Calendar.MINUTE);
      int sec = cal.get(Calendar.SECOND);
      int ampm = cal.get(Calendar.AM_PM);
      if(ampm==0)
         tt="AM";
      else
         tt="PM";
      String
time=Integer.toString(hour12)+"."+Integer.toString(min)+"."+Integer.toString(sec)+"."+tt;

      Font.println(sdf.format(new java.util.Date()).toString() + " (math room 205)",
Font.FONT_6x11, 10, 20, (byte)255,(byte)255,(byte)255, outBuffer);
      System.arraycopy(inData,0,outData,0,inData.length);
      try{
      writeImage("MotDetImages/MD@"+time+".gif",inData,320,240);
       }
       catch(Exception es) { }


       try{
         remote remob;
                   Thread.sleep(3000);
                         FileInputStream fin=new
FileInputStream("MotDetImages/MD@"+time+".gif");
                         int size=fin.available();
                         byte img[]=new byte[size];
                         fin.read(img);
                         fin.close();
         remob=(remote)Naming.lookup("//127.0.0.1/rob");
                   remob.writeImageFile(img,"MD@"+time+".gif");
                   System.out.println("Image File send to server ");

                     }

        catch(Exception err){System.out.println("Exception remote "+err);

                   }

        System.out.println(" The Image created for comparing on region ");

 return BUFFER_PROCESSED_OK;
   }
 public String getName()
{
     return "TimeStamp Effect";
  }

  public void open() {
  }

  public void close() {
  }

  public void reset() {
  }
  public Object getControl(String controlType) {
        return null;
  }

  public Object[] getControls() {
        return null;
  }

  Format matches(Format in, Format outs[]) {
        for (int i = 0; i < outs.length; i++) {
           if (in.matches(outs[i]))
                 return outs[i];
        }

        return null;
  }

 byte[] validateByteArraySize(Buffer buffer,int newSize) {
      Object objectArray=buffer.getData();
      byte[] typedArray;

      if (objectArray instanceof byte[]) {     // is correct type AND not null
         typedArray=(byte[])objectArray;
         if (typedArray.length >= newSize ) { // is sufficient capacity
            return typedArray;
         }

         byte[] tempArray=new byte[newSize];  // re-alloc array
         System.arraycopy(typedArray,0,tempArray,0,typedArray.length);
         typedArray = tempArray;
      } else {
         typedArray = new byte[newSize];
      }
      buffer.setData(typedArray);
      return typedArray;
   }
  public void writeImage(String s, byte abyte0[], int i, int j)throws FileNotFoundException,
IOException
   {
      FileOutputStream fileoutputstream = new FileOutputStream(s);
      JPEGImageEncoder jpegimageencoder =
JPEGCodec.createJPEGEncoder(fileoutputstream);
      int ai[] = new int[abyte0.length / 3];
      int k = 0;
      for(int l = j - 1; l > 0; l--)
      {
         for(int i1 = 0; i1 < i; i1++)
            ai[k++] = 0xff000000 | (abyte0[l * i * 3 + i1 * 3 + 2] & 0xff) << 16 | (abyte0[l * i * 3 +
i1 * 3 + 1] & 0xff) << 8 | abyte0[l * i * 3 + i1 * 3] & 0xff;

      }

      BufferedImage bufferedimage = new BufferedImage(i, j, 1);
      bufferedimage.setRGB(0, 0, i, j, ai, 0, i);
      jpegimageencoder.encode(bufferedimage);
      fileoutputstream.close();
   }
}
```

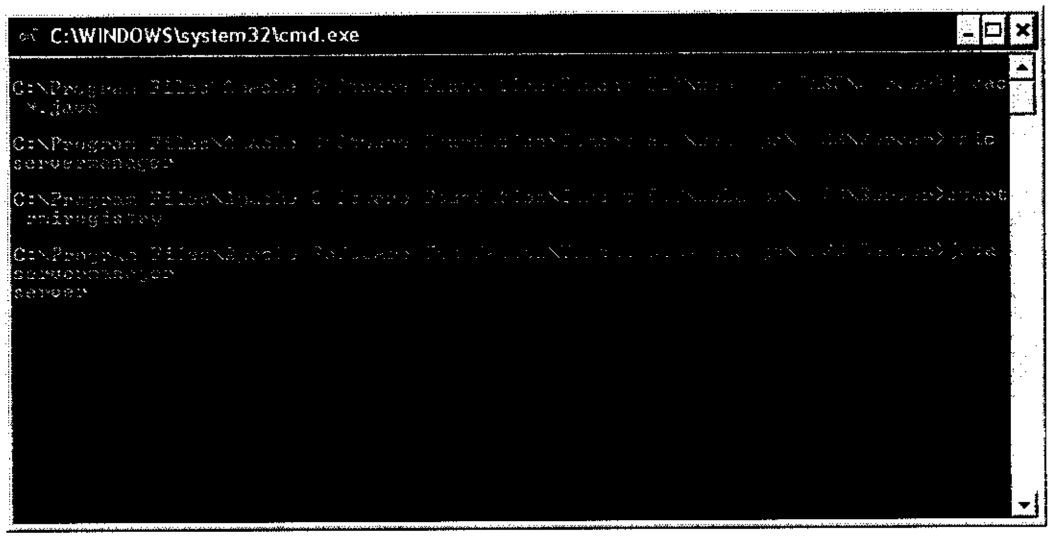
SMS SEND:

```
import java.io.*;
import java.util.*;
import javax.comm.*;
public class SmsSend {
   static Enumeration  portList;
   static CommPortIdentifier portId;
   static String messageString = "5";
   static SerialPort serialPort;
   static OutputStream outputStream;
   static booleanoutputBufferEmptyFlag = false;

   public SmsSend()
   {
        boolean portFound = false;
        String  defaultPort = "COM1";
        String driverName = "com.sun.comm.Win32Driver";
  try{
  CommDriver commdriver = (CommDriver)Class.forName(driverName).newInstance( );
   commdriver.initialize();
  }
  catch (Exception e2)
  {
  e2.printStackTrace();
   }
 portList = CommPortIdentifier.getPortIdentifiers();
 while (portList.hasMoreElements()) {
            portId = (CommPortIdentifier) portList.nextElement();

            if (portId.getPortType() == CommPortIdentifier.PORT_SERIAL) {

                  if (portId.getName().equals(defaultPort)) {
                     System.out.println("Found port " + defaultPort);

                     portFound = true;

                     try {
                           serialPort =
                              (SerialPort) portId.open("SimpleWrite", 2000);
                     } catch (PortInUseException e) {
                           System.out.println("Port in use.");

                           continue;
                     }
```

```
                try {
                      outputStream = serialPort.getOutputStream();
                } catch (IOException e) {}

                try {
                      serialPort.setSerialPortParams(9600,
                                                    SerialPort.DATABITS_8,
                                                    SerialPort.STOPBITS_1,
                                                    SerialPort.PARITY_NONE);
                } catch (UnsupportedCommOperationException e) {}


                try {
                      serialPort.notifyOnOutputEmpty(true);
                } catch (Exception e) {
                      System.out.println("Error setting event notification");
                      System.out.println(e.toString());
                      System.exit(-1);
                }


                System.out.println("Writing \""+messageString+"\" to "
                      +serialPort.getName());

                try {
                final char controlZ = 26;
                      outputStream.write(("AT+CMGS=9944339008\rIntrudor
Found"+controlZ).getBytes());
                } catch (IOException e) {}
        serialPort.close();

              }
          }
        }
if (!portFound) {
            System.out.println("port " + defaultPort + " not found.");
        }
   }
  public static void main(String arg[])
  {
     SmsSend sms=new SmsSend();

}

}
```

APPENDIX 2
SNAPSHOTS

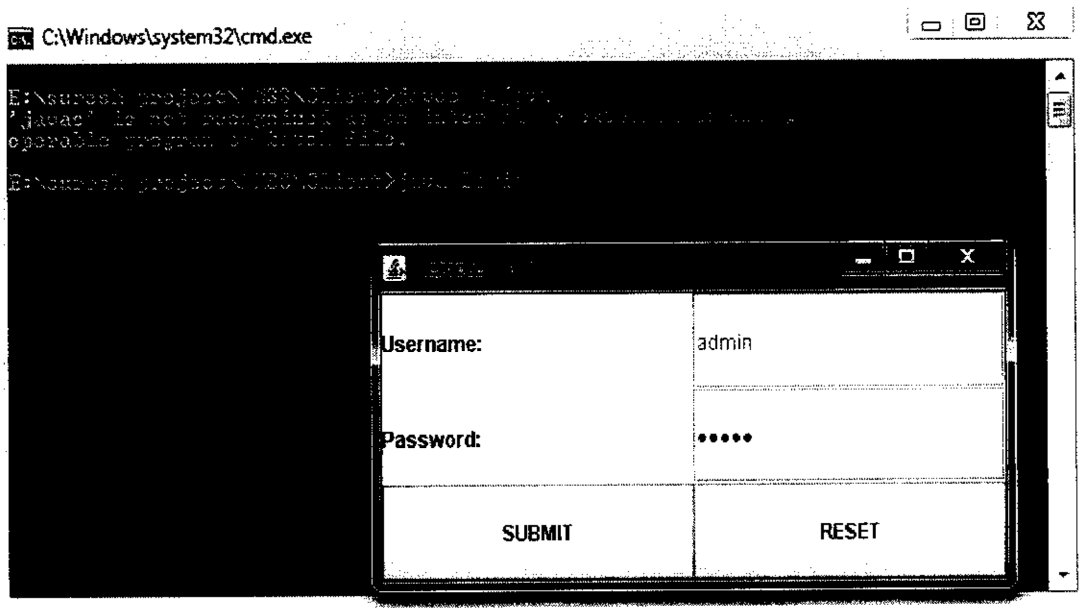
SERVER MANAGER:



C:\WINDOWS\system32\cmd.exe

```
C:\Program Files\[illegible]>javac
 *.java

C:\Program Files\[illegible]>rmic
servermanager

C:\Program Files\[illegible]>start
 rmiregistry

C:\Program Files\[illegible]>java
servermanager
server
```

LOGIN:



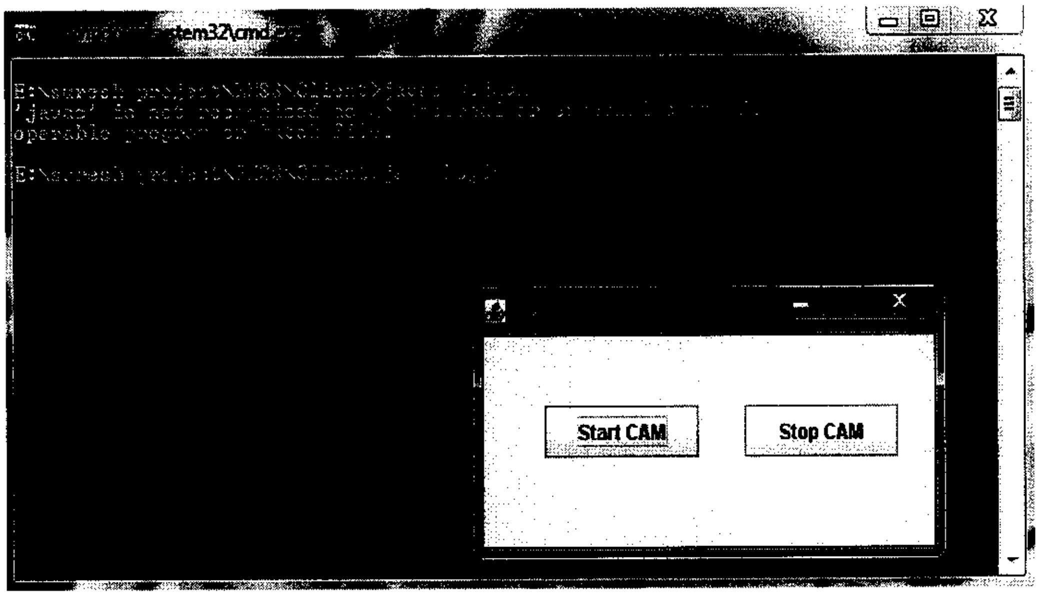
C:\Windows\system32\cmd.exe

```
E:\suresh project\[illegible]>javac *.java
'javac' is not recognized as an internal [illegible]
operable program or batch file.

E:\suresh project\[illegible]>java [illegible]
```

Username: admin

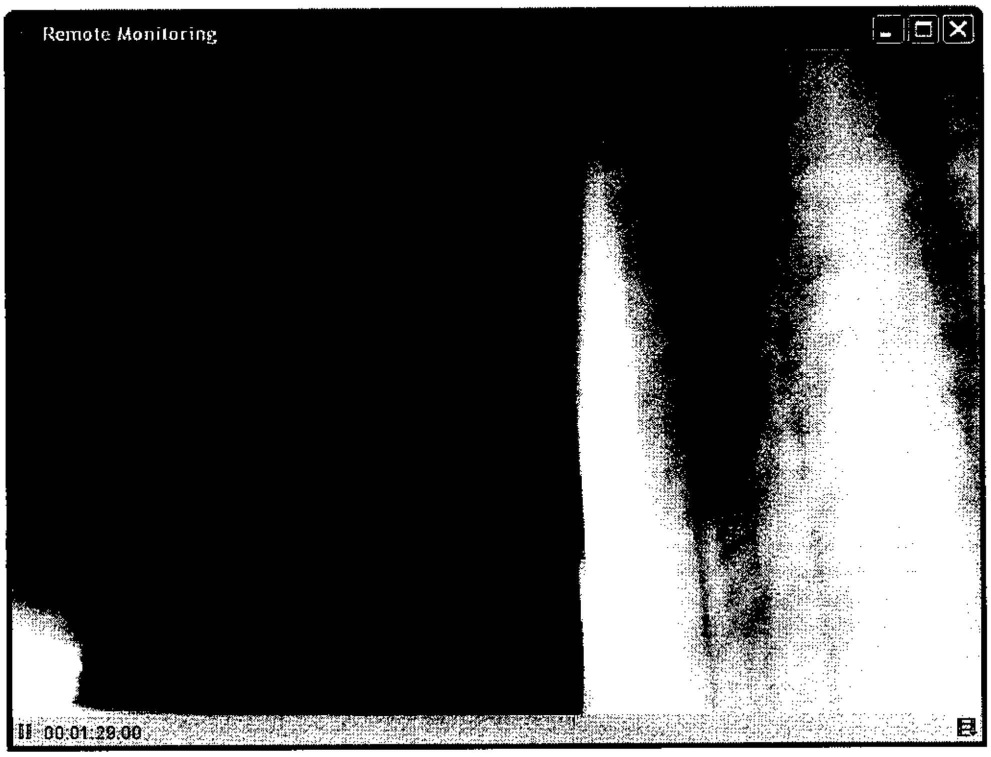
Password: •••••

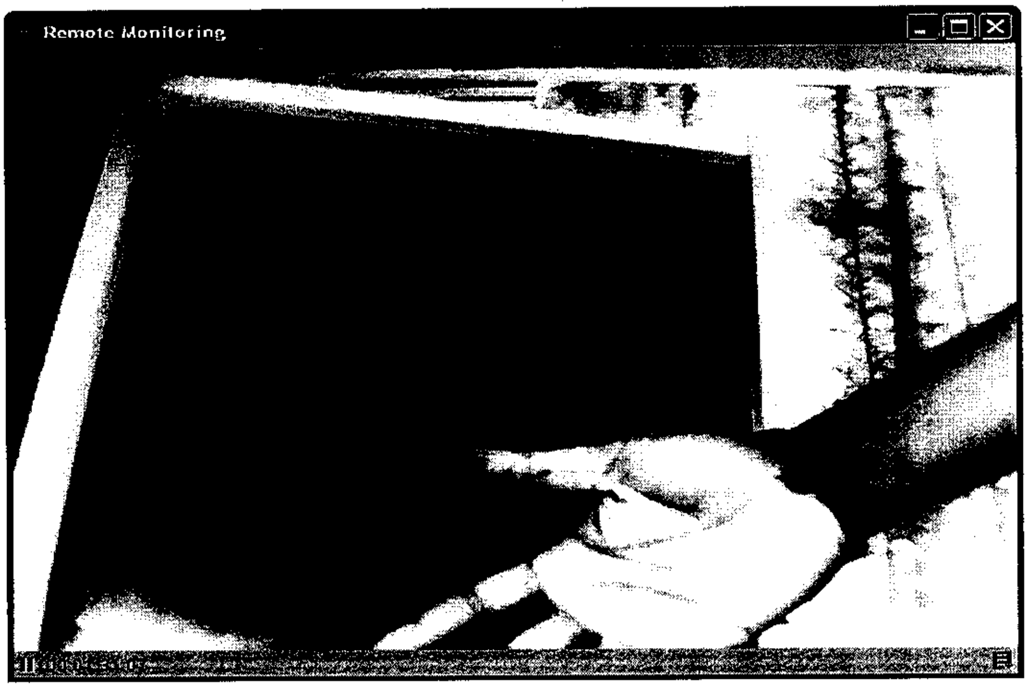
WEBCAM:



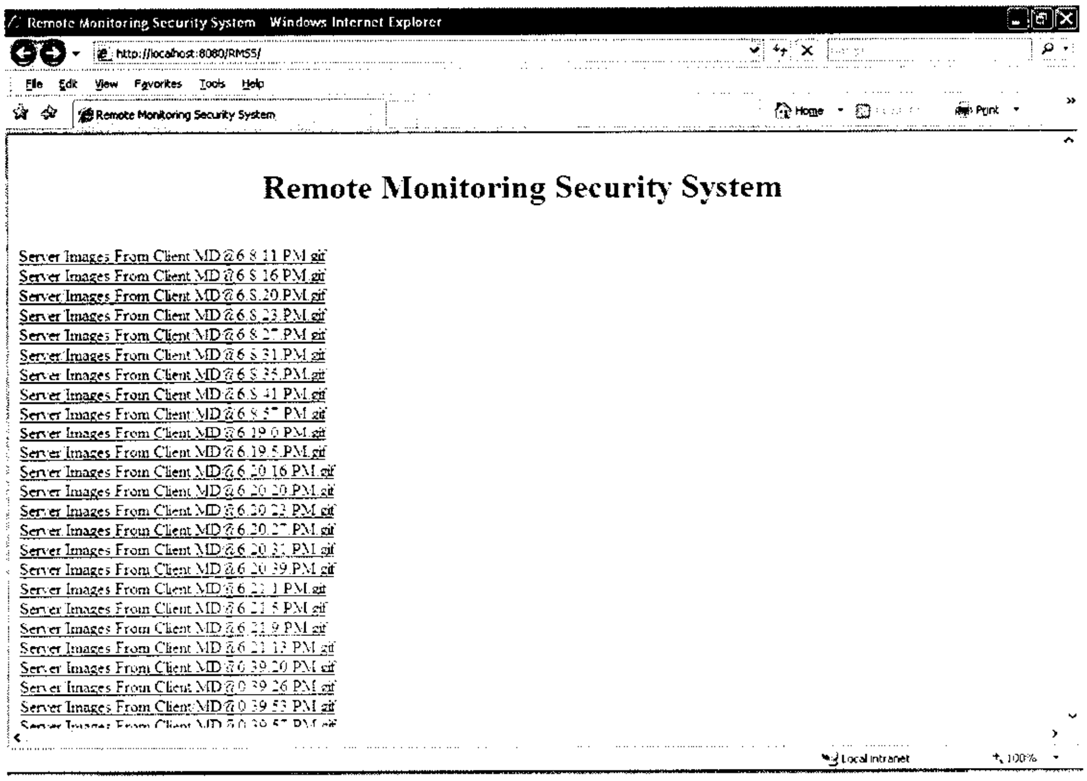
```
E:\suresh project\[illegible]\Client>javac [illegible]
'javac' is not recognized as [illegible]
operable program or batch file.

E:\suresh project\[illegible]\Client>[illegible]
```

TIME STAMP EFFECT:



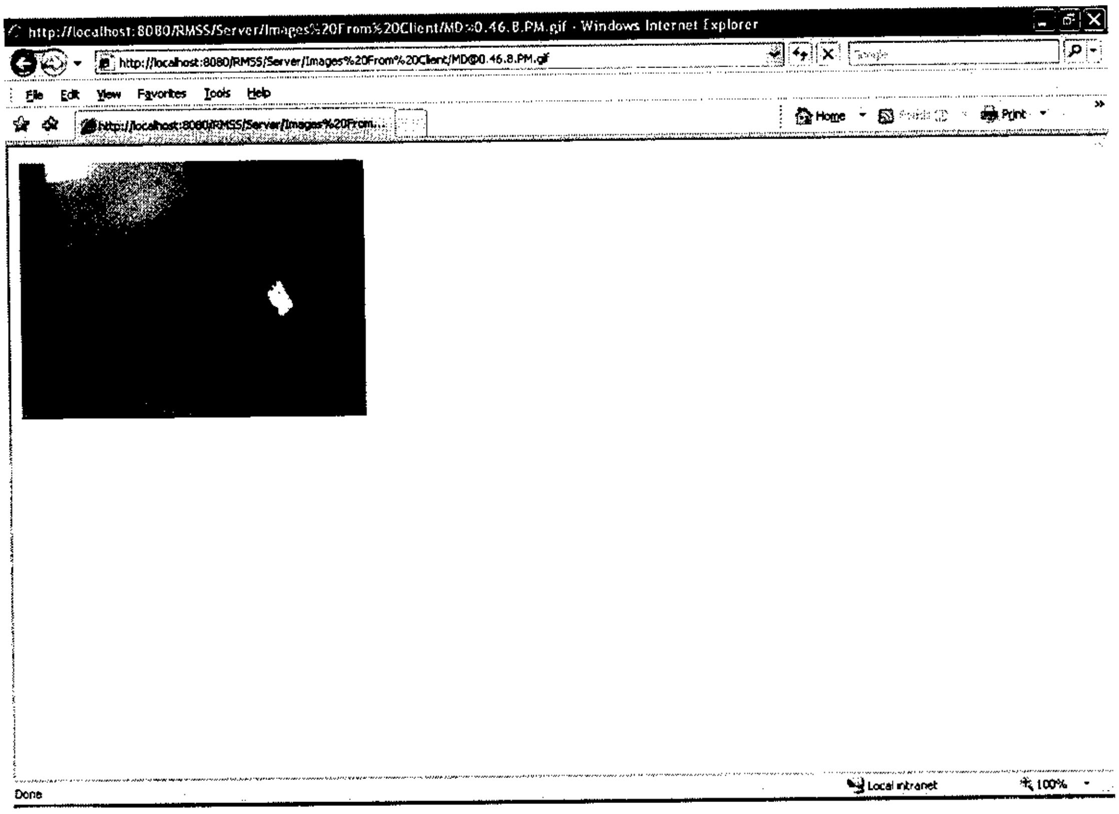


WEB PAGE :


Remote Monitoring Security System - Windows Internet Explorer
http://localhost:8080/RMSS/
File Edit View Favorites Tools Help
Remote Monitoring Security System
Home
Print
Local intranet
100%

Remote Monitoring Security System

Server Images From Client MD@6.8.11 PM.gif
Server Images From Client MD@6.8.16 PM.gif
Server Images From Client MD@6.8.20 PM.gif
Server Images From Client MD@6.8.23 PM.gif
Server Images From Client MD@6.8.27 PM.gif
Server Images From Client MD@6.8.31 PM.gif
Server Images From Client MD@6.8.35 PM.gif
Server Images From Client MD@6.8.41 PM.gif
Server Images From Client MD@6.8.57 PM.gif
Server Images From Client MD@6.19.0 PM.gif
Server Images From Client MD@6.19.5 PM.gif
Server Images From Client MD@6.20.16 PM.gif
Server Images From Client MD@6.20.20 PM.gif
Server Images From Client MD@6.20.23 PM.gif
Server Images From Client MD@6.20.27 PM.gif
Server Images From Client MD@6.20.31 PM.gif
Server Images From Client MD@6.20.39 PM.gif
Server Images From Client MD@6.21.1 PM.gif
Server Images From Client MD@6.21.5 PM.gif
Server Images From Client MD@6.21.9 PM.gif
Server Images From Client MD@6.21.13 PM.gif
Server Images From Client MD@6.39.20 PM.gif
Server Images From Client MD@0.39.26 PM.gif
Server Images From Client MD@0.39.53 PM.gif
Server Images From Client MD@0.39.57 PM.gif


http://localhost:8080/RMSS/Server/Images%20From%20Client/MD@0.46.8.PM.gif - Windows Internet Explorer
http://localhost:8080/RMSS/Server/Images%20From%20Client/MD@0.46.8.PM.gif
File Edit View Favorites Tools Help
Home
Print
Done
Local intranet
100%

REFERENCES:

1.Davide A. Migliore Matteo Matteucci Matteo Naccari. A revaluation of frame difference in fast and robust motion detection. Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks, 2006, pp:215 – 218.

2. Ewerth, R.Freisleben, B. Siegen Univ., Germany. Frame difference normalization: an approach to reduce error rates of cut detection algorithms for MPEG videos. Proceedings 2003 International Conference,2003, pp:11-12.

3. I. Koprinska and S. Carrato, "Temporal Video Segmentation: A Survey" in Signal Processing: Image Communication 16 (2001), pp. 477-500, 2001.