

Enhancing Surveillance Capabilities: A Low-Cost, Mobile Phone-Based Solution with Moving Object Recognition

Dr. Bhargav Gangadhara

Senior Technical lead, Jack henry and associates, USA

Email: [bhargavmtechem\[at\]gmail.com](mailto:bhargavmtechem[at]gmail.com)

Mob.no:6514947376

Abstract: *This study introduces a cost-effective, mobile phone-based surveillance system utilizing moving object recognition technology. The system, designed for both security and environmental monitoring, incorporates a background subtraction algorithm optimized for mobile phone capacities. A self-adaptive background model enhances environmental adaptability. The systems effectiveness is demonstrated through a prototype developed using J2ME technology, highlighting its potential for widespread application in mobile security. Firstly, the basic principles of moving object detection are presented. Limited by the memory consuming and computing capacity of a mobile phone, a background subtraction algorithm is presented for adaptation. Then, a self-adaptive background model that can update automatically and timely to adapt to the slow and slight changes of natural environment is detailed. When the subtraction of the current captured image and the background reaches a certain threshold, a moving object is considered to be in the current view, and the mobile phone will automatically notify the central control unit or the user through phone call, SMS (Short Message System) or other means. The proposed algorithm can be implemented in an embedded system with little memory consumption and storage space, so it's feasible for mobile phones and other embedded platforms, and the proposed solution can be used in constructing mobile security monitoring system with low-cost hardware and equipments. Based on J2ME (Java2 Micro Edition) technology, a prototype system was developed using JSR135 (Java Specification Requests 135: Mobile Media API) and JSR120 (Java Specification Requests 120: Wireless Messaging API) and the test results show the effectiveness of proposed solution. The moving object recognition technology led to the development of autonomous systems, which also minimize the network traffic. With good mobile ability, the system can be deployed rapidly in emergency. And can be a useful supplement of traditional monitoring system. With the help of J2ME technology, the differences of various hardware platforms are minimized. All embedded platforms with camera equipped and JSR135/JSR120 supported can install this system without making any changes to the application. Also, the system can be extended to a distributed wireless network system. Many terminals work together, reporting to a control center and receiving commands from the center. Thus, a low-cost wide-area intelligent video surveillance system can be built. Furthermore, with the development of embedded hardware, more complex digital image process algorithms can be used to give more kinds of applications in the future.*

Keywords: Mobile phone-based, moving object detecting, background subtraction algorithm, Java to Micro Edition technology, JSR135/JSR120, distributed wireless network system

1. Introduction

The increasing need for intelligent video surveillance in public, commercial and family applications makes automated video surveillance systems one of the main current application domains in computer vision. Intelligent Video Surveillance Systems deal with the real-time monitoring of persistent and transient objects within a specific environment.

Intelligent surveillance system has been evolution to third generation, known as automated wide-area video surveillance system. Combined computer vision technology, the distributed system is autonomous, which can also be controlled by remote terminals. A low-cost intelligent wireless security and monitoring solution using moving object recognition technology is presented in this paper. The system has good mobility, which makes it a useful supplement of traditional monitoring system. It can also perform independent surveillance mission and can be extended to a distributed surveillance system. Limited by the memory consuming and computing capacity in a mobile phone, background subtraction algorithm is presented to be adopted in mobile phones. In order to be adapted to the slow and slight changes of the natural environment, a self-

adaptive background model updated automatically and timely is detailed. When the subtraction of the current captured image and the background reaches a certain threshold, a moving object is thought to be in the current view, and the mobile phone will automatically notify the central control unit or the user through phone call, SMS, or other means. Based on J2ME technology, we use JSR135 and JSR120 to implement a prototype [1].

1.1. Limitations

- Depends on the mobile phone camera used by the user (i.e. the better the pixels of the particular mobile phone better the clarity of the picture).
- Can be used only with mobile phones supporting J2ME applications like we can use all the major phone vendors (Nokia, Motorola, Sony-Ericsson, etc), which has OS like Symbian.
- On the server side a static IP address is required to receive pictures from the mobile phone (client) and also to send message to the mobile phone (administrator).

1.2 Problem Definition

Volume 12 Issue 12, December 2023

www.ijsr.net

Licensed Under Creative Commons Attribution CC BY

To develop a J2ME-Based Wireless Intelligent Video Surveillance System using Moving Object Recognition Technology.

The purpose of this document is to demonstrate the requirements of the project "Motion Detector". The document gives the detailed explanation of both functional and non-functional requirements. With this application the mobile phone user should be able to do the following things:

- Can process the consecutive snapshots to find the motion detection.
- Can transmit the captured image to the server.
- Can reset or disconnect the mobile application from the server.

The application should have user-friendly GUI. The communication between the phone and the computer should take place through GPRS. This means that the mobile can transmit the information to a remote user.

The mobile part of the application should be written in J2ME as it provides very good GUI.

1.3 Requirement Specifications

The basic purpose of software requirement specification (SRS) is to bridge the communication between the parties involved in the development project. SRS is the medium through which the users' needs are accurately specified; indeed SRS forms the basis of software development. Another important purpose of developing an SRS is helping the users understand their own needs.

In this chapter the requirements to develop program namely the hardware requirements, software requirements, functional requirements and non-functional requirements are specified so that the user program deploy the program with ease.

1.4 Functional Requirements

- To protect a room from intruder attack.
- To respond when the room is under attack.
- To prevent most attacks from succeeding.

1.5 Non Functional Requirements

1.5.1 Hardware Interfaces

- Computer: PC installed with Internet connection with unique IP address. The application should be running to accept the request and to send response to mobile based on the option.
- Mobile: GPRS enabled mobile with CLDC 1.1 and MIDP 2.0 or higher version.

1.5.2 Software Interfaces

- Network: Internet.
- Application: Java, J2ME, NetBeans 1.6.
- Operating System: Windows XP.
- J2ME Wireless Toolkit 2.5.1

1.5.3. Communications Interfaces

- GPRS will be used as Communication interface between server and mobile.

2. Detailed Design

2.1 Sequence Diagram

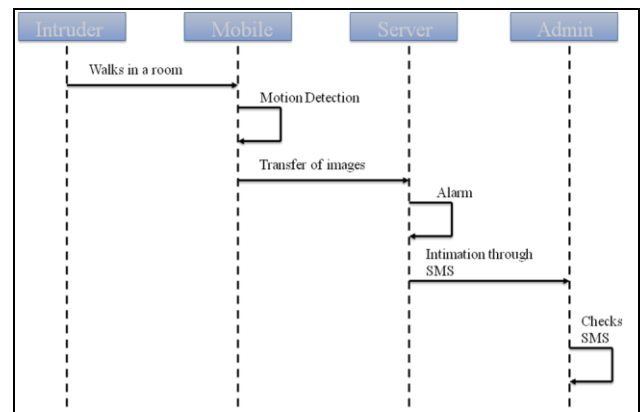


Figure 1: Sequence Diagram

2.2 Activity Diagram

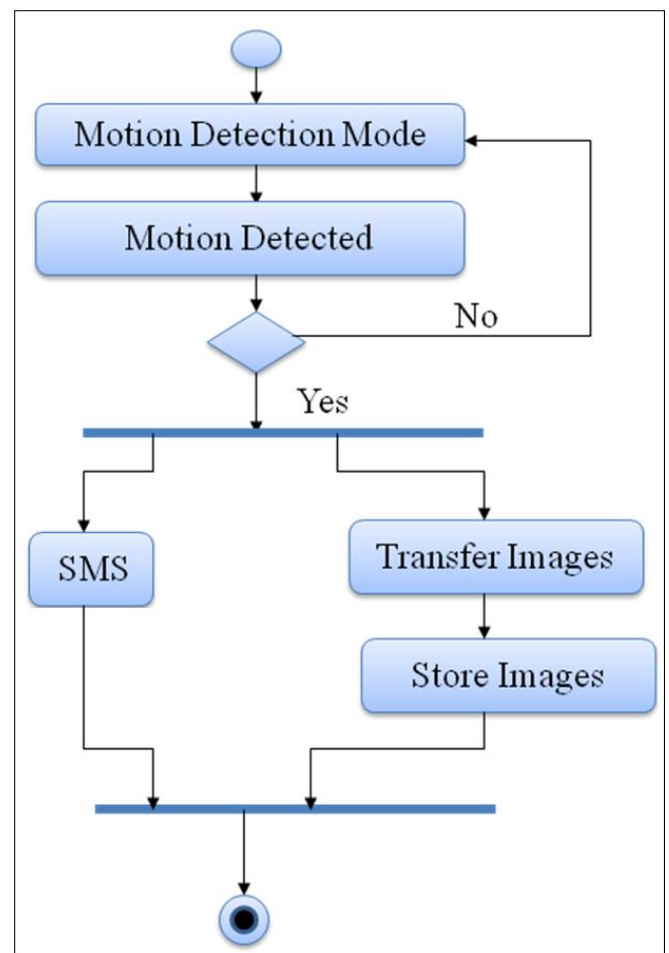


Figure 2: Activity Diagram

2.3 Deployment Diagram

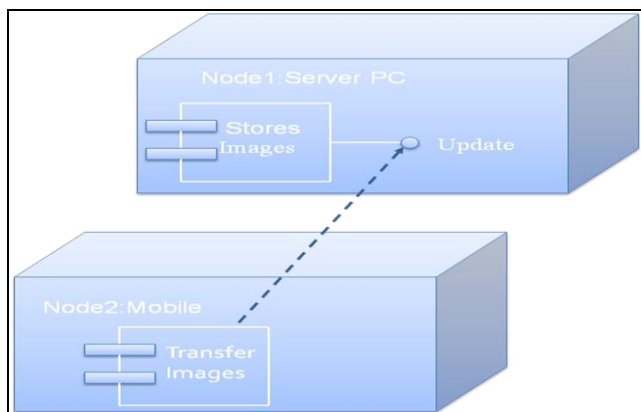


Figure 3: Deployment Diagram

2.4 Piximity Algorithm

This algorithm is used to calculate the difference between the first snap and consecutive snaps. We take the RGB value of each pixel of a image and store each different combination in a Colour array, we then calculate the Colour count of each of these values and store it in a Colour count array. These are explained below in figure

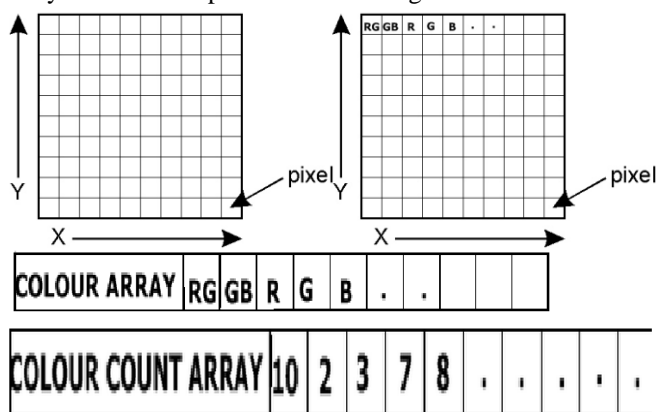


Figure 4: Piximity Algorithm

3. Modules

3.1 Server Side

3.1.1 Module Name: MotionDetectorWelcome

This module is used for displaying the welcome screen when the application is launched on the server side.

Input: None

Step1: Obtain the image from the disk

Step 2: Set boundaries for the image

Step 3: Set time duration for which the image must be displayed

Step 4: End

3.1.2 Module Name: MotionDetectionProcessor

This module is used for receiving the image, storing the image, displaying it on the canvas & calling the SendMail module which sends a message to particular number.

Input: Command, Arguments

Step1: Obtain the inputstream from the client side

Step 2: Check the command received in the stream

Step 2.1: If command is to receive image then call getImage() function which receives the image

Step 2.2: If command is motion detection alarm then call the function MotionDetectionSuit which alarms at the server side and sends a message to the specified user

Step 3: End

3.1.3 getImage

Step 1: Display a transmitting bar in the canvas on server side

Step 2: Read the image in form of bytes from the input stream

Step 3: Call the StoreImage module which stores the byte array in the form of JPEG in the disk

Step 4: End

3.1.4 Module Name: MotionDetectionSuit

This module is a supporting module for Motion Detection Processor which is used to call the audio file placed in the disk, create the directory name for storing image.

Step 1: Call the audio file and play it thrice with a gap of 1 min between them

Step 2: Get the current date time and give that as the name to the directory

Step 3: Call the inbuilt File module to create the directory

Step 4: End

3.1.5 Module Name: StoreImage

This module is a supporting module for Motion Detection Processor which is used to store the image in the directory created and print on the screen that is command prompt if image is created or not it is stored in jpeg format.

Step 1: Print on the command prompt which directory the image is being stored

Step 2: Use the FileOutputStream inbuilt module to store the jpeg image in the directory

Step 3: Using Write function we store the bytes in to jpeg file

Step 4: End

3.1.6 Module Name: MotionDetectorNavigator

This module implements the server side GUI which gives option to Disconnect, Reset Connection, Play the stored video at three different speeds, and even run loop.

Step 1: Create the GUI by extending the module to JFrame

Step 2: Create the Checkbox, Create buttons for play video, change the connection

Step 3: End

3.1.7 Module Name: PlayVideoChooser

This is a supporting module for MotionDetectionServer which gives the user option to play from a set of videos stored

Step 1: This module extends JFrame which is used to display the list.

Step 2: When the user clicks on a video file the file from it played using PlayVideo module

Step 3: End

3.1.8 Module Name: PlayVideo

This is a supporting module for PlayVideoChooser which obtain the video and plays it.

Step 1: Obtain the video from the disk
 Step 2: Play the video by using DrawImage inbuilt function
 Step 3: End

3.1.9 Module Name: SendMail

This module is used to send sms to a airtel number using a gmail id

Step 1: Establish a smtp connection with gmail server
 Step 2: Mention user name and password of the gmail account
 Step 3: Mention the mobile number
 Step 4: Mention the message to be sent
 Step 5: End

3.2 Mobile Side

3.2.1 Module Name: WelcomeScreen

This module is used to display image on the mobile before the launch of the application with ok and exit as two options

Step 1: Obtain the image from the disk
 Step 2: Align it to the center of the screen
 Step 3: Add two commands OK and EXIT
 Step 4: End

3.2.2 Module Name: RecordStorage

This module is used to store the values of the time, ipaddress, threshold

Step 1: If there is Records already stored then we take it from the records or we take it from user
 Step 2: We would have three functions to receive time, threshold, ipaddress
 Step 3: End

3.2.3 Module Name: MotionDetectionScheduler

This module is used to schedule the application so that it will capture the motion after some time as scheduled

Step 1: Here we use sleep the function to delay for the time as specified
 Step 2: After the scheduled time we call the capture module
 Step 3: End

3.2.4 Module Name: MotionCapture

This module is used to calculate the difference between the two images of the first image and consecutive images if the difference between them is greater than threshold then alarm signal will be sent. This is a supporting module for above to modules it uses piximity algorithm.

Step 1: We check if the snap received is the first one we take it as test snap
 Step 2: From then on we take the first image as back ground image & then calculate the difference of consecutive images using Piximity algorithm and is MotionDetected()
 Step 3: End

3.2.5 Piximity Algorithm

```
int piximityAlg(int image[])
```

```
{
    int LENGTH = image.length;
    int color_array[] = new int[LENGTH];
    int color_count[] = new int[LENGTH];
    int temp;
    int array_count=1;
    boolean color_found=false;
```

```
    color_array[0]=image[0];
    for(int i=0;i<LENGTH;i++)
    {
        for(int j=0;j<=array_count;j++)
        {
            color_found=false;

            if(color_array[j]==image[i])
            {
                color_count[j]=color_count[j]+1;

                color_found=true;
                break;
            }
        }
        if(!color_found)
        {
            array_count=array_count+1;
            color_array[array_count]=image[i];
            color_count[array_count]=1;
        }
    }
    return array_count;
}
```

3.2.6 isMotionDetected().

Step 1: Here we use sleep the function to delay for the time as specified
 Step 2: After the scheduled time we call the capture module
 Step 3: End

3.2.7 Module Name: AlertMessage

This is a supporting module for MotionCapture used to send the AlertMessage

Step 1: We create a socket connection with the server using the ipaddress given by the user
 Step 2: We use output data stream to write on the server side
 Step 3: End
 Step 6: Done.

4. Results

4.1 Server Side

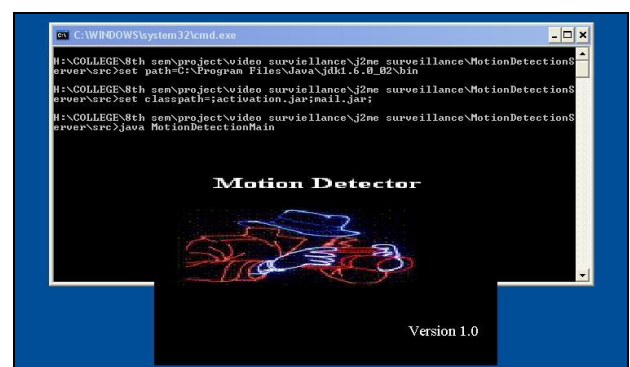


Figure 5: Start up screen

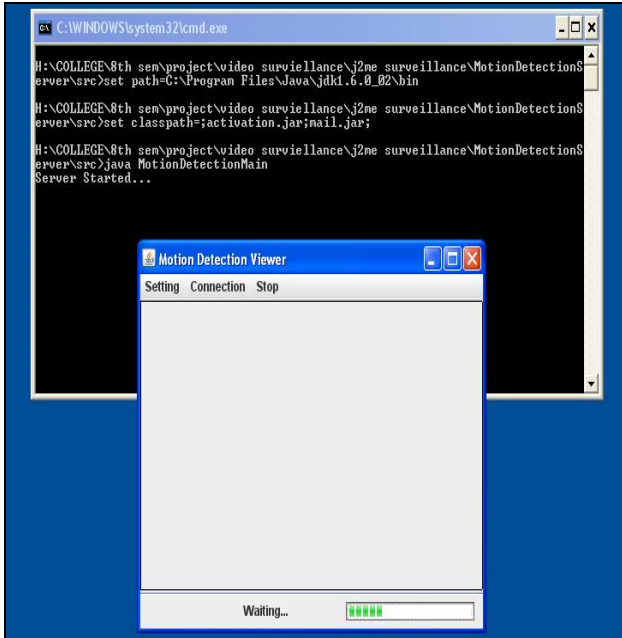


Figure 6: Motion detector viewer



Figure 7: The image shows that a message was sent, the directory where image is being stored is mentioned

4.2 Mobile Side

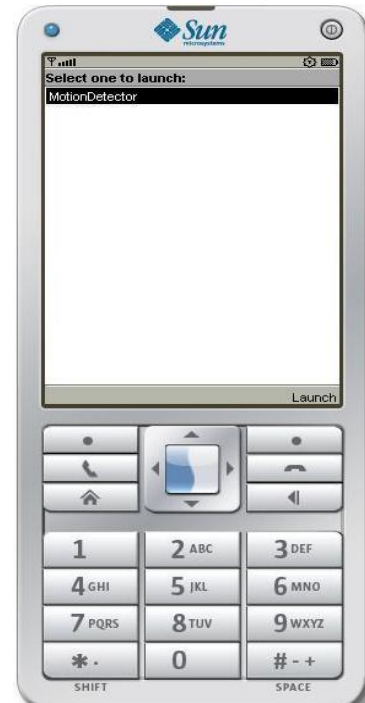


Figure 5: The user has to launch the application



Figure 9: The welcome screen is being launched

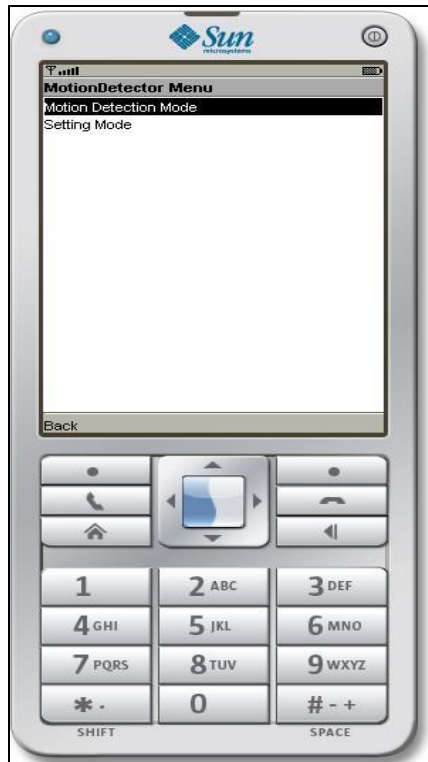


Figure 10: User has to choose between two modes



Figure 6: The Difference between images is being calculated

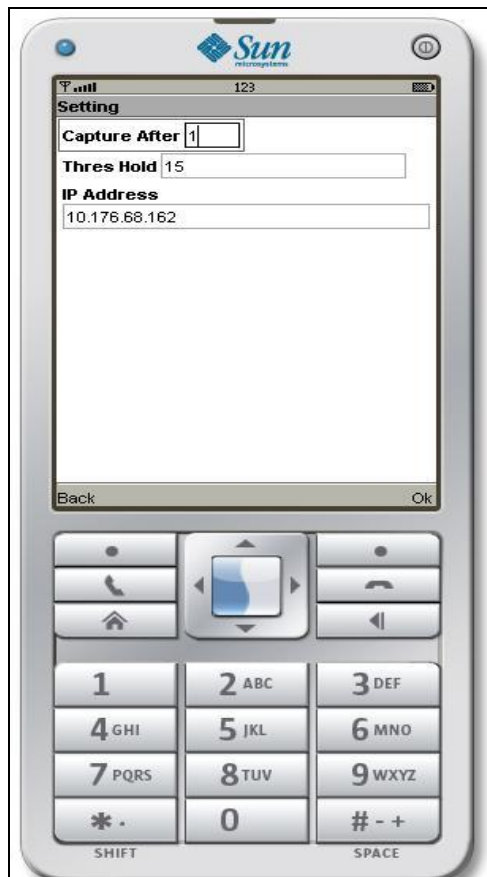


Figure 11: User enters the three options Capture After, Threshold, and IP Address



Figure 7: SMS received by the user

5. Future Enhancements

The current implementation of security in Wireless Vigilance can be enhanced with the following features. Some of the many possible future enhancements are:

- Storing images in other format than JPEG
- Quality of the image captured can be improved
- Messages can be sent to a group of users instead of one

There might be more scope for improvement, which could not have come to our notice. We would definitely keep track of the feedback for people's opinion and would try and implement to bring out the necessary improvements to appeal others.

6. Conclusion

This article's significance lies in its presentation of a low-cost, efficient surveillance solution adaptable to various environments, offering a significant advancement in the field of mobile-based security systems. The moving object recognition technology led to the development of autonomous systems, which also minimize the network traffic. With good mobile ability, the system can be deployed rapidly in emergency. And can be a useful supplement of traditional monitoring system. With the help of J2ME technology, the differences of various hardware platforms are minimized. All embedded platforms with camera equipped and JSR135/JSR120 supported can install this system without making any changes to the application. Also, the system can be extended to a distributed wireless network system. Many terminals work together, reporting to a control center and receiving commands from the center. Thus, a low-cost wide-area intelligent video surveillance system can be built. Furthermore, with the development of embedded hardware, more complex digital image process algorithms can be used to give more kinds of applications in the future.

References

- [1] M Valera, SA Velastin, Intelligent distributed surveillance systems: a review. IEE Proceedings on Visual Image Signal Processing, April. 2005, vol. 152, vo.2, pp.192-204.
- [2] [M. Picardy, Background subtraction techniques: a review, IEEE International Conference on Systems, Man and Cybernetics, Oct. 2004, vol. 4, pp. 3099–3104.
- [3] [Alan, M. McIvor, Background Subtraction Techniques, Proceedings of Image & Vision Computing New Zealand 2000 IVCNZ'00, Reveal Limited, Auckland, New Zealand, 2000.
- [4] C. Enrique Ortiz, The Wireless Messaging API developers.sun.com, 2002
- [5] [Sun Microsystems, Inc., JSR 120 Wireless Messaging API (WMA) specification, 2002.
- [6] Herbert Schildt Java™ 2: The Complete Reference Java 2, TATA McGraw Hill, Fifth edition, 2002
- [7] Patrick Naughton, The java Handbook, Tata McGraw-Hill Edition, 1997
- [8] <http://computer.howstuffworks.com/program.htm>
- [9] <http://ieeexplore.ieee.org/iel5/4566097/4566248/04566311.pdf?arnumber=4566311>
- [10] <http://ieeexplore.ieee.org/iel5/90/4542818/04542819.pdf?arnumber=4542819>
- [11] <http://www.geom.uiuc.edu/~daeron/docs/apidocs/java.io.html>
- [12] <http://www.geom.uiuc.edu/~daeron/docs/apidocs/packages.html>
- [13] <http://www.kbs.twi.tudelft.nl/Documentation/Programming/Java/jdk1.4/api/javafx/swing/JComponent.html>

- [14] <http://www.kbs.twi.tudelft.nl/Documentation/Programming/Java/jdk1.4/api/javafx/swing/package-summary.html>