A&QT-R 2002 (THETA 13)
2002 IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics
May 23-25, 2002, Cluj-Napoca, Romania

SURVEILLANCE CONSIDERING JAVA MEDIA FRAMEWORK

Authors: Mircea-Florin Vaida, Liviu Lazar

Technical University of Cluj-Napoca, Gh. Baritiu str. 25-27, Romania
E-mail: Mircea.Vaida@com.utcluj.ro , Liviu.Lazar@email.ro

ABSTRACT: Considering the numerous advantages of the Java platform, we developed a dedicated application that implements a surveillance system. The application will emulate this system in a real-time environment, providing facilities like intrusion detection and single frame capturing. The user interface also provides intuitive and easy access to the image archive, allowing operations of zooming and printing.

Keywords: Java Media Framework, surveillance system, movement detection, image archive browsing

1. INTRODUCTION

The Java Media Framework (JMF) API specifies a simple, unified architecture, messaging protocol and programming interface for media recording, transmission and playback.

JMF, currently at version 2.1.1, is Sun's initiative to bring time-based media processing to Java. Time-based media is data that changes meaningfully with respect to time, such as audio and video clips, MIDI sequences, and animations. Also a complete reference implementation, JMF 2.1.1 enables users to do anything imaginable with multimedia. Among others uses, JMF can:

- Play various multimedia files in a Java applet or application. The formats supported include AU, AVI, MIDI, MPEG, QuickTime, and WAV;
- Play streaming media from the Internet;
- Capture audio and video with microphone and video camera, then store the data in a supported format;
- Process time-based media and change the content-type format;
- Transmit audio and video in realtime on the Internet;
- Broadcast live radio or television programs. [1]

2. THE GHOST SURVEILLANCE SYSTEM APPLICATION

The application's goal is to realize a surveillance process over an interest area and automatically save frames after movement detection, [2].

A basic security access control is implemented as you start the application. The user is asked to enter a password, which is by default "UTCN". After authentication the application displays the application's logo and copyright information (**Figure 1**).

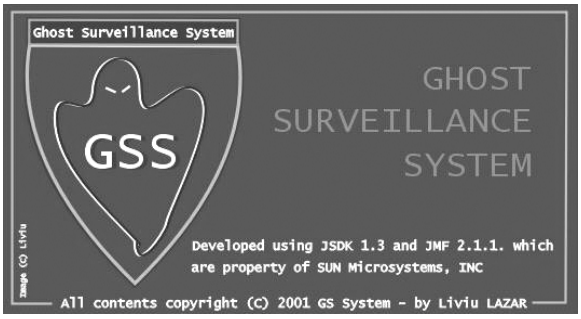

Figure 1. Application's logo and copyright information

The main page of the application is used for setting up the surveillance parameters (**Figure 2**). The components are:

- *Live video window*: it displays continuously the pictures received from the video camera. The window's size is 320x240 pixels. Pushing the Pause switch can interrupt the process.
- *Information window*: it displays useful information like current date and time, for how long is the application running and number of intruders detected. The information is updated automatically.
- *The application's logo* is displayed in the upper right corner.
- *Sensitivity adjustment*: it allows the adjustment of the surveillance sensitivity. The values range is between 1 and 100. A value of 0 means that the difference between two consecutive frames should be 100% in order to detect movement. A value of 100 means that the difference should be 0 in order to detect movement and save the frames. In this case the application saves all the frames. The tests leaded to an optimal value of 65.
- *Verification rate adjustment*: it allows the adjustment of the period between two frames comparison. The values are between 1 and 10. A value of 1 means that the comparison is made every 2 seconds. A value of 10 means that the comparison is made every 0.2 seconds.
- *Surveillance history*: it provides useful information about the difference between frames. It is activated only when the surveillance is activated. A red line shows the sensitivity threshold (which can be modified using the verification rate adjustment). By yellow rectangles is represented the difference between consecutive frames. If the difference passes the sensitivity threshold, the application automatically saves the last analysed frame. This module is very useful for system calibration in different environments. The file name contains the exact time of capture (at milliseconds precision) and it is stored in a folder named by the date of the capture.
- *Link to Images submenu*: links to a submenu that allows browsing through image database. The user can visualize, zoom and print images.
- *Capture now*: the application captures an independent frame, even if the surveillance is OFF.

- *Surveillance switch (ON/OFF)*: it allows the user to start/stop the surveillance system.

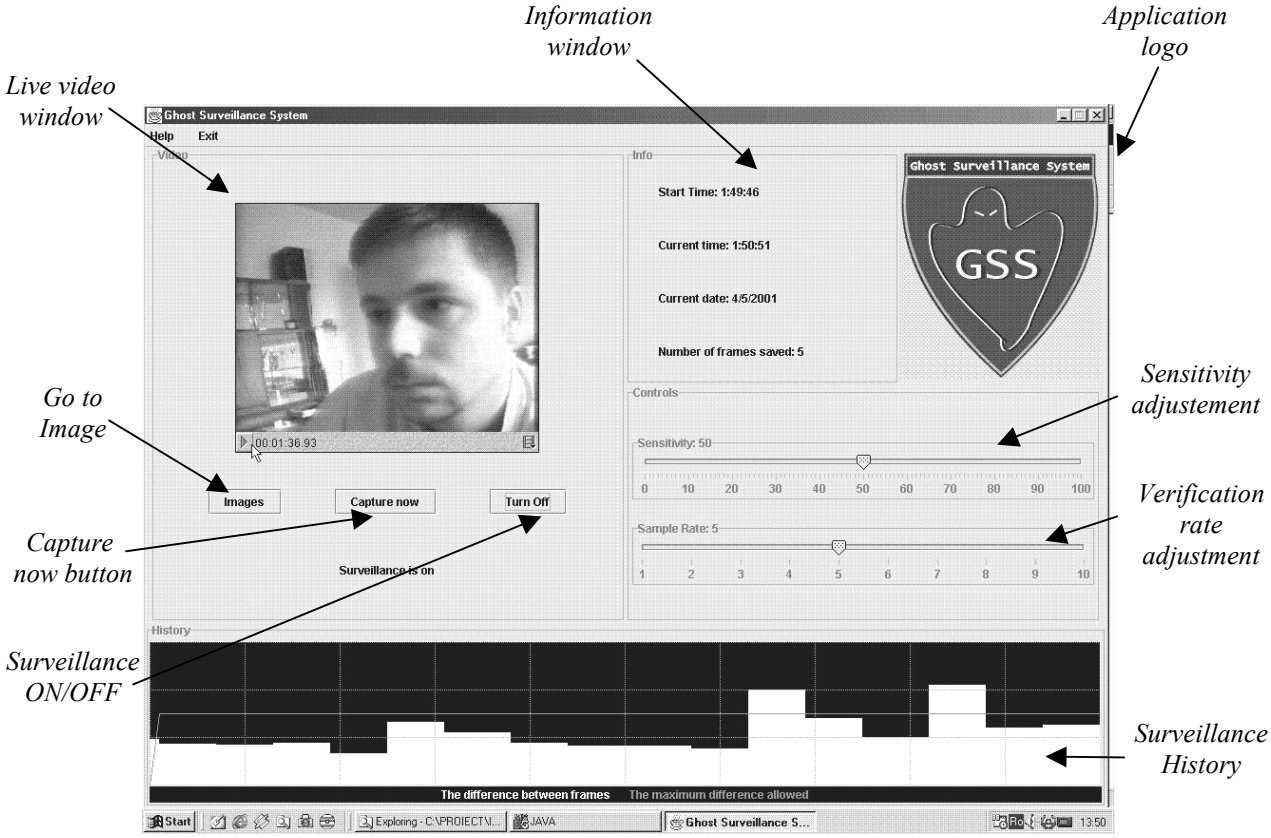

Figure 2. Main menu of Ghost Surveillance System application

The *Images* submenu (**Figure 3**) allows the user to browse the image database and processes the images. The components are:

- On the top left side are displayed thumbnails of the images (*images preview*). By clicking on one of those images, the application displays the full size image on the right side of the screen.
- Under the full size image are displayed informations about the accurate *date and time* when the image was captured. The informations are automatically extracted from the file name and directory of the image file.
- The *Zoom button*: allows users to magnify an area of interest. Mouse dragging selects the area. The final step is to press the Zoom button. This is very useful the after surveillance process (the user can magnify the face of the intruder).
- After Zoom the user can reload the original image by pressing the *Revert button*
- The user can print the image in the last state (zoomed or normal) using the *Print button*
- By pressing the *Back button* the system returns to main page of the application.

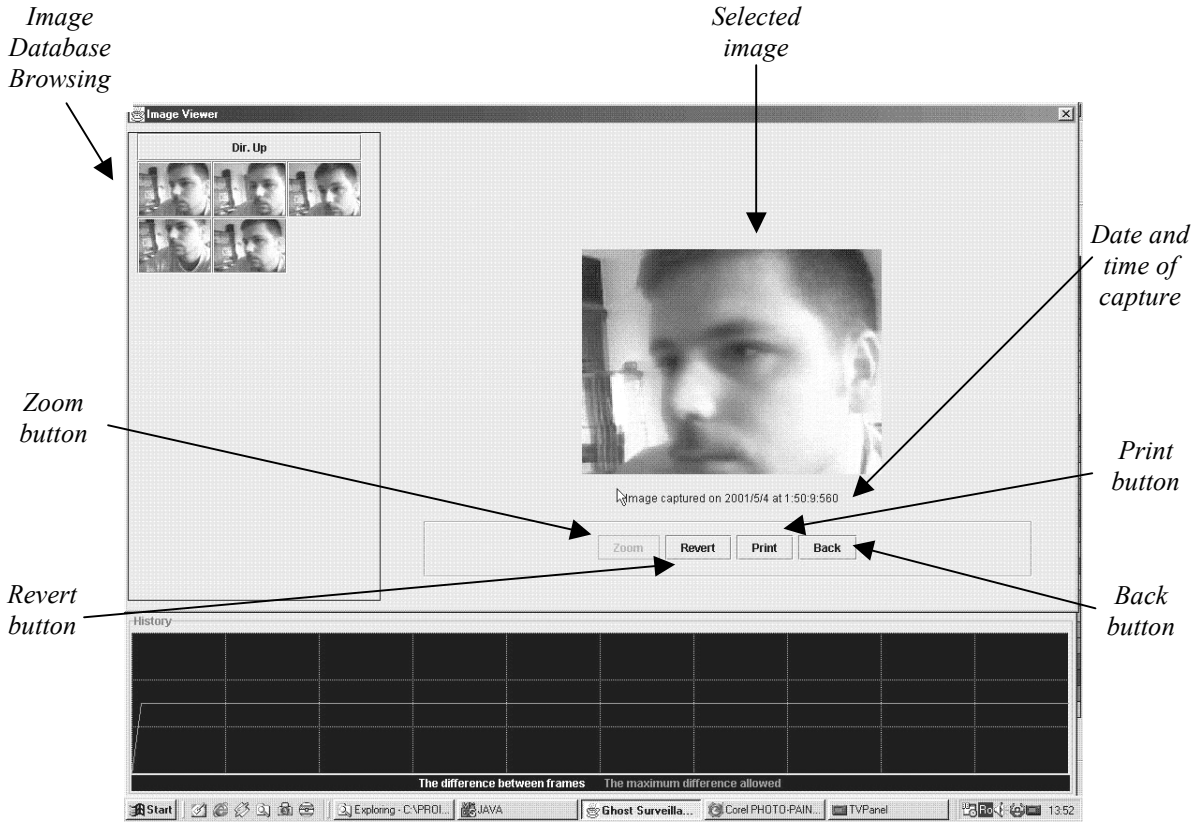

Figure 3. *Images* submenu

Calculating the average of differences between two consecutive analyzed frames makes the movement detection. The difference is calculating using an algorithm based on the XOR operator. The average is then compared to the sensitivity (which is set by the user). If the average is lower than the sensitivity, the frame is ignored, otherwise is considered like an intrusion and the frame is saved.

3. ADVANTAGES AND DISADVANTAGES OF GSS

The *advantages* of Ghost Surveillance System application are:

- The application runs on any system with JDK 1.3 and JMF 2.1.1. The advantage is that those packages can be downloaded with out charge from www.java.sun.com.
- The Java Platform provide for full portability of the application.
- The saved images are in JPEG (Joint Photographic Expert Group). The JPEG standard converts the captured image in a spatially represented frequency using a DCT (Discret Cosine Transform) 2D. Therefore an image of 320x240 takes about 20 kb on the scratch disk. This allows the user to save about 32500 images on a regular CD.
- The saving process is structured. Each image is stored in a file that contains in its name the exact hour of the capture. The image is structured in a folder

named by the capture date. Because of this the browsing process is very easy and intuitive.

- Because of the optimized surveillance mechanism an intruder is detected in less than 2 seconds. The image analyses and printing takes less than 2 minutes.
- The application's size is about 100 kb

The *disadvantages* of Ghost Surveillance System application are:

- Medium to High hardware requirements: because GSS is making a lot of colour comparison, the processor is highly solicited. A good ventilation system is also indicated on both processor and video card.
- The application was designed to run correctly at 1024x768. Using the application on other resolutions can cause incorrect displaying.

4. CONCLUSIONS

The Ghost Surveillance System application can be used on any computer equipped with a good processor and a video camera. The camera could be pointed to a door (surveillanced area).

The extensibility of Ghost Surveillance System applications is obvious because of the platform used for development (JAVA and JMF). A next version of the application could implement complementary functions like delayed starting of surveillance (so the user have time to leave the surveillanced area) or playing sounds (police siren) if an intruder is detected. Also it could be implemented a module to notify the user trough Internet (SMS, email) about an intrusion. For data transmission it could be used another facility of JMF, RTP (Real-Time Transport Protocol).

REFERENCES

[1] Budi Kurniawan, [2001], Program multimedia with JMF Part 1, http://javaworld.org/javaworld/jw-04-2001/jw-0406-jmf1.html

[2], M.-F Vaida, C. Porumb, R. Fotea, F.O. Hurducas, L. Lazar, Java 2 Enterprise Edition (J2EE). Multimedia Applications, Edit. Albastra, 2002, Cluj-Napoca