

# Recognition Assisted Dynamic Surveillance System Based on OSGi and OpenCV

Jenq-Shiou Leu, Wei-Hsiang Lin, Min-Chieh Yu, Hung-Jie Tzeng

Department of Electronic Engineering

National Taiwan University of Science and Technology, Taipei, Taiwan

jsleu@mail.ntust.edu.tw, M9702108@mail.ntust.edu.tw, M9802114@mail.ntust.edu.tw, M9802138@mail.ntust.edu.tw

**Abstract**— Thanks to the Internet development, a surveillance user can monitor the distant events via a browser on a computer. These events can be even viewed on a mobile phone with the assistance from modern mobile networks. Besides, two well-known platforms – OSGi and OpenCV are widely used for application construction. OSGi is constructed to provide a service platform with high application interoperability while OpenCV is used to provide a bundle of application programming interfaces (APIs) about image processing. In this paper, we design a recognition assisted surveillance system based on OSGi service platform and the OpenCV development platform. The system features dynamic monitoring by use of a camera mounted on an embedded system which is carried by a robot and a J2ME based viewer/commander program on a mobile phone. The successful implementation removes the barrier that hinders the user from clear recognizing the objects in the limited screen of mobile phones.

**Keywords**— Surveillance System, OSGi, OpenCV, J2ME

## I. INTRODUCTION

The surveillance system makes a big progress with the assistance of Internet. The revolutionized system can make surveillance users monitor distant events by an Internet browser. Thus the remote live events can easily be seized at anywhere users can access Internet. In a traditional surveillance system [1, 2], many fixed cameras are pinned at fixed locations to capture events. Users then keep a close watch on the videos sent from these cameras via a monitor at a fixed location. Internet indeed facilitates the remote monitoring but it still lacks some flexibility. If the monitoring application can be extended to a handheld device such as a cellular phone, surveillance users can acquire remote events at anytime and anywhere via any available mobile wireless network, such as the GPRS, Wi-Fi, and UMTS to the developing 3GPP-LTE or WiMAX. On the other hand, the pre-installed cameras at fixed locations might have less flexibility. If the camera can be mounted on a movable device, a surveillance user may view more unseen corners which may not be caught by the fixed cameras. In the past, many efforts are put on improving the inflexibility in a traditional surveillance system. A web-based surveillance system in [3] was proposed for surveillance viewers to perform monitoring and controlling remotely by cellular phones. A network camera equipped on a mobile robot [4, 5] can help capture events in wider and more dynamic angles. However, if the mobile camera and the mobile viewer/commander on cellular phone can be integrated on a

common service platform, the service expandability can be achieved. Still, the limited screen of mobile phones may hinder the user from clear recognizing the objects. In this article, we aim to implement a recognition assisted dynamic surveillance system by integrating the Open Services Gateway Initiative (OSGi) [6] open service platform and adding in the recognition function invoking the APIs provided by the Computer Vision Library OpenCV (OpenCV) [7] development platform. The system also includes a movable camera attached on an embedded system carried by a robot [8] and a viewing J2ME [9] application on a cellular phone. The system architecture overview is shown in Fig. 1.

The rest of this paper is organized as follows. Section 2 illustrates some related techniques about designing our dynamic surveillance system. Section 3 describes the system architecture of the proposed system. Section 4 depicts the operation flow and class diagrams on the system. In Section 5, we evaluate the performance of the proposed system by the actual frame receiving rates and conduct an analytical comparison with other surveillance systems. A brief concluding remark is finally drawn in Section 6.

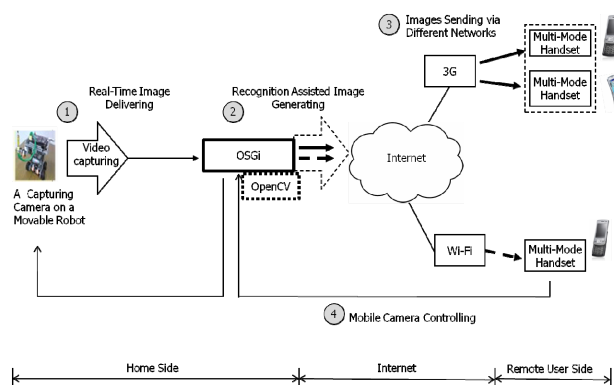


Figure 1: Dynamic Surveillance System Architecture Overview

## II. RELATED WORK

The OSGi alliance was formed to conduct open specifications for delivering services to local networks and devices [10]. Recently, many developments relating middleware construction are based on the OSGi standard [11-13]. The OSGi service platform, a general-purpose, secure, managed Java software framework, expands Internet

services to homes [14-16], automotive [17], health care [18] and so on. OSGi specification defines a service-oriented framework which provides a publishing, finding and binding model for using services so it can support the deployment of extensible and downloadable service applications known as bundles. Bundles are built around a set of collaborative services available from a shared service registry. Therefore, OSGi can play a good broker role to aggregate applications. On the other hand, the OpenCV development platform, which consists of several libraries of C functions and C++ classes, has been used to realize the common image processing and computer vision algorithms including face detection, object segmentation and recognition and so on. As the source codes of OpenCV are simple and efficient, it has been widely used in image processing fields. Therefore, the OpenCV can save some cost with a fast development of image processing related functions.

As hardware costs are coming down dramatically and capabilities of robot are increasing fast, robotics is becoming more important in everyday life. Lego MindStorms series products have been developed by LEGO Company for years to enable developers to have an inexpensive robotic environment. One of its major contributions is to build simple, portable, and inexpensive experiments for proving the concept of a mobile robot service in the development beginning by not having extensive lab facilities [19, 20].

Thanks to the wireless technology revolution, a wireless user can access Internet services at anytime, anywhere via ubiquitous networks by a multi-mode device. In the mobile world, users visualize the cyber net by their handheld mobile phones with a small screen. WAP (Wireless Application Protocol) and J2ME (Java 2 Micro-Edition) [21] are two typical mobile based development technologies. J2ME sets up 3-tier of J2ME Virtual Machine, Connected Limited Device Configuration (CLDC) and Mobile Information Device Profile (MIDP) to achieve modularity and flexibility across many smart phones [22]. Compared with WAP, J2ME has following advantages: direct access to Internet in a client-server mode without a gateway like WAP and more compatible file formats than WAP. J2ME is more suitable to be used to develop a mobile application with higher computation capability. A J2ME-based wireless intelligent video surveillance system [23] was implemented to make a useful supplement of a traditional monitoring system with its good mobility.

In our work, Java, renowned for its openness and portability across many platforms, is chosen as the main developing technique to realize our dynamic surveillance system. That includes building Java-based bundles on the OSGi service platform, invoking the image recognition APIs toward OpenCV, controlling the NXT brick between bundles and a Lego MindStorms NXT over a Bluetooth connection by the iCommand Java package and the RXTX library [24], delivering the captured videos back to OSGi through a Java-based socket program and using J2ME to develop a handy remote surveillance viewer on a multi-mode mobile phone.

### III. SYSTEM ARCHITECTURE

The system architecture and some main application modules in our dynamic remote mobile surveillance system are shown in Fig. 2.

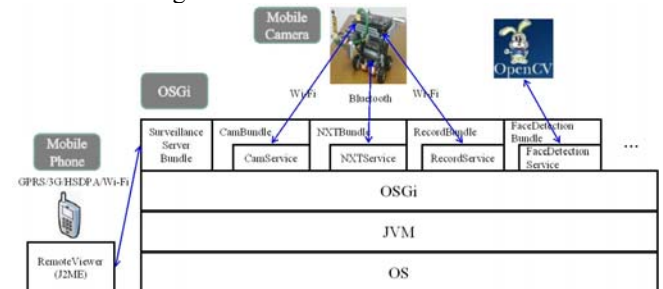


Figure 2: The Proposed Dynamic System Architecture based on OSGi and OpenCV

#### A. OSGi Service Platform

The OSGi platform mainly comprises a JVM (Java Virtual Machine), a set of running applications called bundles, and an OSGi framework. A bundle is a minimal deliverable application in OSGi and managed by the OSGi framework. Meanwhile, a service is the minimal functioning unit in OSGi. Therefore, a bundle is designed as a set of cooperating services, which are discovered after being published in the OSGi service registry.

There are four bundles and three services on the platform:

- **SurveillanceServerBundle:**  
SurveillanceServerBundle is responsible for authorizing users, handling service requests including delivering captured videos to the remote viewer by invoking functions in CamService and controlling the MindStorms NXT as the remote viewer demands by invoking functions in NXTService.
- **CamBundle:**  
CamBundle is responsible for registering, publishing CamService, which works for retrieving the remote surveillance videos, in OSGi.
- **NXTBundle:**  
NXTBundle is responsible for registering, publishing NXTService, which works for controlling NXT, in OSGi.
- **FaceDetectionBundle:**  
FaceDetectionBundle is responsible for registering, publishing FaceDetectionService, which works for recognizing images captured from the mobile camera by call the related APIs in OpenCV, in OSGi.

#### B. Mobile Camera

Lego MindStorms NXT already has a digital camera accessory but the video quality it captures is not good enough for remote viewing. Hence we mount a digital camera on an embedded system - Dmatek ARM9 DMA-NAV2450 which is carried by Lego MindStorms NXT to

capture the events at the locations where the viewer can ask NXT to move to. The camera can also capture an event in a wider range in up and down direction with the assistance of the raising arm, which the camera is hooked on. The physical construction of the mobile camera is shown in Fig. 3.

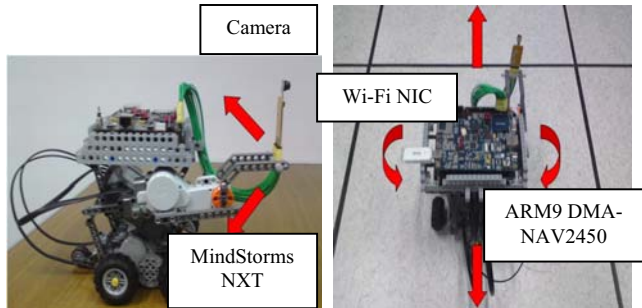


Figure 3: A Mobile Camera Mounted on a Embedded System Carried by a Robot

### C. Mobile Phone

J2ME is used to develop the controlling and viewing functions on a mobile phone. A viewer can use the keypad of mobile phone to remotely control the moving direction of the mobile camera via NXTService as well as to retrieve the surveillance videos captured by the remote camera via CamService.

## IV. OPERATION FLOW AND CLASS DIAGRAMS

About the operation flow of our system, the event is first captured by the mobile camera and sent back to the CamService on OSGi via a Wi-Fi network interface card. After the CamService gets the captured frame, it will invoke the FaceDetectionService to conduct an image recognition process – face detection. Then the system would deliver the recognition assisted images to the mobile phone. Since the limited viewing screen on the mobile phone, we degrade the original captured video frame dimension from 320x240 to meet the screen sizes of mobile phones, such as 235x210 for SonyEriccson G900. The parameter for the frame size is auto-negotiated when the mobile device connects to SurveillanceServerBundle. Frame downsizing can make the surveillance video transmission more effectively. Meanwhile, we also implement an asynchronous viewing mode on RemoteViewer. There are two independent threads on SurveillanceServer in such a mode. One thread is responsible for requesting a frame toward MobileCamera and the other one is for delivering the captured frame from MobileCamera to RemoteViewer. On a fire-and-forget basis, RemoteViewer does not need to repeatedly check to see if a new captured frame arrives. However, due to the different processing capabilities on SurveillanceServer and RemoteViewer, and the different network transmission capabilities on MobileCamera-to-SurveillanceServer and SurveillanceServer-to-RemoteViewer, the received frame on RemoteViewer would gradually get mismatched with the captured frame on MobileCamera, especially for a larger accumulated delay. Therefore, we add a reasonable pause time control on SurveillanceServer to relieve the annoying effect caused by the asynchronous mode. SurveillanceServer would wait for a pause time per frame request toward MobileCamera. The related operation flow is shown in Fig. 4.

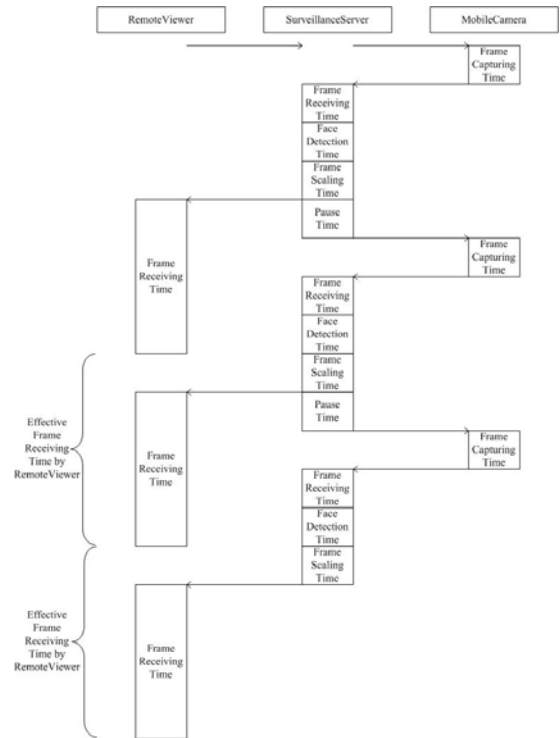


Figure 4: The Operation Flow of the Recognition Assisted Dynamic Surveillance System

Related class diagrams in the system are depicted in Fig. 5. RemoteViewerMIDlet is a J2ME based application framework to define the callback functions based upon user's input. RemoteViewerCanvas is designed to accept users' command to control NXT and show the received surveillance video frames. The CamService, NXTService, and FaceDetectionService define the service interfaces and are registered and published in the OSGi framework by CamBundle, NXTBundle, and FaceDetectionBundle, respectively. SurveillanceServerBundle then invokes functions in CamBundle, NXTBundle, and FaceDetectionBundle to retrieve videos, control NXT, and conduct an image recognition process.

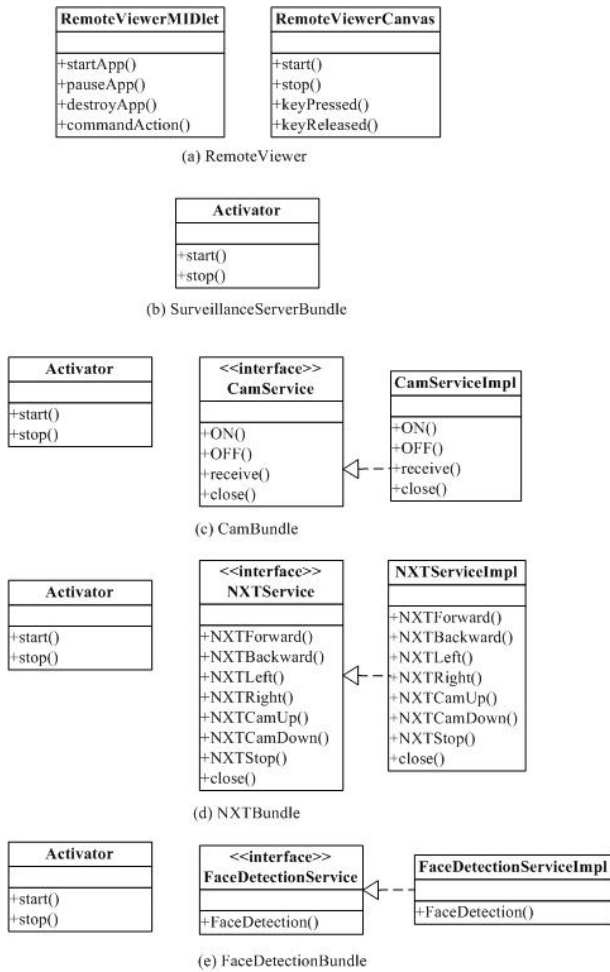


Figure 5: Related Class Diagrams in the Recognition Assisted Dynamic Surveillance System

## V. SYSTEM EVALUATION AND ANALYTICAL COMPARISON

To enhance the mobility and responsiveness, our surveillance system broaches the cellular or wireless transmission to surveillance image delivery between two mobile ends. Recognition assisted images captured by a mobile camera on NXT are sent to a mobile phone for viewing as Fig. 6 shows. The surveillance system would ask the mobile camera mounted to capture an image with an origin dimension - 320x240 (around 10KB) every 0.05 sec (inter-frame gap) pause time. After conducting an image recognition process on the origin image, the system then downsizes the frame dimension according to the screen size of the mobile phone and then sends it to the mobile user for being shown on the mobile phone via a 3G network or a Wi-Fi network which the remote user attaches to.

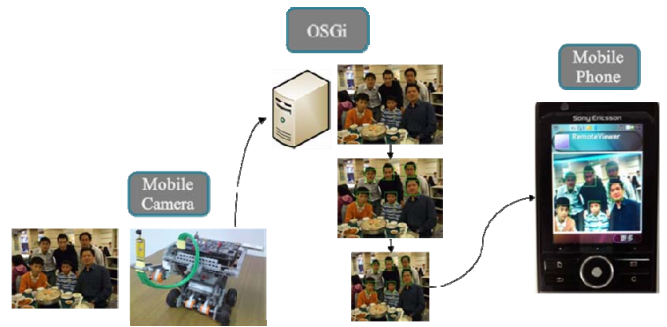


Figure 6: System Evaluation Scenario

Our evaluation focuses on the actual frame receiving rates on the handset via different wireless networks and the evaluation results are shown in Tab.1.

TABLE I. EVALUATION RESULTS

Device	Sony Ericsson G900	
	3G	Wi-Fi
Access Network	3G	Wi-Fi
Source FrameDimension	320x240	
Scaled Frame Dimension	235x210	
Average FaceDetection Time	0.539	0.529
Average Frame Time	0.881	0.749
Average Frame Receiving Rate (f/s)	1.135	1.335

Meanwhile, we compare different surveillance systems in Tab. 2. With the assistance of wireless networks, the dynamic mobile surveillance system features high mobility including the camera end and the viewer/commander end. Mobile viewers can use their snazzy smartphones to issue commands to move the remote movable camera and to capture events at the remote site. Our system also can deliver motion images assisted by recognition processing back to the viewer instead of a still snapshotted image. Meanwhile, as an OSGi and OpenCV based system, which can facilitate the integration of future developed functions and many image processing APIs, the proposed system features high service extensibility as well.

TABLE II. COMPARISONS OF DIFFERENT SURVEILLANCE SYSTEMS

Surveillance Systems	Traditional Systems [1][2]	Systems with a Mobile Viewer [3]	Systems with Network Camera and a Mobile Viewer [4][5]	Our Recognition Assisted Dynamic Surveillance System
Camera	Fixed	Fixed	Movable	Movable
Viewer/Commander	Fixed	Movable	Movable	Movable
Access Network	Fixed Networks	Fixed/Wireless Networks	Wireless Networks	Wireless Networks
Mobility	Low	Medium	High	High
Liveness Quality of Capturing Events	Motion Video	Motion JPEG	Snapshotted Image	Motion JPEG
Service Extensibility	Low	Low	Low	High

## VI. CONCLUSION

The recognition assisted dynamic surveillance system in this paper provides an integrated solution of surveillance service. In our development, OSGi is chosen as the central service platform because of its future extensibility and application interoperability while OpenCV is chosen as the development assistance mechanism because of its mature APIs about the image processing. The integrated system can raise the mobility at two ends – a mobile user end to monitor and control the capturing and a mobile camera end to the event capturing flexibility. Besides, the recognition function can assist in visualize the captured events in a limit screen on a mobile phone more clearly. We develop this system by an OSGi service platform, an OpenCV development platform, a camera mounted on an embedded system carried by a robot, and a J2ME based viewer and commander program on a mobile phone to realize this concept. In this article, we show the design skeleton of the integrated system and evaluate the service performance while accessing the dynamic surveillance service via different wireless networks.

## REFERENCES

- [1] G.L. Foresti, "A real-time system for video surveillance of unattended outdoor environments," *Circuits and Systems for Video Technology*, IEEE Transactions on, vol. 8, no. 6, 1998, pp. 697-704.
- [2] M. Esteve, et al., "A video streaming application for urban traffic management," *Journal of Network and Computer Applications*, vol. 30, no. 2, 2007, pp. 479-498.
- [3] Y. Imai, et al., "Development and a brief evaluation of a web-based surveillance system for cellular phones and other mobile computing clients," *Proc. Human System Interactions*, 2008 Conference on, 2008, pp. 526-531.
- [4] T. Yu-Chee, et al., "iMouse: An Integrated Mobile Surveillance and Wireless Sensor System," *Computer*, vol. 40, no. 6, 2007, pp. 60-66.
- [5] P. Jung-Hyun and S. Kwee Bo, "A design of mobile robot based on Network Camera and sound source localization for intelligent surveillance system," *Proc. Control, Automation and Systems*, 2008. ICCAS 2008. International Conference on, 2008, pp. 674-678.
- [6] "OSGi," <http://www.osgi.org/>.
- [7] "OpenCV," <http://opencv.willowgarage.com/wiki/>.
- [8] "Lego MindStorms NXT," <http://mindstorms.lego.com/>.
- [9] "Java 2 Platform, Mobile Edition," <http://java.sun.com/javame/index.jsp>.
- [10] D. Marples and P. Kriens, "The Open Services Gateway Initiative: an introductory overview," *Communications Magazine*, IEEE, vol. 39, no. 12, 2001, pp. 110-114.
- [11] L. Choonhwa, et al., "Enabling smart spaces with OSGi," *Pervasive Computing*, IEEE, vol. 2, no. 3, 2003, pp. 89-94.
- [12] T. Gu, et al., "Toward an OSGi-based infrastructure for context-aware applications," *Pervasive Computing*, IEEE, vol. 3, no. 4, 2004, pp. 66-74.
- [13] D. Valtchev and I. Frankov, "Service gateway architecture for a smart home," *Communications Magazine*, IEEE, vol. 40, no. 4, 2002, pp. 126-132.
- [14] W. Chao-Lin, et al., "Service-Oriented Smart-Home Architecture Based on OSGi and Mobile-Agent Technology," *Systems, Man, and Cybernetics, Part C: Applications and Reviews*, IEEE Transactions on, vol. 37, no. 2, 2007, pp. 193-205.
- [15] J.E. Loez de Vergara, et al., "An autonomic approach to offer services in OSGi-based home gateways," *Computer Communications*, vol. 31, no. 13, 2008, pp. 3049-3058.
- [16] L. Cheng-Liang, et al., "A wrapper and broker model for collaboration between a set-top box and home service gateway," *Consumer Electronics*, IEEE Transactions on, vol. 54, no. 3, 2008, pp. 1123-1129.
- [17] L. Yuantao, et al., "OSGi-based service gateway architecture for intelligent automobiles," *Proc. Intelligent Vehicles Symposium*, 2005. Proceedings. IEEE, 2005, pp. 861-865.
- [18] C. Ing-Yi and T. Chen-Hsin, "Pervasive Digital Monitoring and Transmission of Pre-Care Patient Biostatics with an OSGi, MOM and SOA Based Remote Health Care System," *Proc. Pervasive Computing and Communications*, 2008. PerCom 2008. Sixth Annual IEEE International Conference on, 2008, pp. 704-709.
- [19] B.S. Heck, et al., "A LEGO experiment for embedded control system design," *Control Systems Magazine*, IEEE, vol. 24, no. 5, 2004, pp. 61-64.
- [20] S.H. Kim and J.W. Jeon, "Introduction for Freshmen to Embedded Systems Using LEGO Mindstorms," *Education*, IEEE Transactions on, vol. 52, no. 1, 2009, pp. 99-108.
- [21] K. Read and F. Maurer, "Developing mobile wireless applications," *Internet Computing*, IEEE, vol. 7, no. 1, 2003, pp. 81-86.
- [22] D.S. Kochnev and A.A. Terekhov, "Surviving Java for mobiles," *Pervasive Computing*, IEEE, vol. 2, no. 2, 2003, pp. 90-95.
- [23] L. Xu, et al., "A J2ME-Based Wireless Intelligent Video Surveillance System Using Moving Object Recognition Technology," *Proc. Image and Signal Processing*, 2008. CISP '08. Congress on, 2008, pp. 281-285.
- [24] "RXTX," <http://www.rxtx.org/>.