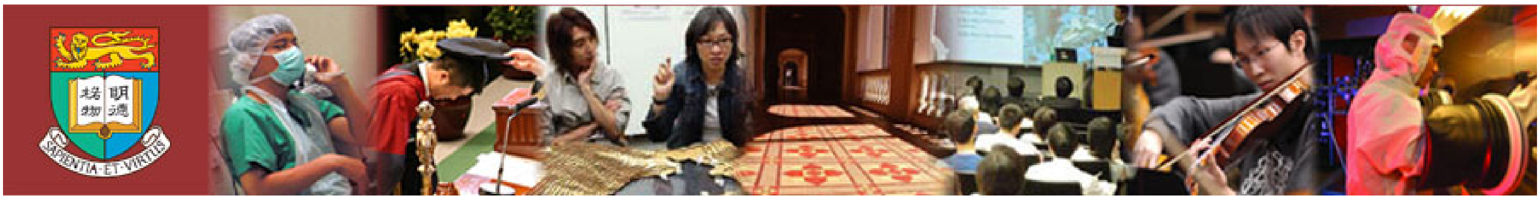


ON THE DESIGN AND IMPLEMENTATION OF A HIGH DEFINITION MULTIVIEW INTELLIGENT VIDEO SURVEILLANCE SYSTEM

S. Zhang[1], S. C. Chan[1], R. D. Qiu[2], K. T. Ng[1], Y. S. Hung[1] and W. Lu[2]

[1]Department of Electrical and Electronic Engineering, The University of Hong Kong
{*szjeff; scchan; ktng; yshung*}*@eee.hku.hk*

[2]School of Electronic and information Engineering, Tianjin University
qrundong@gmail.com, *luwei@tju.edu.cn*

ABSTRACT

This paper proposes a distributed architecture for high definition (HD) multi-view video surveillance system. It adopts a modular design where multiple intelligent Internet Protocol (IP)-based video surveillance cameras are connected to a local video server. Each server is equipped with storage and optional graphics processing units (GPUs) for supporting high-level video analytics and processing algorithms such as real-time decoding and tracking for the video captured. The servers are connected to the IP network for supporting distributed processing and remote data access. The DSP-based surveillance camera is equipped with real-time algorithms for streaming compressed videos to the server and performing simple video analytics functions. We also developed video analytics algorithms for security monitoring. Both publicly available data set and real video data that are captured under indoor and outdoor scenarios are used to validate our algorithms. Experimental results show that our distributed system can support real-time video applications with high definition resolution.

Index Terms— Intelligent video surveillance, GPU, IP camera, video analytics, object tracking, AVS

1. INTRODUCTION

Intelligent video surveillance is an important topic in video processing and computer vision because of its potential and increasing applications in military, commercial and social security. A survey of early works in intelligent distributed surveillance systems can be found in [1]. The technological evolution of video surveillance systems started with conventional analogue CCTV systems [2], which consist of a number of cameras, usually CCD sensors, located at different locations and are connected to certain control rooms for human monitoring or recording. However, these analogue systems are limited by relatively low video resolution and the flexibility in enforcing data security, storage and retrieval, interfacing, video analytics and other automated processing. On the contrary, in Internet Protocol (IP)-based video surveillance systems [3], IP cameras can be connected to different computing devices, or even mobile devices, through the IP network. Therefore, complicated video camera networks can be conveniently realized. Different from analogue systems, digital representation and transmission over IP-based network offer improved data security through encryption and authentication methods, which are highly desirable in social security applications. Though the flexibility of IP-camera based systems makes it easier to integrate more cameras, the computational requirement of various video analytics tasks will also increase dramatically with the number of cameras. In addition, the trend for high-definition (HD) videos at up 720p or 1080p will further increase the computational complexity.

In this paper, we propose a distributed intelligent video surveillance system supporting HD, multi-view processing and real-time video analytics. It adopts a modular design where multiple intelligent IP-based video surveillance cameras are connected to a local video server. Each server is equipped with storage and optional graphics processing units (GPUs) for supporting high-level video analytics and processing algorithms such as real-time decoding and object tracking for the video captured. The servers are connected to the IP network for supporting distributed processing and remote data access. The DSP-based surveillance camera is equipped with real-time algorithms for streaming compressed videos to the server and performing simple video analytics functions. We also developed video analytics algorithms for security monitoring. Both publicly available data set and real video data that are captured under indoor and outdoor scenarios are used to validate our algorithms.

The rest of the paper is organized as follows. Section 2 outlines the system architecture and system design of the proposed high resolution multi-view video surveillance system. Section 3 summarizes the video analytics functions developed and their GPU implementation. In section 4, experimental results are shown to illustrate the effectiveness

978-1-4673-2193-8/12/$31.00 ©2012 IEEE

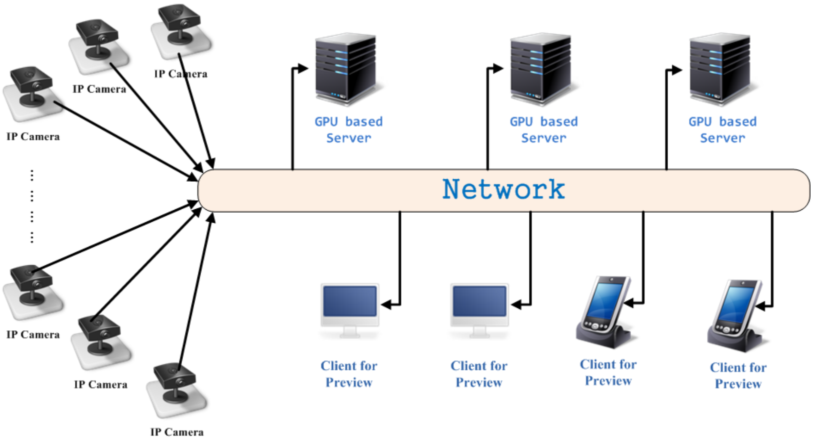


Figure 1: Block diagram of the proposed high definition multi-view intelligent video surveillance system.

of the proposed system. Conclusion and future works are presented in section 5.

2. GPU AND IP BASED VIDEO SURVEILLANCE SYSTEM ARCHITECTURE AND DESIGN

Figure 1 shows the block diagram of the intelligent video surveillance system that we have developed. It adopts a modular design where multiple intelligent IP-based video surveillance cameras are connected to a local video server. Each server is equipped with a storage and optional GPU(s) for supporting high-level video analytics and other processing algorithms for the video captured. The servers are connected to the IP network so as to support remote access from handheld devices, video database, etc., in future development. The IP-cameras are equipped with real-time compression algorithms for streaming compressed video to the server and advanced digital signal processor (DSP) for performing simple video analytics functions.

We have built a prototype video surveillance system based on the concept of Figure 1. The TI-DM368 [4] cameras with 30 frames/s and TI-DM8127 [5] with 60 frames/s cameras are tested in the proposed system. Both of them support real-time compression and streaming of 720p high resolution videos through the network to the video server through RTSP protocol. The video server receives data from the remote IP cameras, decodes and displays various videos for monitoring, and records the video data into local storage.

The display and storage modules of the video servers were developed using DirectShow Software Development Kit (SDK). The Microsoft DirectShow application programming interface (API) is a media-streaming architecture for Microsoft Windows. Using DirectShow, our application can perform high-quality video and audio playback and capture. Figure 2 shows the filter graph of the monitoring software that we have developed. The video server can receive video data from the remote IP cameras and display in different windows for monitoring. Currently nine video windows are supported. Administrator(s) of the video server can select which cameras to display and also can access the recorded videos by selecting the camera and time slots required. Figure 3 shows the graphic user interface (GUI) of the software developed. It can receive and display nine different 720p video streams simultaneously. The GUI can also individually control each specific video stream of the IP cameras, such as start, stop and pause. The video server supports two formats for recording the video data into files: 1) raw data format; and 2) compressed video streams. As the video compression is performed at the TI IP cameras, it supports a wide range of compression formats such as the H.264 standard. On the other hand, as the DSP-based IP camera is programmable, other video compression standard such as the Audio Video Coding Standard (AVS) [11] can be implemented in the more powerful TI DM-8127 camera. We have developed a software-only AVS encoder at 720p resolution on an Intel i7 3.30 GHZ PC without GPU acceleration. It is expected that this codec can be ported readily to GPU to achieve 1080p resolution and the DSP for low cost implementation. In our system, the compressed video data streams are saved as multiple data files, say one minute each, which will facilitate the access of the video data from the surveillance servers.

As mentioned, the video server is optionally equipped with GPU(s), which can support real-time decoding and more video analytics functions. It can also provide the required processing for existing cameras without video analytics functions. The video analytics functions that we have developed in our GPU local server will be described below.

3. VIDEO ANALYTICS ALGORITHMS FOR GPU-BASED VIDEO SURVEILLANCE SERVER

Our video analytics algorithm mainly addresses real-time high-definition observation of moving objects (such as humans and vehicles) in various environments, leading to a description about the activities of the objects in the

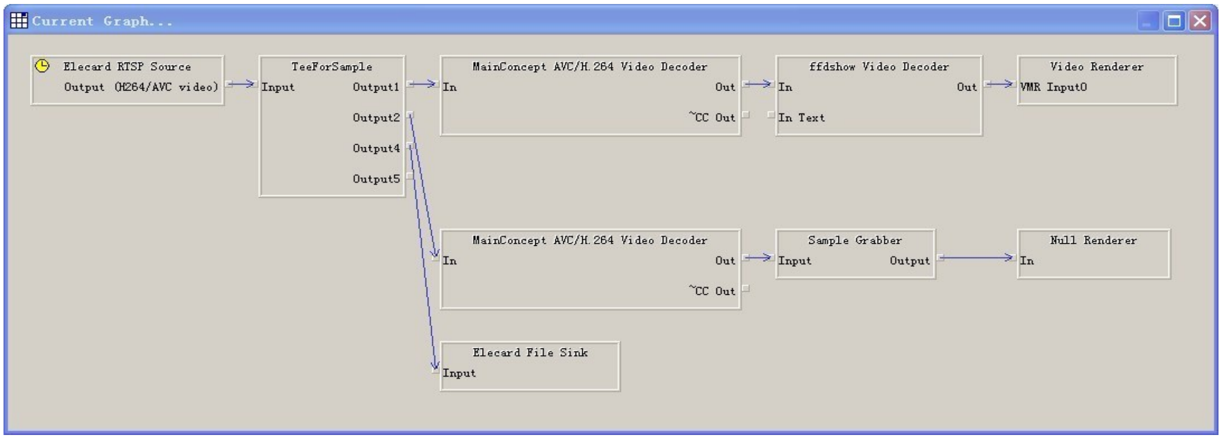
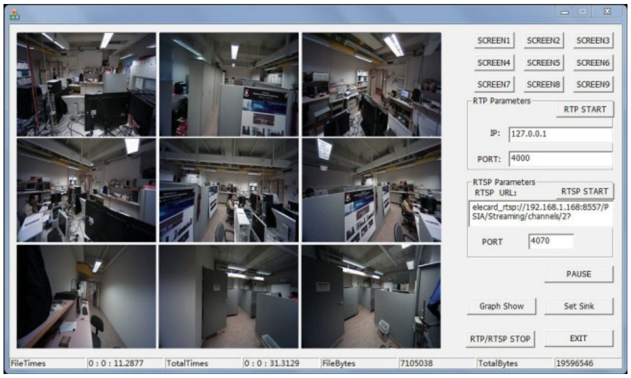


Figure 2: The Filter Graph of the surveillance video server.



IP: 127.0.0.1

PORT: 4000

RTSP URL: elecard_rtsp://192.168.1.168:8557/PSIA/Streaming/channels/2?

PORT 4070

FileTimes 0 : 0 : 11.2877 TotalTimes 0 : 0 : 31.3129 FileBytes 7105038 TotalBytes 19596546

Figure 3: The user interface of the software developed for the real-time monitoring among different video streams from remote IP cameras.

environment or among the objects. It can be used mostly for security monitoring, as well as for traffic flow measurement, accident detection on highways, speeding detection, etc. Figure 4 shows the multi-view video analytics algorithm pipeline.

3.1 Single view video analytics modules

The video analytics system that we have developed on our GPU server includes four modules, namely: background detection, object detection, object tracking and trajectory generation, as depicted in Figure 4.

3.1.1 Background modeling

Background detection using Gaussian mixture model (GMM) [6] is the first stage in the pipeline. This module performs foreground/background segmentation for each pixel captured by the IP camera. Traditionally, it cannot be implemented in real-time because of the high complexity of the algorithm. The processing time of CPU (Intel i7 3.30 GHZ) on a 768×576 PETS2001 dataset [7] is about 85 ms/frame. In the proposed system, we parallelized the algorithm in the GPU server so as to achieve real-time operation. According to our implementation, it costs merely 31 ms/frame on our GPU server, which is more than 2 times improvement in performance.

3.1.2 Object detection

The object detection module detects new objects and deletes old objects using foreground mask obtained by the background detection module. New objects will be passed to the object tracking module. The object tracking module provides frame-by-frame tracking of the object position and size. We used a hybrid object tracker that consists of two components (connected components (CC) tracker and particle filter (PF) tracker) as described in [8]. If there are no object collisions, we can use the CC tracker to perform object tracking. Otherwise, a particle filter tracker will take over the job because of its capability of handling short-time occlusions.

3.2 Multi-view video analytics setting

Based on the framework of our system, it is possible to perform multi-view tracking in emerging multi-view video surveillance systems. The advantage of our system is that it can track multiple objects across multiple cameras by extending the framework of single view we have described in section 3.1. More precisely, the process can be divided into three steps: I) Determination of the Field of View (FOV) lines of each camera. II) Brightness calibration of neighboring cameras, and III) Consistent labeling across cameras when a new object is detected in one or more views. Figure 4 shows the block diagram of a multi-view video surveillance system where consistent labels of objects discovered in different cameras are identified. Details on calibration of multiple cameras and the strategy of consistent labeling can be found in [9].

As we mentioned in section 3.1.1, even background modeling of single view can hardly be implemented in real-time in HD resolution. Therefore, in order to perform multiple views and multiple objects tracking in real-time, we

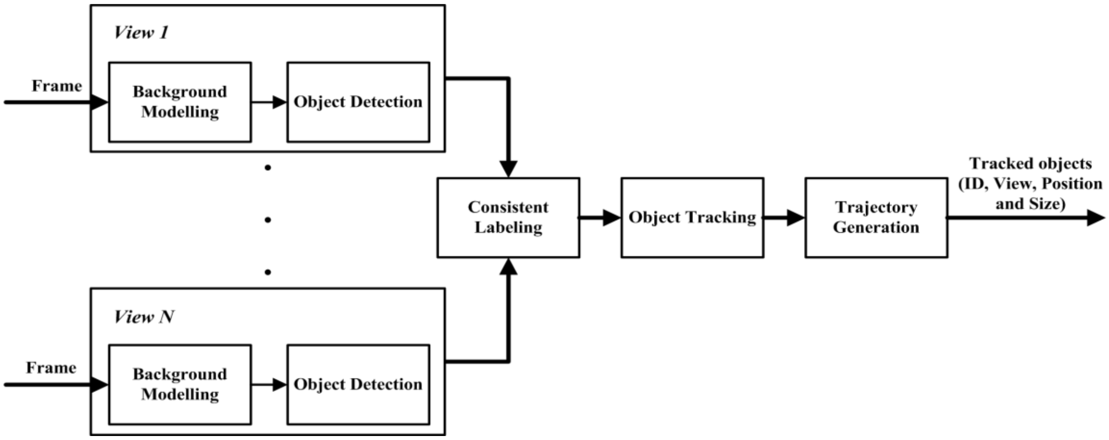


Figure 4: Multi-view video surveillance pipeline.

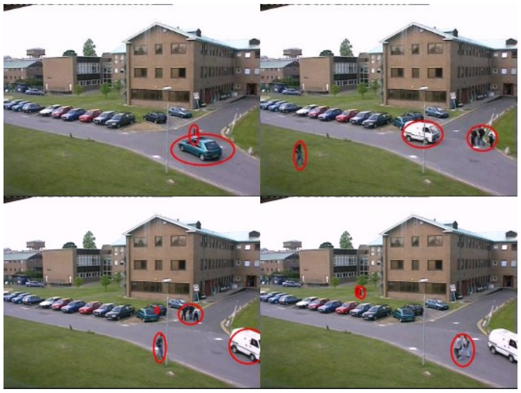


Figure 5: Multiple objects tracking results of PETS2001. The red ellipsoids indicate the tracked object(s).

further integrate our tracking algorithms into the GPU-based video surveillance system framework. The background modeling and object tracking of different cameras are implemented separately by different GPUs. Conventionally, surveillance IP cameras are static, therefore, steps I) and II) in the multi-view video analytics setting only need to be performed at the beginning of the monitoring. Hence, comparing to single view tracking, the work load of multi-view tracking is only increased by the consistent labeling module. Also, in the consistent labeling module, since the global labeling is performed subject to FOV and color similarity constraints, only very little system resources are required and hence and the computing complexity is low.

4. EXPERIMENTAL RESULTS

In order to evaluate the hardware system design and video analytics algorithms, extensive experiments have been carried out. The video analytics algorithms are tested on public datasets PETS2001 and real data that are captured by the IP cameras. All the experiments are performed on two Intel i7 3.30 GHZ-based video servers with 8GB RAM and 3 GTX295 GPUs, individually. TI-DM8127 IP cameras with 60 frames/s and TI-DM368 IP cameras with 30 frames/s are used for capturing. The resolution of the PETS2001 is 768×576 and the resolution of IP camera captured scenarios is 720p. From the results, we find that the system can steadily capture, stream, display and store HD videos. Furthermore, video analytics algorithms can successfully run at 20-25 frame/s on the GPU servers. Figure 5 shows the multiple objects tracking result of the PETS2001 data set. We can see that all moving objects (human and vehicles) are precisely tracked. Figures 6 and 7 show the indoor and outdoor multi-view and multiple objects tracking results. In Figures 6 and 7, consistent labeling is preformed when objects cross the FOV lines of a camera. Therefore, the labels of the tracked objects are successfully maintained when they pass through the FOV lines from one camera to another.

5. CONCLUSION

The design and implementation of a HD multi-view intelligent video surveillance system has been presented. It adopts a modular design where multiple intelligent IP-based surveillance cameras are connected to a local video server. The DSP IP-cameras can provide compression and simple video analytics, while optional. GPU-based video servers can provide further processing such as multi-view and multiple object tracking. The feasibility of this scalable architecture is demonstrated by a real-time prototype system.

Future research will focus on: 1) developing IP cameras with AVS format output and 2) high-level video analytics algorithms such as object recognition, behavior analysis and image-based retrieval. As the complexity of the particle filter (PF) tracker depends highly on the size and sample dimension of the tracking sample, it will occupy a lot of system recourses in real-time implementation. Hence, we have also developed a novel object tracking algorithm called Bayesian Kalman filter (BKF) tracker [10] for video surveillance. It achieves similar performances as the PF tracker but with lower computational complexity. Currently,

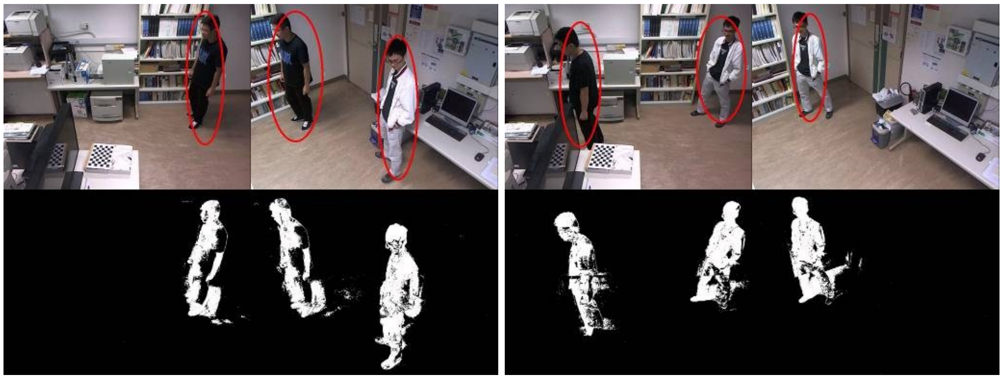

(a) (b)

Figure 6: Two-view indoor tracking results corresponding to frame numbers 550 (a) and 765 (b), each of which includes two adjacent views. The red ellipsoid(s) in the upper figures indicate the tracked object(s), while the lower figures depict the corresponding background subtraction results. Note that the object which appeared in both views is successfully identified as the same object because of the consistent labeling.

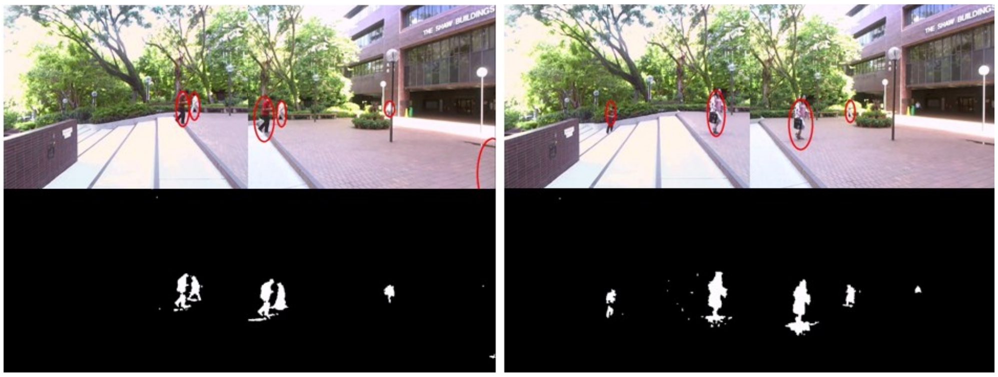

(a) (b)

Figure 7: Two-view outdoor tracking results corresponding to frame numbers 660 (a) and 709 (b), each of which includes two adjacent views. The red ellipsoid(s) in the upper figures indicate the tracked object(s), while the lower figures depict the corresponding background subtraction results. Note that the object which appeared in both views is successfully identified as the same object because of the consistent labeling.

the BKF tracker works under simulation. In the near future, we will parallelize it in the GPU server.

ACKNOWLEDGMENT

This work was supported partly by Hong Kong Research Grant Council (RGC) and the Innovation and Technology Fund (ITF).

REFERENCES

[1] M. Valera and S. A. Velastin, “Intelligent distributed surveillance systems: review,” *IEE Proc. Vision, Image and Signal Process*., vol. 152(2), pp. 192-204, Apr. 2005.

[2] H. Kruegle, *CCTV surveillance: analog and digital video practices and technology*, Butterworth-Heinemann, MA: Burlington, 2006

[3] W. T. Chen, P. Y. Chen, W. S. Lee and C. F. Huang, “Design and implementation of a real time video surveillance system with wireless sensor networks,” in *Proc. IEEE Vehicular Tech. Conf.*, pp. 218-222, May, 2008.

[4] IP camera TI-DM368 [Online]. Available: http://processors.wiki.ti.com/index.php?title=DM368

[5] IP camera TI-DM8127 [Online]. Available: http://www.Approp ho.com/NewWeb/Product_DM8127J3.php

[6] Z. Zivkovic and F. van der Heijden, “Efficient adaptive density estimation per image pixel for the task of background subtraction”, *Pattern Recognition Letters*, vol. 27, pp. 773-780, May 2006.

[7] PETS2001 dataset. [Online]. Available: ftp://pets2001.cs.rdg.ac.uk/

[8] T. P. Chen, et al., “Computer vision workload analysis: case study of video surveillance systems”, *J. Intel Tech*, vol. 9(2), pp. 109-118, May 19, 2005.

[9] L. Z. Zhu, J. N. Hwang and H. Y. Cheng, “Tracking of multiple objects across multiple cameras with overlapping and non-overlapping views,” in *Proc. IEEE intl. Symp. Circuits and Syst.*, pp. 1056-1060, May 24, 2009.

[10] S. C. Chan, B. Liao and K. M. Tsui, “Bayesian Kalman filtering, regularization and compressed sampling”, in *Proc. IEEE Intl. Midwest Symp. On Circuits Syst.*, pp. 1-4, Apr. 2011.

[11] *Information Technology Advanced Audio Video Coding Standard Part 2: Video*, Audio Coding Standard Group of China (AVS), Doc. AVS-N1063, Dec. 2003.