# A HIGH RESOLUTION SMART CAMERA WITH GIGE VISION EXTENSION FOR SURVEILLANCE APPLICATIONS

*E. Norouznezhad, A. Bigdeli, A. Postula and B. C. Lovell*

ITEE, The University of Queensland, Brisbane, QLD 4072, Australia
NICTA, 300 Adelaide Street, Brisbane, QLD 4000, Australia

## ABSTRACT

*Intelligent video surveillance is currently a hot topic in computer vision research. The goal of intelligent video surveillance is to process the captured video from the monitored area, extract specific information and take appropriate action based on that information. Due to the high computational complexity of vision tasks and the real-time nature of these systems, current software-based intelligent video surveillance systems are unable to perform sophisticated operations. Smart cameras are a key component for future intelligent surveillance systems. They use embedded processing to offload computationally intensive vision tasks from the host processing computers and increasingly reduce the required communication bandwidth and data flows over the network. This paper reports on the design of a high resolution smart camera with a GigE Vision extension for automated video surveillance systems. The features of the new camera interface standard, GigE Vision will be introduced and its suitability for video surveillance systems will be described. The surveillance framework for which the GigE Vision standard has been developed is presented as well as a brief overview of the proposed smart camera.*

***Index Terms—*** Intelligent Video Surveillance, Smart Camera, GigE Vision

## 1. INTRODUCTION

Video surveillance systems are becoming increasingly important in public access facilities such as airports, transport stations and banks to provide security. The fundamental problem of traditional video surveillance systems is that they are often monitored by human operators. While mounting more video cameras is relatively cheap, finding and funding human resources to observe the video feeds is very expensive. Moreover, human operators performing the surveillance monitoring rapidly become tired and inattentive due to the dull and boring nature of the activity [1, 2]. Therefore there is strong interest in reducing the role of humans in video surveillance systems and using humans only when it is required. Intelligent video surveillance systems enhance efficiency and security levels by means of using machines instead of humans to monitor the surveillance areas. The goal of intelligent video surveillance systems is to analyse the captured video by machine, extract specific information and take appropriate action based on that information [3, 4].

Currently intelligent video surveillance systems use traditional cameras. In these systems, the video streams from all the cameras are directed to the central processing units and the central processing units should process all the received video. Therefore the whole processing load is borne by the host computers. There are a number of factors which limit the efficiency of current software-based video surveillance systems. The first reason is the fact that computer vision tasks are inherently computationally intensive. Performing computer vision tasks on a number of video streams requires high computing power. Making decisions in real time will further constrain the processing that can be applied. That is the primary reason why current software based automated video surveillance systems can only perform simple tasks, such as lane control at train stations. Even with recent outstanding advances in microelectronics, standard PC architectures are frequently unable to deliver the required performance for such applications.

Moreover with the emergence of high resolution image sensors, video transmission from tens to hundreds of cameras to the central processing workstation requires high bandwidth communication networks. It is predicted that future intelligent video surveillance systems will need even more computing power and higher communication bandwidth. This is of course due to not only higher resolution image sensor and higher frame rates but also increasing number of cameras in video surveillance networks. Therefore novel solutions are required in order to address stringent constraints on video surveillance systems, both in terms of communication bandwidth as well as computing power.

978-1-4244-2665-2x/08/$25.00 ©2008 Crown

Embedded processing can overcome the aforementioned constraints by executing low-level tasks within the camera platform, before data transmission to the host system. Cameras with embedded processing resources are known as "Smart Cameras", and are the subject of growing interest [5]. We propose a high resolution smart camera with the GigE vision interface to be used in intelligent video surveillance systems. This paper addresses the brief overview of the proposed smart camera architecture, its communication interface and the surveillance framework in which this smart camera will be utilized. The rest of this paper is organized as follows. The brief overview of smart cameras and their advantages toward automated video surveillance systems is presented in section 2. The features of new camera interface – GigE Vision – and its suitability for video surveillance networks will be discussed in section 3. Section 4 describes the proposed surveillance framework. Finally, the system architecture of the smart camera will be described in section 5, followed by some conclusions in section 6.

## 2. SMART CAMERA

### 2.1. Smart Camera Overview

Smart cameras are becoming increasingly popular with recent advances in both machine vision and semiconductor technology [6-8]. The concept of smart camera is to embed processing core into the camera architecture itself, and process the images inside the camera where the image quality is best.

### 2.2. Smart Camera for Surveillance systems

Computing power and communication bandwidth were considered as the main challenges toward intelligent video surveillance systems in previous section. Using smart cameras instead of typical surveillance cameras can drastically decrease the required computing power of central processing units and network communication bandwidth [9, 10]. In terms of computing power, using smart cameras reduces the processing load of the central processing units by means of execution of low-level image processing tasks within the camera platform, before data transmission to the host system. This way, amounts of data to be transmitted is radically reduced since instead of sending the whole image information contents to the host system, it only sends the specific information. Furthermore transmitted data is more pertinent than the raw pixel flow, meaning that received data can be promptly used by the central processing units, without pre-processing time consuming tasks.

For instance, to perform face recognition in an intelligent video surveillance system, the smart camera will detect the faces, extract the coordinates of the faces in the image and only send the faces data to the host processing units. So that, the host computer only needs to perform face recognition task using the provided face information.

### 2.3. Smart Camera Building Blocks

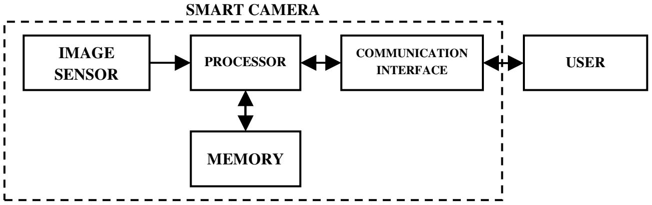

Figure 1: Basic Smart Camera Architecture

In figure 1, a basic structure of a smart camera is presented. Image sensor, processor, memory and communication interface are the four major parts of each smart camera. The image sensor in every imaging system is the first stage of the system. The two current available technologies are Charged Coupled Device (CCD) and CMOS sensors. Due to the high resolution, low noise, low power and high speed nature of CMOS image sensors, it is expected in the future that CMOS based image sensors will outperform CCD based image sensors [11].

Four different platforms can be considered as smart camera processor for implementation of algorithms, Application Specific Integrated Circuit (ASIC), General Purpose processor (GPP), Digital Signal Processors (DSP) and Field Programmable Gate Arrays (FPGA). Due to the parallel nature of FPGAs, they are becoming increasingly attractive for image processing and computer vision tasks, as FPGAs can exploit the inherently parallel nature of many vision problems [12-16]. James et al. described the advantageous and disadvantages of using FPGAs for image processing tasks in detail in [17].

Communication interface is the third major component in the smart camera architecture. Different types of camera interface and GigE Vision in particular will be described in the following section.

## 3. GIGE VISION

### 3.1. Camera Interfaces

Currently, there are five commonly used high bandwidth camera interface standards. Camera Link, Gigabit Ethernet (GigE), USB2.0, Firewire 400 or IEEE 1394a and Firewire 800 or IEEE 1394b. Table 1 shows the comparison of these standards. With the emergence of new high resolution CMOS image sensor technology in machine vision industry, camera networks require much more bandwidth than what is supported by USB2.0 and Firewire400. While Cameralink supports ultra high bandwidth data transfer, it cannot be used in video surveillance networks, as it only supports point-to-point connections. Thus GigE Vision and FireWire 800 interfaces are considered as the most suitable interfaces for future video surveillance networks utilizing high resolution image sensors. Due to several reasons which will be discussed later in this section, we believe that GigE Vision will be the dominant camera interface in future surveillance networks.

Table 1: Specification of Camera Interfaces

| Criteria | GigE Vision | Firewire 800 | USB 2.0 | Camera Link |
|---|---|---|---|---|
| Connection Type | Point to Point | Peer to Peer | Master-Slave | Point To Point |
| Bandwidth | <1000 Mpbs | < 800 Mpbs | <480 Mpbs | 2,380<br>4760<br>7,140<br>Mpbs |
| Distance | < 100m | < 4.5 m | < 5m | < 10m |
| Wireless Support | Yes | No | No | No |
| Max # of Cameras | Unlimited | 63 | 127 | 1 |

### 3.2. GigE Vision Overview

The GigE Vision is a new standard developed by a committee of the Automated Imaging Association (AIA), for high performance machine vision cameras [18-20]. The GigE vision standard includes the hardware interface standard (Gigabit Ethernet), communication protocols, and standardized camera control registers which are based on a command structure called GenIcam [21]. GenIcam which has been developed by European Machine Vision Association (EMVA) seeks to provide a generic camera description file for all camera types, regardless of the interface technology they use (i.e. GigE, Firewire, Camera Link, etc.).

### 3.3. GigE Vision Protocol Architecture

The best way to describe GigE Vision is using the Open Systems Interconnection (OSI) Reference Model. Figure 2 depicts the GigE Vision layers in comparison to TCP/IP according to the OSI model. To support high bandwidth data transfer, the GigE Vision standard is based on the User Datagram Protocol (UDP). Instead of establishing a host-to-host connection as with TCP, UDP uses ports to allow application-to-application connections. While this makes UDP less reliable than TCP, it increases high speed image transfer which is what is really required for machine vision applications.

To overcome the unreliability of UDP, some extra protocols have been added to GigE Vision. These two protocols introduced by the GigE Vision standards committee are GigE the Vision Control Protocol (GVCP) and the GigE Vision Streaming protocol (GVSP). The GVCP is an application layer protocol which runs on top of UDP IPv4. It defines how to control and configure compliant devices (such as cameras), specifies stream channels, and provides mechanisms for cameras to send image and control data to the central processing units. The main task of GVCP is to add some mechanisms to UDP to guarantee the reliability of image transmission. The GVSP is another application layer protocol that allows an application to receive image data, image information, and other information from a device. GVSP provides mechanisms to guarantee the reliability of packet transmission (through GVCP) and to minimize the flow control required due to the unreliability of UDP [18].

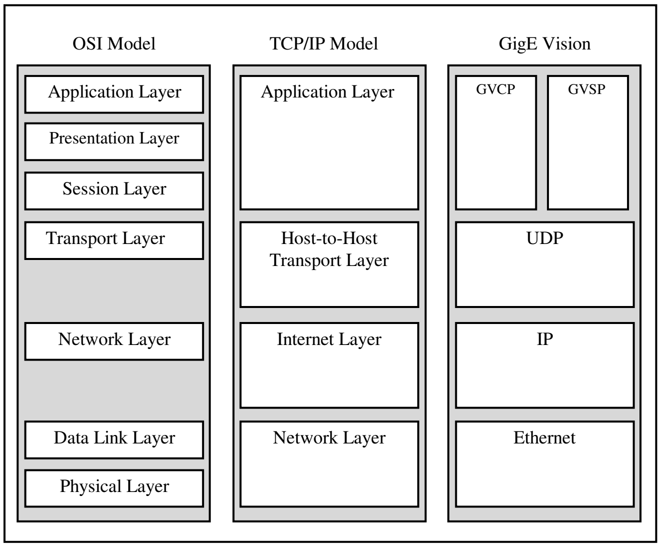

Figure 2: Structure of GigE Vision Protocol

### 3.4. GigE Vision for Video Surveillance Networks

GigE Vision offers many features which make it quite suitable for a networking interface in distributed video surveillance systems. One of the biggest costs in deploying a surveillance system is the infrastructure (e.g., cabling) required to transport the video from the cameras to a central location where it can be analyzed and stored [3]. Since the GigE Vision standard builds on Ethernet technology, standard Ethernet hardware, architecture, and network structure is used with any system that incorporates GigE Vision. The wide spread usage of the Gigabit Ethernet on computers and the compatibility of GigE Vision with the full range of standard Gigabit Ethernet devices such as hubs, switches and routers, allow networking of cameras with existing structured wiring for data and telephony at no extra cost. Using Power over Ethernet (PoE) to provide camera power further reduces costs and simplifies installation.

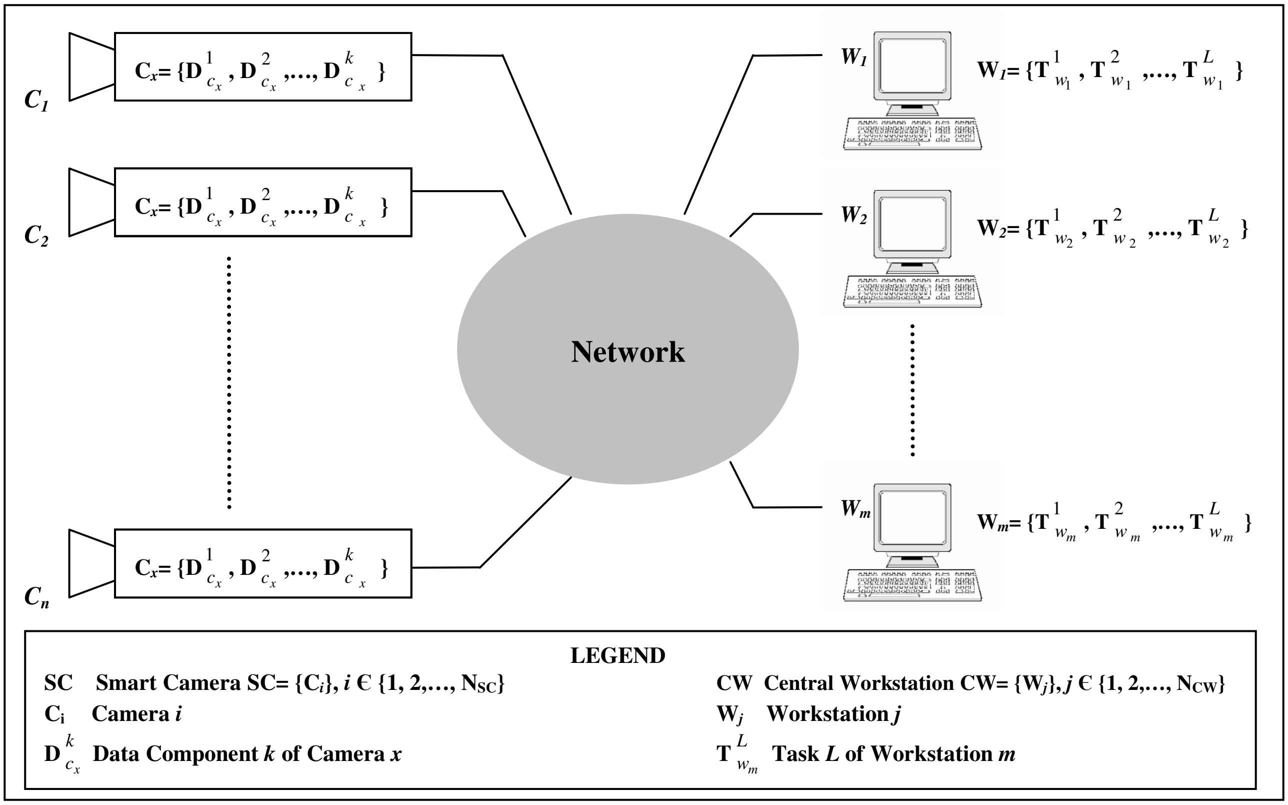
$C_1$
$C_x$= {$D^1_{c_x}$, $D^2_{c_x}$,…, $D^k_{c_x}$ }
$C_2$
$C_x$= {$D^1_{c_x}$, $D^2_{c_x}$,…, $D^k_{c_x}$ }
$C_n$
$C_x$= {$D^1_{c_x}$, $D^2_{c_x}$,…, $D^k_{c_x}$ }
Network
$W_1$
$W_1$= {$T^1_{w_1}$, $T^2_{w_1}$,…, $T^L_{w_1}$ }
$W_2$
$W_2$= {$T^1_{w_2}$, $T^2_{w_2}$,…, $T^L_{w_2}$ }
$W_m$
$W_m$= {$T^1_{w_m}$, $T^2_{w_m}$,…, $T^L_{w_m}$ }
LEGEND
SC Smart Camera SC= {$C_i$}, $i$ Є {1, 2,…, $N_{SC}$}
CW Central Workstation CW= {$W_j$}, $j$ Є {1, 2,…, $N_{CW}$}
$C_i$ Camera $i$
$W_j$ Workstation $j$
$D^k_{c_x}$ Data Component $k$ of Camera $x$
$T^L_{w_m}$ Task $L$ of Workstation $m$

Figure 3: Proposed Surveillance Framework

The wiring capacity is not easily scalable and because of need for manual labor, instantaneous/portable installation of network is difficult [3]. Therefore there is a growing trend towards using wireless cameras in surveillance networks due to its ease of use and installation. With the advent of Wireless Gigabit Ethernet, a wireless medium can be used as a physical layer in GigE Vision networks.

Gigabit Ethernet networks have the potential for even higher bandwidths with 10 Gigabit Ethernet projected as an option for the future. Therefore GigE Vision provides enough bandwidth to transmit video in real time which is an important issue for intelligent video surveillance systems.

Due to the reasons outlined in this section, it is the authors' believe that GigE vision will be the dominant camera interface for future intelligent video surveillance systems. This can be emphasized in the context of our proposed surveillance framework which is described in the following section.

## 4. PROPOSED SURVEILLANCE SYSTEM

### 4.1. Proposed Surveillance Framework

The proposed framework for intelligent video surveillance system consists of three modules: smart cameras, Network, and central processing workstations. In this framework, there are a large number of smart cameras which are connected via a wide-area network to cooperate to carry out a number of video surveillance tasks, with each performing its own video analysis, extracting specific information and exchanging high-value, low-bandwidth information over a network to the central processing workstations. The wide-area network consisting of networking devices such as switches, routers and servers, is responsible for distributing the data streams among the processing nodes. Central processing workstations use the generated data streams by smart cameras to perform video surveillance tasks.

The proposed system consists of a set of modules: Smart Cameras SC= {$C_i$}, $i$ Є {1,2,…,$N_{SC}$}, where $N_{SC}$ is the number of smart cameras, Central workstations CW={$W_j$}, $j$ Є {1,2,…, $N_{CW}$}, where $N_{CW}$ is the number of central processing workstations. Each module is built from a set of components, which fall under two main categories. The first components, denoted by $D^k_{c_x}$ are the data components which are generated by the smart camera. For instance, the module $C_1$ consists of $n$ components, labelled $C_1$= {$D^1_{c_1}$, $D^2_{c_1}$,…, $D^k_{c_1}$}. In the proposed system, $D^k_{c_x}$ denotes $k$th data component generated by the $x$th smart camera. The second components denoted by $T$ are the task components which are executed on the central processing workstations. A module $W_l$ consists of $m$ components, labelled $W_l$= {$T^{1}_{w_1}$ , $T^{2}_{w_1}$ ,…, $T^{L}_{w_1}$ }. $T^{L}_{w_y}$ denotes the $L^{th}$ Task which will be executed on the $y^{th}$ central workstation. The main purpose of the proposed framework is that each of the video surveillance tasks ($T^{L}_{w_y}$ ) which executes on central processing units should have access to each of the generated data ($D^{k}_{c_x}$ ) components by the smart cameras. Using this mathematical model, we developed an application layer for the smart camera's communication interface. The network topology and system architecture of the central processing workstations is beyond the scope of this paper.

### 4.2. Communication Interface Architecture

We have developed a GigE Vision protocol for the proposed intelligent video surveillance system. The GigE Vision User Protocol (GVUP) is added to the GigE Vision's application protocol to meet the goals outlined in previous sections.

As described in previous section, GigE Vision uses UDP standard as a transport layer. The UDP Packet structure is depicted in figure 4. Each data packet consists of source and destination IP address and port numbers, beside the other fields of data. The added module to the application layer, provides this possibility to generate data packets with different IP and port numbers for different data components of smart cameras. So that each of the tasks of different workstations ($T^{L}_{w_y}$ ) can choose any of the data components generated by the smart cameras ($D^{k}_{c_x}$ ).

| | Bits 0-7 | 8-15 | 16-23 | 24-31 |
|---|---|---|---|---|
| 0 | Source Address | | | |
| 32 | Destination Address | | | |
| 64 | Zeros | Protocol | UDP Length | |
| 96 | Source Port | | Destination Port | |
| 128 | Length | | Checksum | |
| 160 | Data | | | |

Figure 4: UDP IPV4 Packet Structure

As a result, the data packets which have different contents will be distributed more intelligently among the network nodes. In this system, instead of sending the entire video data streams from all cameras to all host processing units, specific parts of the video content of each camera will be sent to the specific host processing unit.

Also at the destination node, i.e. central processing workstations, the different components of the received data are distinguishable**,** so that each host processing units can select different data components which are useful for its particular tasks.

The full description of the GVUP module is beyond the scope of this paper and will be presented in a separate publication currently under preparation.

## 5. SMART CAMERA ARCHITECTURE

### 5.1. System Overview

A block diagram of our smart camera is depicted in figure 5. The sensing unit is composed of the CMOS image sensor with an interface. As this project's goal is to have the whole processing system on a chip, all the processing units comprising the Pixel-based processing unit, ROI processing unit, Network processor, Memory controller and a Microblaze processor are integrated on a single FPGA chip. The communication interface is composed of Gigabit Ethernet (GigE) and an optional Firewire interface. Also an external memory is used to store the video content of the image sensor and interim results of processing elements. The brief overview of all of these units is given in this section. For full details of the NICTA smart camera specification and internal architecture, please refer to [6].

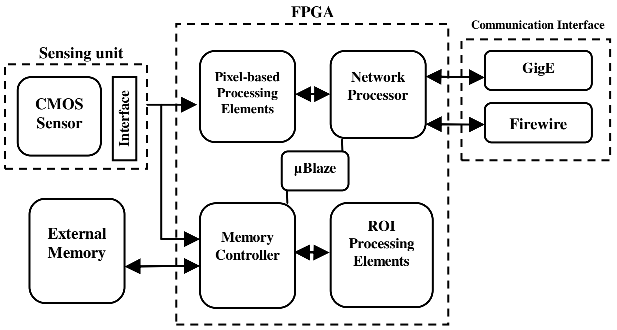

Figure 5: System Architecture

### 5.2 Hardware Platform

Choosing the suitable target hardware for a smart camera processor is an important issue. Due to the parallel nature of FPGAs, they are becoming increasingly attractive for image processing and computer vision tasks, as FPGAs can exploit the inherently parallel nature of many vision problems [12-16]. Beside the parallel nature of vision tasks, the proposed smart camera should be able to perform the processing tasks concurrently between two successive frames. Due to the above mentioned reasons, FPGA has been chosen as a target platform of our smart camera. Among available FPGA families available in the market, a XC3S5000 from Xilinx

Spartn-3 family has been chosen as the main processing unit of our smart camera due to its low price and high volume of logic gates. The price is a factor of concern because the product should be able to compete with existing surveillance cameras.

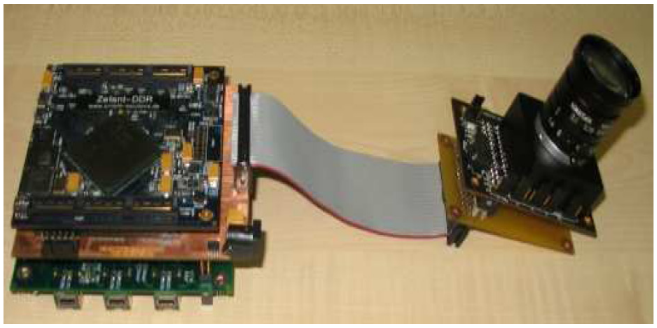

Figure 6: Smart Camera Prototype

### 5.3 Image sensor

CMOS image sensor is chosen for the smart camera due to the aforementioned reasons in section 2.3. There are many CMOS image sensors in the market. Frame rate and image resolution are the two main factors for choosing the image sensor. Most of image sensors which are used by research groups who are active in the field of smart camera design serve low resolution. As our target application is intelligent crowd surveillance, we chose five mega pixel CMOS image sensor, because higher image resolution can provide much more detailed information regarding the objects of interest which is a key factor in any machine vision applications. The specification of a utilized image sensor is given in table 2. It can operate up to 14 FPS at full resolution [22].

Table 2:Image Sensor Specification

| | |
|---|---|
| Manufacturer | Micron |
| Number | MT9P001 |
| Sensor | CMOS |
| Colour Filter array | RGB Bayern Pattern |
| Resolution | 2592 x 1944 |
| Max Frame Rate | 14 FPS |
| ADC - Resolution | 12 bit |

### 5.4. Processing Unit

The Processing unit of the smart camera is divided into two major parts, Pixel-based processing unit and ROI processing unit. There are three main data flows for the generated raw video data by image sensor. Besides sending the raw video data to the memory controller to be stored in the external memory of the system, they are passed on to Pixel-based processing and ROI processing unit via the memory controller to be processed by these units simultaneously. Each of these processing units is composed of several processing elements which can execute in parallel with each other and perform specific task on incoming video data. The main difference between ROI processing elements and Pixel-based processing elements is that pixel-based processing elements can start their process by buffering just a few lines of each frame, while ROI processing elements need the whole data of each frame to detect regions of interest.

The Pixel-based processing unit is composed of a number of blocks called Pixel-based processing Elements (PPE). They perform some low-level image processing tasks to provide different versions of the incoming captured video. Current PPE blocks provide VGA, RGB, Greyscale and binary presentation of captured video. The VGA block, converts the 5 mega pixel images to VGA (640*480) size. The RGB block, converts the Bayern pattern images to RGB 24-bits per pixel images. The greyscale unit provides the greyscale presentation of the colour image and the binary presentation of the images are provided using edge detection techniques. Each of these processing blocks provides data components which are useful for different tasks which run on the central processing workstations.

The ROI processing unit is the heart of the system. It extracts specific data from the captured video by processing the image data in real time. It consists of several ROI processing elements which can execute concurrently to extract different regions of interest of the video. After extracting the required information, they give the coordinates of the objects of interests to the memory management unit and it sends the data of those regions to the network processor to send over the network. It should be noted that the proposed smart camera architecture maintain the resolution of regions of interest. Besides sending each frame in low resolution in VGA size, it sends the regions of interest at their original resolution. Currently Several region of interests (ROI) blocks are under development at NICTA. The main ones are face detection, number plate detection and road-sign detection.

### 5.5 Memory Controller

Memory controller plays an important role in the proposed smart camera architecture. As memory data transmission consumes much more time than the processing units of the systems, robust data flows is designed to synchronize the memory data transmissions with the processing units of the system to meet real-time constraints of the system. A Microblaze processor is utilized in this system for easier communication between memory controller and other parts of the system. Due to the high resolution image sensor which is utilized in the smart camera, high capacity memory is required to store the data streams. So that 1 GB DDR SDRAM memory has been chosen as the data storage of the system.

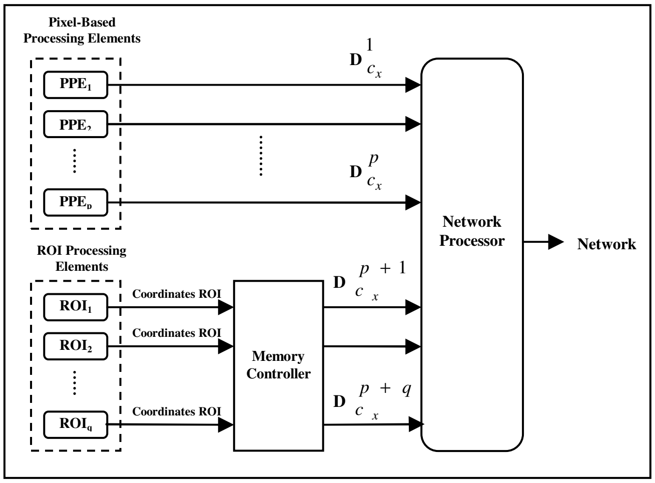

Figure 7: Internal Data Flows

### 5.6 Network processor

The final stage of the smart camera is the network processor unit. As it is shown in figure 6, the network processor should organize the generated data components by Pixel-based processing elements (PPEs) and ROI processing elements, and provide data packets and send them over the network.

While Pixel-based processing elements can start their process by buffering a few lines of each frame, the ROI processing elements need the data of each frame to start processing. So for each frame of captured video, first the data components of Pixel-based processing elements will be sent over the network. Meanwhile the ROI processing elements have enough time to provide the coordinates of regions of interest.

As mentioned in section 4, we use GigE vision interface to send the data to the host processing systems. The GigE Vision IP core is provided by Xilinx which has been developed by the Sensor-to-Image Company. The detailed description of the GigE vision IP core can be found in [23]. The architecture of the GigE Vision IP core and its interfaces to the other parts of the system is depicted in figure 7. The shaded block inside the CPUif is the added GVUP module to the application layer.

## 6. CONLUSION AND FUTURE WORK

Smart cameras can play important role in future intelligent video surveillance systems, increasing significantly the level of intelligence of these systems by means of performing time-consuming low-level image processing tasks inside the camera; prepare useful information for the central processing units to perform high-level vision tasks. This way intelligent video surveillance system will be able to perform more computationally intensive vision tasks. This paper reported on our smart camera architecture for the proposed intelligent video surveillance framework. In the proposed framework, central processing units will only access the data components generated by each of smart cameras which are required in their processing tasks. As a result, due to intelligence of data transmission among the network nodes and pertinence of transmitted data, required computing power and communication bandwidth will be drastically decreased. Our future work would involve adding some adaptive mechanisms to the network interface card so that in case of bandwidth reduction, it can distribute data components more intelligently among the processing nodes.

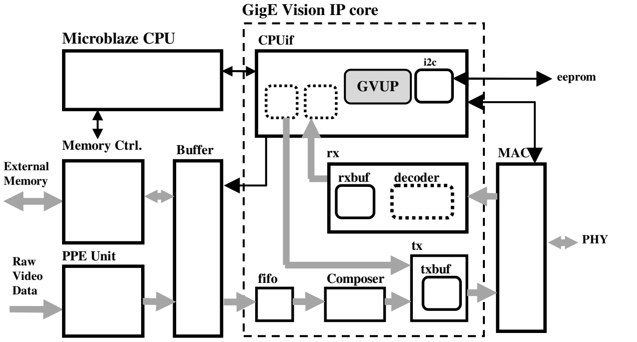

Figure 8: GigE Vision IP Core structure

## 6. ACKNOWLEDGEMENT

All technical discussions in this paper are based on ongoing work in smart camera project at NICTA Queensland Research Laboratory. This project is supported by a grant from the Australian Government Department of the Prime Minister and Cabinet. National ICT Australia is funded by the Australian Government's Backing Australia's Ability initiative and the Queensland Government, in part through the Australian Research Council.

## 8. REFERENCES

[1] A. R. Dick and M. J. Brooks, "Issues in Automated Visual Surveillance", *Proc. of VIIth Digital Image Computing: Techniques and Applications*, Sydney, pp. 195-204, December 2003.

[2] Y. M. Mustafah, A. Bigdeli, A. W. Azman, and B. C. Lovell, "Smart Cameras Enabling Automated Face Recognition in the Crowd for Intelligent Surveillance System", *RNSA Security Technology Conference*, pp. 310-318, September 2007.

[3] A. Hampapur, L. Brown, J. Connell, S. Pankanti, A. Senior and Y. Tian, "Smart surveillance: applications, technologies and implications", IBM T.J. Watson Research centre, www.research.ibm.com/peoplevision/, Mar 2008.

[4] D. Ostheimer, S. Lemay, D. Aayisela, P. F. Dagba, M. Ghazal, and A. Amer, “A Modular distributed video surveillance system over IP”, *Canadian Conference on Electrical and Computer Engineering (CCECE’06)*, Ottawa, pp.518 – 521, May 2006.

[5] F. Dias, F. Berry, J. Serot, and F. V. Marmoiton, “Hardware, Design and Implementation issues on a FPGA-based Smart Camera”, *First ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC '07)*, Vienna, pp. 20-26, September 2007.

[6] Y. Mustafah, A. Azman, A. Bigdeli, and B. C. Lovell, “An Automated Face Recognition System for Intelligence Surveillance: Smart Camera Recognizing Faces In The Crowd”, *First ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC'07)*, Vienna, pp. 147-152, September 2007.

[7] W. Wolf, B. Ozer, and T. Lv, “Smart cameras as embedded systems”, *Computer*, vol.35, pp.48-53, 2002.

[8] M. Kolsch and B. Kiascanin, “Embedded computer vision and smart cameras”, *Tutorial given on Embedded Systems Conference*, Silicon Valley, 2007.

[9 ] V. Nair, P. O. Laprise, and J. J. Clark, “An FPGA-Based People Detection System”, *EURASIP Journal on Applied Signal Processing*, Hindawi Publishing Corporation, pp 1047-1061, 2005.

[10] M. Bramberger, J. Brunner, B. Rinner, and h. Schwabach, “Real-Time video analysis on an embedded smart camera for traffic surveillance”, *Proc. Of the 10th IEEE Real-time and Embedded Technology and Applications Symposium (RTAS’04)*, Toronto, Canada, pp. 174 – 181, May 2004.

[11] Y. Mustafah, T. Shan, A. Azman, A. Bigdeli, and B. C. Lovell, “Real-Time Face Detection and Tracking for High Resolution Smart Camera System”, *Proc. of Digital Image Computing: Techniques and Applications (DICTA'07)*, Adelaide, pp. 387-393, December 2007.

[12] P. Garcia, K. Compton,M. Schulte, E.Blem, and W.Fu, “An Overview of Reconfigurable Hardware in Embedded Systems”, *EURASIP Journal on Embedded Systems*, Hindawi Publishing Corporation, vol. 2006, pp. 1–19, 2006.

[13] S. M. Chai, N. Bellas, G. Kujawa, T. Ziomek, L. Dawson, t. Scamianci, M. Dwyer, and D. Linzmeier, “Reconfigurable streaming architectures for embedded smart cameras”, *Proc. Of the 2006 Conference on computer vision and Pattern Recongnition workshop (CVPR’06)*, New York, NY, pp. 122 – 130, June 2006.

[14] R. Mosqueron, J. Dubois, and M. Paindavoine, “High-Speed Smart Camera with High Resolution”, *EURASIP Journal on Embedded Systems*, Hindawi Publishing Corporation, Vol 2007, pp. 23-38, 2007.

[15] M. Leeser, S. Miller, and H. Yu, “Smart camera based on reconfigurable hardware enables diverse real-time applications”, *Proc. Of the 12th Annual IEEE symposium on field Programmable Custom Computing Machines (FCCM’04)*, Napa, California, pp. 147 – 155, April 2004.

[16] P. Chalimbaud and F. Berry, “Embedded Active Vision System Based on an FPGA Architecture”, *EURASIP Journal on Embedded Systems*, Hindawi Publishing Corporation, Vol 2007, pp. 26-37, 2007.

[17] W. J. Maclean, “An evaluation of suitability of FPGAs for embedded vision systems”, *Proc. Of the 2005 IEEE Computer Society conference on computer Vision and Pattern Recognition (CVPR’05)*, San Diego, CA, pp. 131-137, June 2005.

[18] “The Elements of GigE Vision”, whitepaper, Basler Vision Technologies, http://www.baslerweb.com/

[19] “GigE Vision – CPU load and latency”, whitepaper, Basler Vision Technologies, http://www.baslerweb.com/

[20] “Can GigE Vision deliver on its promise”, Technical white paper, January 2008, http://www.sony-vision.com/

[21] GenICam Standard Specification, Version 1.0, http://www.genicam.org/

[22] Micron 5Mp CMOS Digital Image Sensor MT9P001 Datasheet, Micron Technology, http://www.micron.com/

[23] GigE Vision IP Specification, Revision 1.0.7, http://www.xilinx.com/