Approval of the Department of Electronic & Telecommunication Engineering

.....
Head, Department of Electronic &
Telecommunication Engineering

This is to certify that I/we have read this project and that in my/our opinion it is fully adequate, in scope and quality, as an Undergraduate Graduation Project.

Supervisors:

Dr. Ranga Rodrigo

Signature: .....

Dr. Ajith Pasqual

Signature: .....

Dr. Peshala G. Jayasekara

Signature: .....

Date: .....

Declaration

This declaration is made on October 5, 2017.

Declaration by Project Group

We declare that the dissertation entitled People Counting and Tracking with Xilinx ZC702 Evaluation Kits and the work presented in it are our own. We confirm that:

- this work was done wholly or mainly in candidature for a B.Sc. Engineering degree at this university,
- where any part of this dissertation has previously been submitted for a degree or any other qualification at this university or any other institute, has been clearly stated,
- where we have consulted the published work of others, is always clearly attributed,
- where we have quoted from the work of others, the source is always given. With the exception of such quotations, this dissertation is entirely our own work,
- we have acknowledged all main sources of help.

....

Date R.V.C.N Abeyrathne (130008K)

.....

D.L Dampahalage (130093M)

.....

H.A.S.P Gunasekara (130183N)

.....

W.M.D.K Weerakoon (130633V)

Declaration by Supervisors

We have supervised and accepted this dissertation for the submission of the degree.

.....	.....
Dr. Ranga Rodrigo	Date
.....	.....
Dr. Ajith Pasqual	Date
.....	.....
Dr. Peshala G. Jayasekara	Date

Abstract

People Counting and Tracking with Xilinx ZC702 Evaluation Kits

Group Members: R.V.C.N Abeyrathne, D.L Dampahalage, H.A.S.P Gunasekara, W.M.D.K Weerakoon

Supervisors: Dr. Ranga Rodrigo, Dr. Ajith Pasqual, Dr. Peshala G. Jayasekara

Keywords: FPGA,ZYNQ-7000, people counting, multi-camera, multiple people tracking

Abstract comes here

Acknowledgments

We would like to express our deepest appreciation and sincere gratitude to our project supervisors, Dr. Ajith Pasqual, Dr. Ranga Rodrigo and Dr. Peshala Jayasekara of Department of Electronic and Telecommunication Engineering of University of Moratuwa, for their guidance and constant supervision as well as for providing necessary information regarding the project.

We would like to thank ParaQum Technologies for their guidance and support given to us during the project. We are indebted to our final year project coordinator, Dr. Anjula Silva for his counsel regarding the project. We would also like to take this opportunity to thank Ms. Salgado for her advices on compiling a finer report. We would like to acknowledge with much appreciation the non-academic staff of the department who gave us their fullest support in providing necessary laboratory facilities.

Finally, our thanks and appreciations also goes out to our batch mates who helped in numerous ways to make this endeavor a success.

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

FPGA	Field Programmable Gate Array
YOLO	You Only Look Once
IP	Intellectual Property
SoC	System on Chip
RTL	Register Transfer Level
AXI	Advanced eXtensible Interface
HLS	High Level Synthesis
UIO	Userspace Input/Output
REST	Representational State Transfer
API	Application Programming Interface
AJAX	Asynchronous JavaScript ond XML
JSON	JavaScript Object Notation
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
URL	Uniform Resource Locator
SDRAM	Synchronous Dynamic Random Access Memory
DDR	Double Data Rate

Chapter 1

INTRODUCTION

In the contemporary society making correct decisions is vital for any business organization to stay on par with competitors. For that intent identifying and understanding the customers is a must. Tapping into the customer'sâĂŹ subconscious is the preeminent way of making correct business decisions and for that an organization should track and analyze customer behavior.

For retail stores and shopping malls gathering customer insight could be done by analyzing the behavior of day to day customers. However counting and keeping track of customers is a tedious task for a large store structure by just employing a number of cameras and a manual System.

As a possible solution, we have proposed a people tracking and counting system adaptable to any large scale store structure. The objective of this project is to develop a system that is able to process multiple video streams obtained through separate cameras mounted on a store structure in order to track customers and generate business intelligence.

There have been several systems which handle multi camera people tracking introduced in the past decade. All these systems incorporate a central server which will receive all the frames from all the cameras to handle the people detection and tracking tasks real-time. This is not scalable in a large environment unless the central server has enough processing power. Solution to this problem is to do part of the processing close to the camera which we define as leaf node processing.

We propose a system which utilizes a FPGA + ARM Processor at the leaf node for the people detection. This system will then send the information to the central node for Multi-Camera Tracking, thereby reducing the processing power required at the central server. Also the system is only sending the bounding boxes and features for tracking which reduces the bandwidth usage in such systems.

Scope of this project is to implement a scalable end to end system for Multi-Camera People Tracking utilizing a Zynq SoC for leaf node processing. Thereby we use an FPGA + ARM processor for leaf node processing, a central server for multi-camera people tracking and a web server for generating business intelligence.

People Detection in FPGA is an aspect that has been researched by many over the years. For example [2] suggests a FPGA based embedded platform for people detection using Gaussian Mixture Models and [3] suggests a similar system for people detection using Histogram of Features. Multi-Camera People Tracking systems is also a field which has been researched over the years. [4] suggests a multi-camera surveillance system implemented with background subtraction for detection and color features for tracking. [6] suggests another multi camera people tracking system which uses particle

filters for tracking on the ground plane.

Major drawback of these systems is the scalability. When the number of cameras increase, the bandwidth usage for sending video frames to the central server will be increased and the processing power required at the central server will increase. This raises the requirement for processing video frames close to the camera which is defined as leaf node processing.

In this proposed system by using a FPGA + ARM processor we have achieved leaf node processing of people detection at a lower power and also enable the scalability and lower bandwidth requirement.

Chapter 2

LITERATURE REVIEW

In this project main focus was to implement an end to end system for Multi-Camera People Tracking System and to use this system to generate business intuition. As explained earlier we use the technique of leaf node processing to achieve a scalable, low power, low bandwidth required system.

There are several challenging tasks in implementing this system. One is people detection in FPGA. There are various ways to achieve this. One way is to use a blob detection algorithm as used by Vicente, Alfredo Gardel, et al. [1] in 2009.They have done background subtraction followed by contour detection to select head candidates for videos obtained using overhead cameras and they have implemented people detection part in a low cost FPGA (spartan3).

Feature based methods are also another way to do people detection. In these methods a set of features are calculated around a window and some kind of machine learning algorithm is used to train a classifier that can classify each window into "contains a person" and "not contains a person". One such method is Histogram Oriented Gradient (HoG) based method suggested by Dalal, Navneet, and Bill Triggs. [2] in 2005. A variation of HoG is implemented on FPGA by Negi, Kazuhiro, et al. [3] in 2011.

Another approach that can be used for people detection is to utilize a neural network. If we can come up with a suitable neural network architecture, this method has the potential to outperform all the other methods mentioned earlier. There are several open source implementations of such architectures available. One such architecture is given by Redmon, Joseph, et al. [4] in 2016. Their work has made a huge leap in object detection space. Due to its success their architecture is utilized by many people. Challenge though in using this approach is the complexity of implementing a convolutional neural network in an FPGA. Implementing a CNN architecture in FPGA with the available resources and to achieve real time processing is an immense challenge.

We used a background subtraction based method for people detection in the proposed system. Though we considered using either HoG or CNN, as our focus was to implement the end end architecture for multi-camera people tracking with leaf node processing, background subtractions deemed to be a feasible solution. We could improve the existing system by implementing either HoG or CNN as a future improvement.

Other major challenge we have is people tracking based on our detections. There are many traditional approaches to this such as tracking using kalman filter and assigning detections to tracks using the Hungarian algorithm. But multi target tracking also can be formulated as a discrete-continuous optimization problem [5]. Andriyenko, Anton, Konrad Schindler, and Stefan Roth [5] in 2012 have considered data association as a discrete optimization problem with label costs. And they have posed trajectory estima-

tion as a continuous fitting problem with closed form solutions. And there method has performed well on some known data sets.

Other major challenge that we face is, how to extend multiple target tracking into a multi camera framework. There are some research papers discussing this issue. Tang, Nick C., et al. [6] in 2015 suggest a two pass regression framework to solve this challenge. First pass regression predicts the people count based on the features calculated from intra-camera video frames. And then the second pass is based on the conflicts between the prediction derived from multiple views. They have formulated this as a transfer learning problem. Yang, Tao, et al. [7] in 2007 discusses another method to do this. In this approach first they do single camera tracking and they transform these tracked paths into a global coordinate system. Then by using their multi camera handoff algorithm, they can track people across multiple cameras. Here they calculate a match score for an object appearing in a camera under overlapping or non overlapping conditions for all tracks under all cameras. The track having the maximum score is selected.

By going through the available literature we were able to get an insight of the scope of our project. We identified some potential solutions for the challenges we have. We were also able to identify that implementing the overall system architecture for multi-camera people tracking with leaf node processing will be a novel contribution.

Chapter 3

METHODOLOGY

3.1 Introduction

In this project we have designed and implemented an end to end multi camera people counting and tracking system based on Xilinx ZC702 development board. In this implementation we have considered the possibility of using leaf node processing to develop a scalable solution for multi camera people tracking. In this chapter we present the underline methodology of our system in detail. In order to give the reader a gradual walkthrough of our project we will explore each component of our system separately. First we will give an overview of the basic system. Then we will explore in detail the each component of our system.

3.2 Overall Architecture

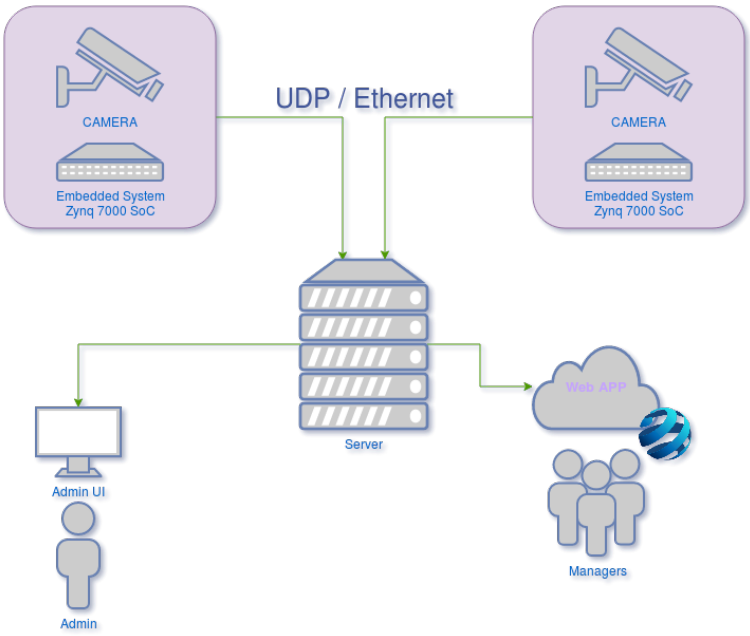


Figure 3.1: Overall block diagram of the end-to-end system

Figure 3.1 shows the overall architecture of our system. It mainly consist of 3 components, namely, An embedded system closer to the camera (Leaf node) People tracking applications running on a server Business intelligence generation and displaying interfaces.

We will explore each of the above components separately in the following sections.

3.3 Embedded System (Leaf Node)

Generally the term 'leaf node' is used for a system placed at the edge or bottom of a hierarchical network (of systems). In our system, leaf node is the embedded system connected to the camera. This embedded system is responsible for capturing live video frames from the camera and preprocessing before sending into the central server. Objective of preprocessing on the leaf node is to reduce the required bandwidth of the network to send information (video in this case) to the central server and to reduce the processing done one the central server in a scalable manner. In a large system consisting of few hundreds of cameras, reduction of the bandwidth required and the processing power of the central system for each camera makes a big difference in the central system processing and bandwidth requirements, making the system more scalable.

In our prototype system, leaf node consists of a web camera connected to a Xilinx ZC702 development board and its responsible for doing people detection of the live video feed and feature calculation of the detected people. Implementations of the people detection and feature calculation algorithms are explained in detail in later sections. These calculated features of the detected people are then sent to the central server via a custom application layer protocol based on User Datagram Protocol (UDP) / Internet Protocol (IP) over Ethernet.

Following sections provide an overview about the feature of Xilinx ZC702 board and ZYNQ-7000 All Programmable (AP) System on Chip (SoC).

3.3.1 Xilinx ZC702 Board

Xilinx ZC702 board, shown in Figure 3.2, is a development board manufactured by Xilinx targeting embedded hardware (FPGA) design development. This board consists of a ZYNQ-7000 AP SoC with various other peripherals. Key features and peripherals of Xilinx ZC702 is listed below.

- Zynq-7000 SoC
- 1GB DDR3 Component Memory
- Enabling serial connectivity with USB OTG, UART, IIC, CAN Bus
- Ethernet which supports 10-100-1000 Mbps transfer rates
- HDMI interface
- FPGA Mezzanine Card (FMC) interface
- Onboard Secure Digital (SD) card reader

Figure 3.2: Xilinx ZC702 Development Board

Features of the onboard ZYNQ-7000 SoC and usage of it in our project is explained in following sections.

3.3.2 ZYNQ-7000 SoC

ZYNQ-7000 SoC consists of a Xilinx XC7Z020-CLG484-1 FPGA and two ARM Cortex-A9 core processors. Block diagram in figure 3.3 shows how the Programmable Logic and the Processing System is connected inside the ZYNQ-7000 SoC.

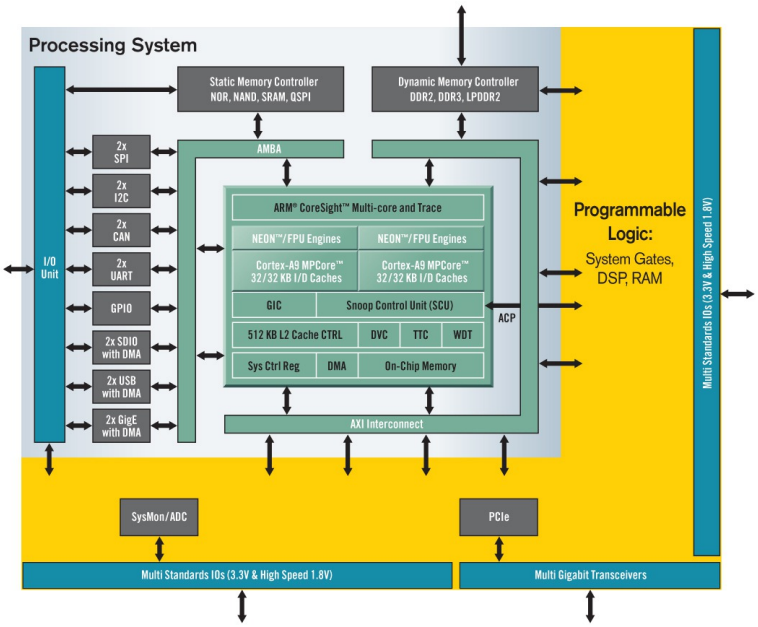


Figure 3.3: Internal architecture of ZYNQ-7000 SoC

In our implementation, we have installed a lightweight Linux distribution on dual ARM processors in ZYNQ-7000 SoC. Reasons for installing Linux and its usage is explained in a latter section.

3.4 Design Flow for ZYNQ-7000 based Embedded System

We used the standard Xilinx toolchain for developing an embedded system for designing and implementing our leaf node system. Figure 3.4 shows a block diagram of the design flow with the design tools we used.

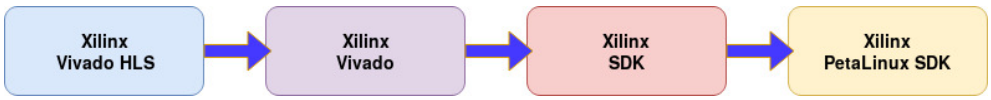


Figure 3.4: Design flow block diagram for ZYNQ-7000 based embedded system

Brief description about the usage of each tool in the design flow is included in the following subsections.

3.4.1 Vivado HLS

HLS stands for High Level Synthesis. Task of Vivado HLS is to synthesize a function developed using a high level language (C, C++ and SystemC) into a RTL design (Hardware Descriptive Language Code) of an IP Core.

3.4.2 Vivado

IP cores designed using Vivado HLS are then imported to Vivado for designing the overall architecture. This is the tool where we decide how the PL and PS parts of ZYNQ SoC is connected (which type of AXI protocol). After overall system is designed, its synthesized and implemented on the target device. As an output of this process, bitstream file and hardware description (HDF) file is generated.

3.4.3 Xilinx SDK

HDF file generated is then imported to Xilinx SDK for testing the functionality of implemented hardware. Xilinx SDK has options for testing the hardware on the target device by writing a simple C/C++ application code for testing the functionality of the implemented hardware.

3.4.4 PetaLinux SDK

After confirming the functionality of the hardware on the target device, HDF file is then imported to a PetaLinux SDK project for generating the Linux boot files for the custom hardware. Enabling the required kernel modules and editing the device tree to enable hardware peripheral access is done here. More details about running Linux on ZYNQ PS is explained in the next section.

3.5 Installing Linux on ZYNQ-7000

Lightweight customized Debian Linux distribution was installed on ARM processors of ZYNQ SoC. When it comes to running applications on PS of ZYNQ, there are two primary options,

1. Running a Linux distribution
2. Running the application in baremetal mode (Using the basic drivers provided by Xilinx)

We choose to install and run a Linux distribution based on the following reasons,

- Complete and working protocol stacks for communication protocols (Ethernet, UART)
- Readily available device drivers for interfacing various hardware (USB web camera, USB tethering)
- Root file system stored in the SD card (Usually bare-metal application use on-board RAM to store data at run time, which will be destroyed after turning off the board)
- Linux comes with a package manager that can be used to install almost any third party Linux compatible software package (Python, C++ compilers).

3.6 People Detection and Feature Calculation Implementation on ZYNQ-7000 SoC

3.6.1 IP core implementation on ZYNQ PL

Gaussian Mixture Model Background Subtraction IP Core

Feature Calculation IP Core

3.6.2 IP Core drivers, Morphological Operations and Communication Protocol Implementation on ZYNQ PS

Linux Drivers for controlling IP cores

People Detection from Background Subtracted Binary Mask

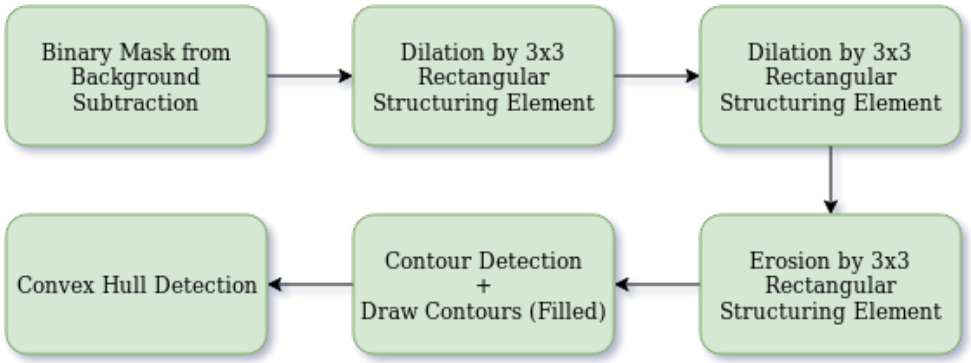


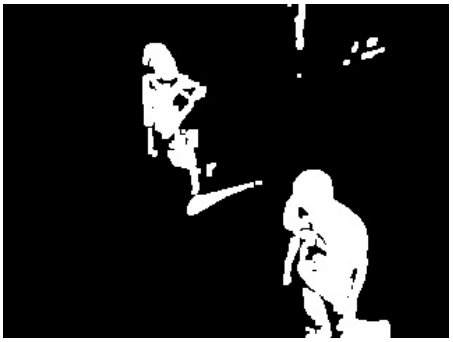
Figure 3.5: Block diagram of the morphological operation flow to detect people from background subtracted binary mask

Once the background subtracted binary mask is obtained a set of operations are performed as shown in figure 3.5 to enhance the binary image for object detections. First a set of morphological operations are performed to clean out the image as necessary. Since the quality of the background model can vary due to various environmental conditions, background subtraction will not give a cleanly filled blob for the shape of a person. There will be a lot of black pixels on the boundary as well as inside the shape of a person. We can use the morphological operation âĂŸclosingâĂŹ to fill out these holes. It is essentially two dilations followed by an erosion. Structuring element used here is a 3x3 rectangle. We have tested out several alternatives for the kernel but this kernel size and shape gave the best results for the sequences of images we have tested. Therefore we used it in our final implementation. The results of morphological operations for a test images is shown in figure 3.6c, 3.6d and 3.6e.

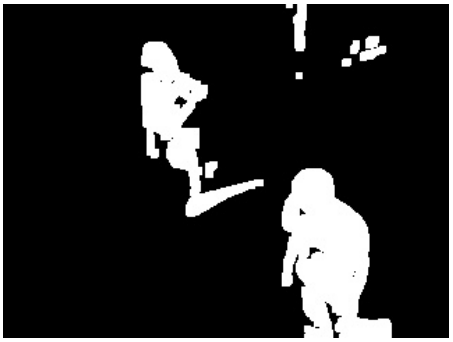
Morphological operations will give a cleaner image compared to the original background subtracted image. Next contour detection is applied to the output image resulting

(a) Original image

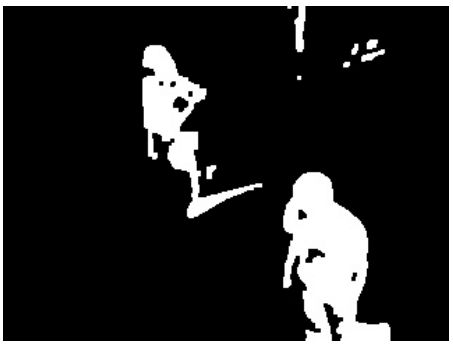
(b) Background subtracted binary mask



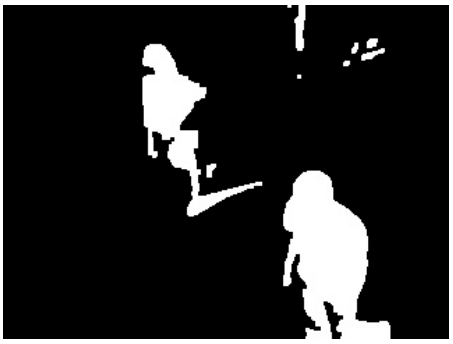
(c) Dilation performed on the binary mask



(d) Twice dilated



(e) Twice dilated and eroded



(f) Contour detection and drawing the filled contours taking 3.6e as input

(g) Detecting bounding boxes using convex hull detection

Figure 3.6: Results of morphological operations for a test image

from morphological operations. We have used inbuilt functions of opencv for this. They have implemented contour detection based on SuzukiâĂŹs [7] border following algorithm. Then again using inbuilt functions of opencv we fill out these contours. It will completely fill the shape of the person. The result of this operation for a test image is shown in figure 3.6f.

Next step in this process is to detect the bounding box surrounding the blob of a person. For this we use opencv inbuilt function for convex hull detection. A convex hull is the smallest polygon enclosing a set of points. OpencvâĂŹs implementation is based on the algorithm proposed by Sklansky [8]. After this process we obtain a set of bounding boxes corresponding to people detected. Resulting bounding boxes for a test image is shown in figure 3.6g.

3.6.3 False Positive Reduction

Although the above process is able to detect people from an image, it also detects a lot of unwanted blobs due to various reasons such as reflections caused by background surfaces, imperfections of background modeling, etc. Therefore we need to utilize a method to reduce number of these false positives. Our initial approach was to select the bounding boxes satisfying a set of conditions as follows and reject all the other detections.

- Width of bounding box(w) > TW
- Height of bounding box(h) > TH
- TAR1 > Aspect ratio(w/h) > TAR2
- TCA1 > Contour Area > TCA2

Here TW, TH, TAR1, TAR2, TCA1 and TCA2 are constants that are chosen arbitrarily. Initially we tuned up these constants manually by looking at the results. But this was a tedious task and this has to be redone once the camera position is changed. Therefore we explored the possibility of machine learning based automatic way to do this.
We selected a few features of the bounding boxes into consideration as follows:

1. Width
2. Height
3. Bounding box area
4. Aspect Ratio (Width / Height)
5. Contour Area
6. Diagonal Length

Table 3.1: Percentage of variance along each pricipal component

Principal Component	Percentage of Variance Explained
1	99.4509
2	0.5401
3	0.0063
4	0.0019
5	0.0000
6	0.0000

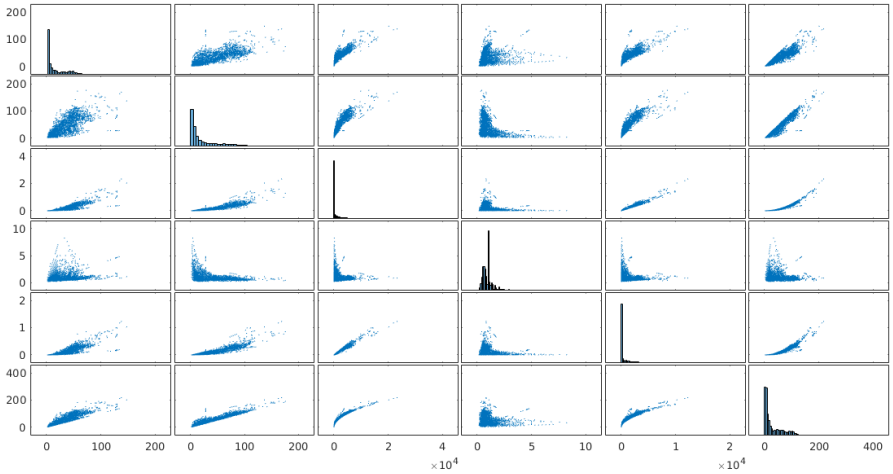


Figure 3.7: Scatter plot matrix for the 6 features collected from running our object detection on a sample video

We can see from the scatter plots above that most of the above features are highly correlated. Therefore we can use Principal Component Analysis to perform dimensionality reduction of the feature space. The percentage of variance explained by each principal component was obtained as follows.

We can see that only the first two principal components give a significant contribution. Even the 2d principal component is less significant compared to the 1st principal component. We can further see this in figure 3.8.

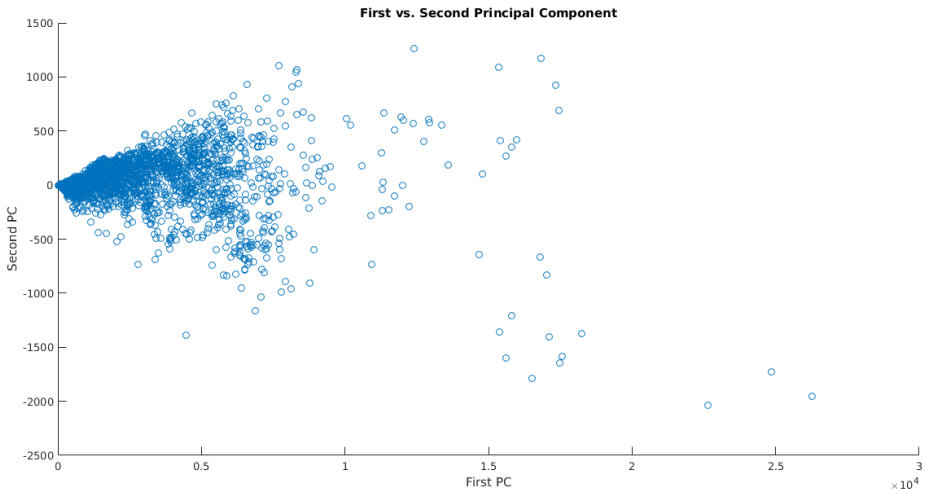


Figure 3.8: First Vs. second pricipal component

Therefore it is justifiable only to select the first principal component into consideration.

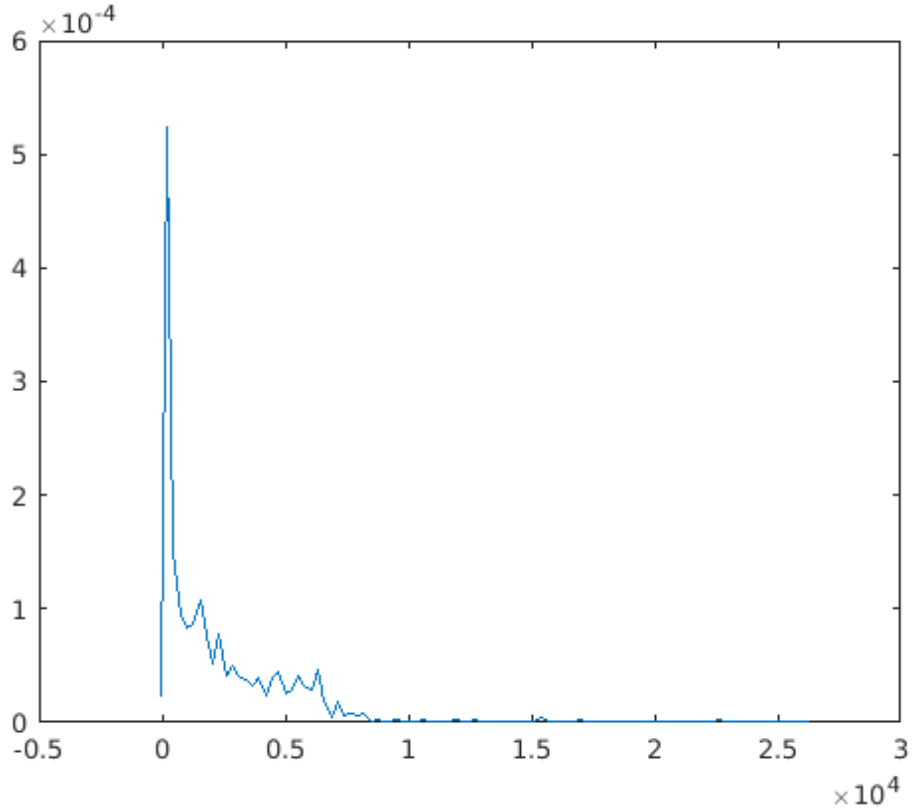
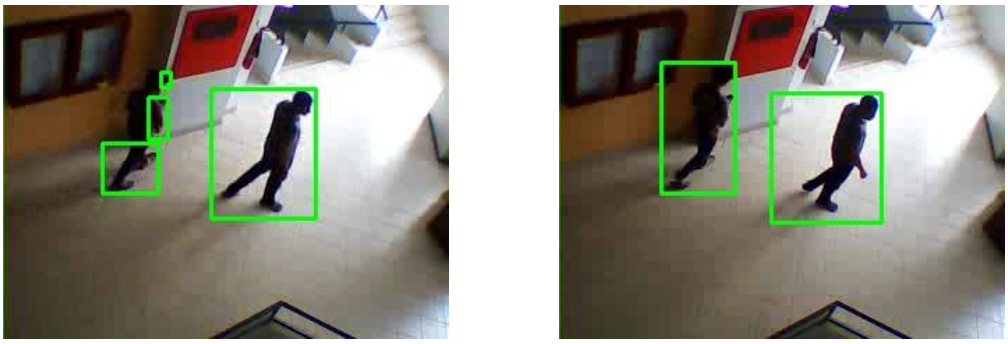


Figure 3.9: Density of the first principal component

Next we looked at the kernel density estimate of the first principal component as shown in figure 3.9. We can approximate this with a bimodal distribution. Therefore



(a) Raw bounding boxes detected

(b) With false positive reduction

Figure 3.10: Results of PCA based false positive elimination algorithm

the problem reduces to finding the optimum threshold for this bimodal distribution. We used OtsuâĂŹs [9] thresholding algorithm to find this threshold.

We utilized this framework for false positive reduction of the detections. Once we tested out this with several video sequences taken from different camera angles we found out that this method indeed gives better results as analysis suggested. Figure 3.10 shows the results of PCA based false positive reduction.

When applying this method the system should first undergo a training stage where raw detection features are collected. Then coefficients of the first principal component is determined using this dataset. Then the optimum threshold is determined. This whole process was implemented in code so that system can automatically learn to reduce false positives. In the normal mode of operation the system directly uses these calculated coefficients and threshold for false positive reduction.

Chapter 4

RESULTS

4.0.1 Discussion

BIBLIOGRAPHY

[1] Vicente, Alfredo Gardel, et al. "Embedded vision modules for tracking and counting people." *IEEE Transactions on Instrumentation and Measurement* 58.9 (2009): 3004-3011.

[2] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." Computer Vision and Pattern Recognition, 2005. CVPR 2005. *IEEE Computer Society Conference on*. Vol. 1. IEEE, 2005.

[3] Negi, Kazuhiro, et al. "Deep pipelined one-chip FPGA implementation of a real-time image-based human detection algorithm." *Field-Programmable Technology (FPT), 2011 International Conference on*. IEEE, 2011.

[4] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[5] Andriyenko, Anton, Konrad Schindler, and Stefan Roth. "Discrete-continuous optimization for multi-target tracking." *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012.

[6] Tang, Nick C., et al. "Cross-camera knowledge transfer for multiview people counting." *IEEE Transactions on image processing* 24.1 (2015): 80-93.

[7] Yang, Tao, et al. "Robust people detection and tracking in a multi-camera indoor visual surveillance system." *Multimedia and Expo, 2007 IEEE International Conference on*. IEEE, 2007.

[8] Flask.pocoo.org. (2017). Tutorial âĂŤ Flask Documentation (0.12). [online] Available at: http://flask.pocoo.org/docs/0.12/tutorial/ [Accessed 6 Oct. 2017].

[9] Pythonprogramming.net. (2017). Python Programming Tutorials. [online] Available at: https://pythonprogramming.net/flask-registration-tutorial/ [Accessed 6 Oct. 2017].

[10] Flask-restful.readthedocs.io. (2017). Flask-RESTful âĂŤ Flask-RESTful 0.3.6 documentation. [online] Available at: https://flask-restful.readthedocs.io/en/latest/ [Accessed 6 Oct. 2017].