Improving Indoor Security Surveillance by Fusing Data from BIM, UWB and Video

Mahsa Rafiee

A Thesis

in

Concordia Institute for

Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science (Information Systems Security)

at

Concordia University

Montreal, Quebec, Canada

January 2014

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: Mahsa Rafiee

Entitled: Improving Indoor Security Surveillance by Fusing Data from BIM, UWB and Video

and submitted in partial fulfillment of the requirement for the degree of

Master of Applied Science (Information Systems Security)

complies with the regulations of the University and meets with the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. C. Wang ______________________ Chair

Dr. M. Mannan ______________________ Examiner

Dr. Z. Zhu ______________________ Examiner

Dr. Amin Hammad ______________________ Supervisor

Approved by ______________________

Chair of Department or Graduate Program Director

__________ ______________________

Dean of Faculty

ABSTRACT

Improving Indoor Security Surveillance by Fusing Data from BIM, UWB and Video

Mahsa Rafiee

Indoor physical security, as a perpetual and multi-layered phenomenon, is a time-intensive and labor-consuming task. Various technologies have been leveraged to develop automatic access control, intrusion detection, or video monitoring systems. Video surveillance has been significantly enhanced by the advent of Pan-Tilt-Zoom (PTZ) cameras and advanced video processing, which together enable effective monitoring and recording. The development of ubiquitous object identification and tracking technologies provides the opportunity to accomplish automatic access control and tracking. Intrusion detection has also become possible through deploying networks of motion sensors for alerting about abnormal behaviors. However, each of the above-mentioned technologies has its own limitations. This thesis presents a fully automated indoor security solution that leverages an Ultra-wideband (UWB) Real-Time Locating System (RTLS), PTZ surveillance cameras and a Building Information Model (BIM) as three sources of environmental data. Providing authorized persons with UWB tags, unauthorized intruders are distinguished as the mismatch observed between the detected tag owners and the persons detected in the video, and intrusion alert is generated. PTZ cameras allow for wide-area monitoring and motion-based recording. Furthermore, the BIM is used for space modeling and mapping the locations of intruders in the building. Fusing UWB tracking, video and spatial data can automate the entire security procedure from access control to intrusion alerting and behavior monitoring. Other benefits of the proposed method include more complex query processing and interoperability with other BIM-based solutions. A prototype system is implemented that demonstrates the feasibility of the proposed method.

ACNOWLEDGEMENT

My greatest appreciation goes to my supervisor, Dr. Amin Hammad for his intellectual and personal support, encouragement and patience. His advice and criticism was my most valuable asset during my studies. Overall, I feel very fortunate having the opportunity to know him and work with him.

I would like to acknowledge the contributions of Dr. Zhenhua Zhu and Ms. Humaira Tajeen for providing me with the image processing code and support, and Mr. Hassaan Siddiqui for developing the data fusion application and performing several tests with me. I would also like to thank my colleagues especially Mr. Mohammad Soltani and Mr. Shayan Setayeshgar for their help and valuable advice. I would like to give my special thanks to Mr. Bahman Rafiee, Ms. Mahrokh Naseri, Mr. Farrokh Rafiee, Mr. Manour Rafiee, Ms. Mehrangiz Rafiee, and Mr. Hamed Roshanaei for their continuous support to accomplish my Master's degree.

This thesis is dedicated to my father, mother and brothers for their endless love and encouragement during my life.

Contents

List of Figures

List of Tables

List of Abbreviations

2D	Two Dimensional
3D	Three Dimensional
6D	Six Dimensional
AIA	Automated Imaging Association
AIITS	Automatic Intruder Identification and Tracking System
ACL	Access Control List
ANN	Artificial Neural Networks
AoA	Angle of Arrival
AP	Access Point
API	Application Programming Interface
BCS	BIM Coordinate System
BIM	Building Information Modeling
CAD	Computer-Aided Design
CCS	Camera Coordinate System
CCTV	Closed-Circuit Tele-Vision
COM	Component Object Model
CTIAI	Corporation de Technologie d'Intelligence et d'Automatisation Internationale
CV	Computer Vision
DARPA	Defense Advanced Research Projects Agency
DIAG	DIAGnostic Responder Protocol
DLL	Dynamic-Link Library
DSC	Distributed Smart Cameras
DSP	Digital Signal Processor
DVR	Digital Video Recorder
FCC	Federal Communications Commission
FPGA	Field Programmable Gate Arrays
FMCW	Frequency Modulated Continuous Wave
FoV	Field of View
GPS	Global Positioning System
GUI	Graphical User Interface
HPoE	High-Power over Ethernet
IDE	Integrated Development Environment
IDS	Intrusion Detection System
IFC	Industry Foundation Classes
IP	Internet Protocol
IPTO	Information Processing Technology Office
IR	Impulse Radio
ISO	International Standards Organization
ITU	International Telecommunication Union
JDL	Joint Directors of Laboratories
JPEG	Joint Photographic Experts Group
k-NN	*k*-Nearest Neighbors
k-SP	*k*-Shortest Paths
LBS	Location-Based Services

LED	Light-Emitting Diode
LoS	Line-of-Sight
LP	Linear Programming
LVQ	Learning Vector Quantization
MEP	Mechanical, Electrical, and Plumbing
MPEG	Moving Picture Experts Group
MVaaS	Managed Video as a Service
MSDF	Multi-Sensory Data Fusion
NIST	National Institute of Standards and Technology
NLoS	None-Line-of-Sight
NVR	Network Video Recorders
PoE	Power over Ethernet
POM	Probabilistic Occupancy Map
PoS	Point of Sale
PSC	Pervasive Smart Cameras
PTZ	Pan-Tilt-Zoom
RF	Radio Frequency
RFC	Remote Function Call
RFID	Radio Frequency Identification
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
RTLS	Real-Time Locating System
SaaS	Software as a Service
SAP	Session Announcement Protocol
SDK	Software Development Kit
SMP	Smallest M-vertex Polygon
SNC	Secured Network Communication
SOC	System On Chip
SVM	Support Vector Machines
TDoA	Time Difference of Arrival
ToA	Time of Arrival
UCS	UWB Coordinate System
UMTS	Universal Mobile Telecommunications System
UWB	Ultra-Wide Band
VCR	Video Cassette Recorders
VIRAT	Video Image Retrieval and Analysis Tools
VLSI	Very Large Scale Integration
WSN	Wireless Sensor Networks
XQuery	cross-data Query

CHAPTER 1 INTRODUCTION

1.1 General Review

Physical security aims at convincing the intruders that attack costs would exceed the gained value and is practically enforced in-depth, at least in four layers: (1) environmental design to deter the threats (e.g. fences); (2) access control to restrict admission only to the authorized persons (historically, by mechanical keys and locks); (3) Intrusion Detection Systems (IDSs) to alarm the suspicious behavior for an appropriate defensive response (e.g. motion sensors); and (4) identification and incident verification to prosecute the criminals (e.g. video surveillance).

Traditionally, human-based and mechanical means could not fully accomplish the "Access Control" as a matter of "who, when, where", i.e. mechanical locks could not restrict the authorized persons' access to the specific dates and times. Besides, mechanical keys could be easily transferred or copied for illegal access, and in case the key owner changed or the key is lost, it had to be replaced. Furthermore, they did not provide access logs; therefore the mechanical means becomes quickly unmanageable for large user populations. Technology development enabled establishment of the electronic *Access Control Systems* that can scale from a single area to enterprise size, be administered by a remote client program, and record any access attempts whether refused or admitted. Electronic *Access Control Systems* grant access to the person only upon presenting (sophisticated) electronic credentials and can control their authorization timeout according to the predetermined time in their electronic credentials.

Ubiquitous computing has enabled devising pervasive IDSs which sense the environment and detect the unauthorized access to alert the human officers or trigger a pre-determined physical defensive action. Deployment of cost-effective Radio Frequency (RF)-based sensor networks enables detecting and tracking the authorized persons to identify any abnormal behavior.

Finally in the last layer, *Video Surveillance Networks* have been in widespread use since many years ago which provide a visual evidence of any incident for later prosecution. *Video Surveillance Networks* also have been drastically changed from the analog Closed-Circuit Tele-Vision (CCTV) networks to the digital Internet Protocol (IP) cameras which can connect to the Network Video Recorders (NVR) and various video analytic tools. State-of-the-art PTZ cameras enable for remote directional and zoom control to reach a large dynamic Field of View (FoV) from a fixed installed point, through scheduled tours or manual rotations and even auto-tracking a target.

Despite the deployment of the layer-specific automated security systems, the human workforces are still intertwined in the different tasks whether as the patrols at the checkpoints, the administrators of the access control system, the alarm responders or the video observers and analyzers. In an effort to increase the automation level of the indoor security, this research is dedicated to integrating the intrusion detection task with the intruder identification, locating and tracking functions within a hybrid system named "*Automatic Intruder Identification and Tracking System"* or shortly AIITS. We assume a hybrid system, by taking advantage of the competent technologies and fusing the complementary and confirmatory data, achieves a higher degree of reliable automated security.

The framework introduced in this research leverages three sources of environmental data; one for space modeling and location mapping, and two for real-time location and identification data sensing. The two sensor components are adopted to provide the supplementary dynamic data for: (a) distinguishing the intruders from the authorized persons, and (b) computing the location of the detected persons for visual tracking. The proposed framework consists of: (1) a **BIM** which includes the space design and sensor network deployments, and assists in removing noisy location data, finding the intruder in the building and routing the security officers through the shortest path through visualizing the sensed data via: (2) an **UWB RTLS** which detects and tracks the (authorized) tagged persons; and (3) a **PTZ Video Camera** which provides the input video for Computer Vision (CV)-based analysis in order to locate and count the human bodies so that the untagged intruders could be located for visual auto-tracking.

1.2 Research Objectives and Contribution

This research aims at improving the indoor security by expanding the basic intrusion detection functionality to support the *Automatic Intruder Identification and Tracking*. In an effort to increase the automation level and the accuracy degree of the indoors security, this research targets the implementation of the intruder tracking function through the following steps: (1) intrusion detection, (2) intruder identification and locating, and (3) visual intruder tracking. The objectives of this research are to:

(1) Describe an architecture for the AIITS, built upon BIM, UWB RTLS and PTZ camera,

(2) Define and analyze the requirements of the proposed architecture,

(3) Elaborate on the proposed methodology for technology integration, data fusion, intruder tracking, and post-event cross-data query (XQuery) processing,

(4) Implement a prototype system that can validate the proposed approach using a case study.

The framework introduced in this research proposes using BIM, a geospatial data source which contributes as a reliable decision basis for the optimum sensor networks deployment, finding noisy location data by mapping on the 3D model, finding the intruder's location in the building, and routing the security officers through the shortest path to the located intruder. BIM, as a standard information system, also enables the security system to be integrated and accessed with other organizational information systems (Motamedi, 2009).

UWB, by achieving the most accurate 3D RF-based indoor positioning as opposed to the traditional narrowband Radio Frequency Identification (RFID) technologies, allows for obtaining the identification and location data in real-time with less computation compared to the CV-based human identification and locating. It also enables system expansion to include many applications that require the spatial relationships between the mobile tagged objects e.g. abnormal activity detection, elderly care and fall detection, etc.

The introduced framework, leveraging the PTZ camera technology can be programmed for a myriad of the intelligent surveillance and tracking applications. While a fixed camera cannot track a target out of its FoV and therefore a large number of fixed cameras with a complicated coordination and hand over algorithm is required for tracking people across a large area, the PTZ camera enables large area coverage and automatic visual tracking.

Finally, although similar researches have been performed on fusing the radio and video data for indoor positioning and tracking purposes, to the best of our knowledge no system has been built upon the joint use of the UWB RTLS, PTZ cameras and the BIM geospatial resource. The key feature of this thesis which makes it different from the others is proposing the data fusion among the complimentary and confirmatory items of the geospatial, radio and video data from the BIM, UWB RTLS, and PTZ camera resources.

1.3 Thesis Organization

This study will be presented as follows:

Chapter 2 Literature Review: this chapter presents our technological review about the state-of-the-art technologies which can be integrated for improving indoor security surveillance, including: BIM, UWB RTLS, and PTZ camera. The idea of data fusion which makes the proposed technology worthwhile to research is elaborated through the introduction of the suggested techniques and adopting the best appropriate paradigm. Our review of the related work in fusing the radio and video identification and locating data is presented afterwards to highlight the problem statement which motivated us for this research.

Chapter 3 Requirement Analysis and Proposed Methodology: this chapter starts with the requirement analysis for a reliable AIITS and technology adoption and integration criteria. Based on the discussed requirements and criteria, we propose an approach for *Automatic Intruder Identification and Tracking,* including: (1) registering the authorized persons in the RTLS for identification, locating, and tracking; (2) CV-based person detection and locating; (3) identification of the untagged person (intruder) among the detected persons; (4) finally visual intruder auto-tracking; and (5) XQuery processing. Then, a system design is proposed to be implemented as a prototype AIITS in next chapter.

Chapter 4 Implementation and Case Study: this chapter presents the implementation approach for the design proposed in Chapter 3 which includes: introduction of the software and hardware components of the leveraged technologies, their APIs, the programming platform and language, and ultimately the developed software modules. A case study is designed to validate the feasibility and competence of the proposed AIITS design and approach.

Chapter 5 Conclusions and Future Work: this chapter summarizes the present work, highlights its contributions and added values, and suggests the future work which enhances the current results and expands its functionalities and scope.

CHAPTER 2 LITERATURE REVIEW

2.1 Introduction

In this chapter the state-of-the-art technologies for the automatic real-time object detection and tracking are introduced and assessed. The technologies discussed in the present chapter support our data fusion idea for achieving a higher degree of automation and reliability in indoors security. This literature review comprises a brief introduction of the BIM and thorough discussions about the UWB RTLS and video surveillance technologies. Also, the related research in the area of human identification and tracking are reviewed. The purpose of this literature review is to highlight the capabilities of the state-of-the-art technologies and the shortcomings of the current research trends, which motivate us for fusing complementary and confirmatory data from multiple technologies, as it will be discussed in the next chapter.

This chapter is organized as follows: Section 2.2 reviews building modeling technologies and highlights BIM's advantages; Section 2.3 reviews two complementary indoor tracking technologies, namely UWB RTLS and PTZ video cameras; Section 2.4 discusses data fusion techniques; Section 2.5 reviews the related works that use the above-mentioned technologies, individually and in hybrid designs which fuse the radio and video data, and then highlights the problem statement with the state-of-the-practice; Section 2.6 provides a summary and conclusion about the discussed topics.

2.2 BIM

According to (AGC, 2006), *Building Information Modeling* (BIM) is the development and use of a software model to simulate the construction and operation of a facility. BIM software is a collaboration program that provides a repository for each discipline to add digital, facility-specific knowledge into a shared dynamic model. The resulting model, BIM, is a data-rich, object-oriented, intelligent and parametric digital representation of the building facilities. Different views and data can be extracted from this BIM and analyzed to generate the information that can be used for decision making and improving the process of delivering the facilities. This representation makes it much easier to visualize the completed building, how people will move around, and how spaces relate to each other. A facility in the conventional 3D drafting was illustrated in terms of dimensions (e.g. length, width and height). However, the BIM system is described to be 6D while the model itself is 3D. The fourth dimension is generally denoted as time, the fifth as cost and the sixth as life-cycle, e.g. the building operations over a 20-year period (Strauchs, 2012; Hensworth, 2011).

BIM as an approach requires suitable technology to be implemented to support continuous and immediate availability of project design, scope, schedule, and cost information in high quality, reliability, and interoperability. The technologies which contributed to the development of BIM in increasing order of effectiveness include: *CAD*, *Object CAD*, and *Parametric Building Modeling* (Autodesk, 2003). At the highest effectiveness level, *Parametric Building Modeling* is analogous to the decision support systems used in the financial community. These systems combine a data model (geometry and data) with a behavioral model (change management) that gives meaning to the data through relationships. This provides an integrated system that can be used to simulate the behavior of a real-world system, in this case a building. BIM process obviates extra costs and unanticipated problems, increases efficiency and consistency among stakeholders, and improves scheduling (Autodesk, 2003). The overall effectiveness achievable by each technology, at a given level of effort is illustrated in Figure 2-1.

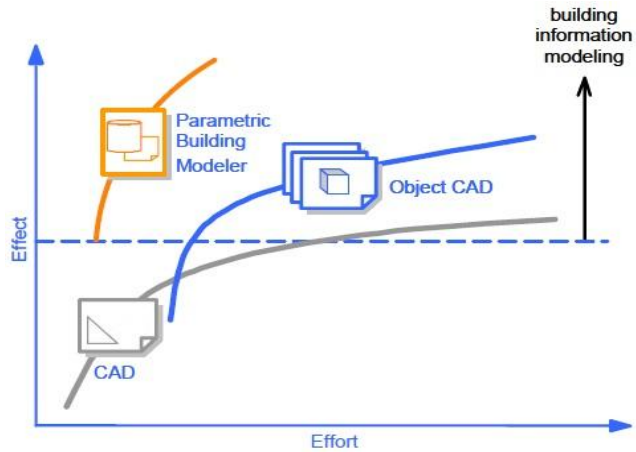


Figure 2-1 Relative Effect and Effort of Building Modeling Technologies (Autodesk, 2003)

By the advent of *Parametric Building Modeling* that uses numbers or characteristics to determine behaviors of a graphical entity and defines the relationships of the model's components, editing the models became much easier. BIM solutions which utilize parametric building modelers can provide more coordinated, reliable, and consistent building information compared to the *Object CAD* software. In *Parametric Building Modeling*, much of the required data for design and performance analysis are naturally captured during the design process**.** However, traditional models generated from *CAD* or *Object CAD* solutions do not contain enough information for building performance analysis. Therefore, a great deal of human intervention and interpretation is required for such analysis (Autodesk, 2007).

For security purpose, the BIM can be utilized to contribute in scheduling the security preventive and maintenance tasks, optimizing guard postings especially guard tours on a continual real-time basis (Strauchs, 2012). The BIM's spatial constraints for building services enable early assessment of the security considerations. BIM's powerful visualization and views empower to select the optimum locations for surveillance cameras and sensors, also to select specific FoVs for the cameras. Moreover, as the BIM shows the details of what exists and where a facility is located in the building, security system can be easily planned to be scaled up (Hensworth, 2011).

2.3 Indoor Tracking Technologies

In this section two widely used sensor technologies for indoor security surveillance are reviewed.

2.3.1 Real-Time Locating Systems (RTLS)

Nowadays, automatic real-time object tracking is possible through the wireless sender-receiver mechanism used in *Positioning Systems* (Zegelin, 2003). This automatic positioning capability is enabler of enormous Location-Based Services (LBS) for management of the assets and personnel, such as: finding the lost items in the warehouse (Hariharan, 2006), locating the construction equipments on the site (Song et al., 2006), collision prevention and construction safety (Hammad et al., 2012), localization of the moving persons in a building (Lee et al., 2008), rescuing persons in underground mines (Zhang & Yuan, 2006), etc.

The *Positioning Systems* exist in two kinds: *local* and *global*. The most well-known *Positioning System* is the *Global Positioning System* or shortly GPS. GPS is satellite-based and it entirely fails for indoors applications, despite successful outdoor positioning and tracking, due to the inability of the satellite signals to penetrate the buildings. Compared with the RFID solutions used for the identification of the assets or personnel, RTLS can additionally offer the location data in 2D or 3D (Liu & Wang, 2004). A network of RTLS sensors are deployed throughout the area where the positioning will be carried out. The sensors are anchors or reference points from which the RTLS tags are detected and positioned. Wireless RTLS tags are attached to the moving objects or persons of interest in order to communicate with the sensors in the reference points through emitting signals which carry the identification of the tags.

For indoor positioning, considering the severe multipath effects and the low probability of Line-of-Sight, a number of wireless technologies has been proposed, such as: Infrared (Want et al., 1992), ultrasound (Priyantha et al., 2000), WiFi (Bahl & Padmanabhan, 2000; King et al., 2006), RFID (Ni et al., 2003; Bekkali et al., 2007), and UWB (Ingram et al., 2004).

Among these technologies, the Active RFID-based RTLS has received great attention for the continuous monitoring. Passive RFID tags do not have a power source and only transmit a signal upon receiving RF energy emitted from a reader in their proximity. Active RFID tags, on the other hand, are powered by a battery to automatically broadcast their signal. Active RFID tags achieve significantly larger readability range compared to the passive ones. Enabling detection of the more distant tags, active RFID is used for navigation in large-scale environments (Kulyukin et al., 2004). RFID has been jointly used with cameras to enhance the recognition of the objects and humans (Cerrada et al., 2007), (Jia et al., 2007). In (Zhang et al., 2005) passive RFID and cameras are used to avoid revealing the privacy of the authorized persons in the video streams. For this purpose, the face of the person who authenticates himself by means of the pre-assigned RFID card before entering the room is blurred in the video stream.

Locating techniques utilize the metrics of the received radio signals which are mostly based on the Angle of Arrival (AoA), Time of Arrival (ToA), Time Difference of Arrival (TDoA) measurements, or the Received Signal Strength (RSS) measurements from several reference points. The reported metrics are processed by the positioning algorithm for estimating the object's location. Three major families of the locating techniques are identified, based on the way the signal metrics are used by the positioning algorithm, namely: *triangulation*, *scene analysis*, and *proximity*. The accuracy of the estimated location is determined by the accuracy of the signal metrics and the complexity of the positioning algorithm (Papapostolou & Chaouchi, 2011).

Triangulation estimates the target's location based on the geometric properties of the triangles and has two derivations: *Lateration* and *Angulation*. *Lateration* measures the distance of the target from multiple reference points using ToA and TDoA. *Angulation* calculates the angles of the object's location relative to the multiple reference points using AoA. *Scene Analysis* refers to the algorithms which collect features (fingerprints, some location-dependent characteristics of a signal) of the scene and match the online measurements with the closest *a priori* location fingerprints in order to estimate the location of an object. *Location Fingerprinting* algorithms usually use pattern recognition and could be categorized into at least five types: probabilistic methods, *k*-Nearest Neighbor (*k*-NN), Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Smallest M-vertex Polygon (SMP). *Proximity* algorithms provide symbolic relative location information, e.g. located in the first office of the third floor, by employing a dense grid of antennas fixed in known positions. In this method, when an object is detected it is considered to be located near the sensor which receives the strongest signals from it (Liu et al., 2007).

Although in an ideal environment these measurements would be perfect and the calculation of the tag's position would be exact, the automatic real-time location is not error-free in the real-world. In metallic indoor environments, electromagnetic waves are influenced by reflection, scattering, attenuation, and diffraction. These effects cause the multipath problem in which the RTLS sensors receive additional spurious signals which are especially undesirable while measuring the Received Signal Strength Indicator (RSSI), AoA, and TDoA parameters. A benchmark is provided by (Liu et al., 2007) to investigate the performance of the indoor wireless *positioning systems* with the metrics which includes: *accuracy, precision, complexity, scalability, robustness* and *cost*. Although *accuracy* (or location error) is the most important requirement of the *positioning systems,* the location *precision* must also be considered as a measure of the robustness of the positioning techniques employed in the system. *Complexity* metric can be attributed to the hardware, software or computation, and operation factors. *Scalability* is an important requirement in the axes of: range of coverage for accurate locating, and density of the objects to be located per time unit. *Cost* metric also depends on many factors such as money, time, space, weight, and energy. Figure 2-2 highlights the competency of the UWB technology amongst the other wireless technologies for indoor positioning applications.

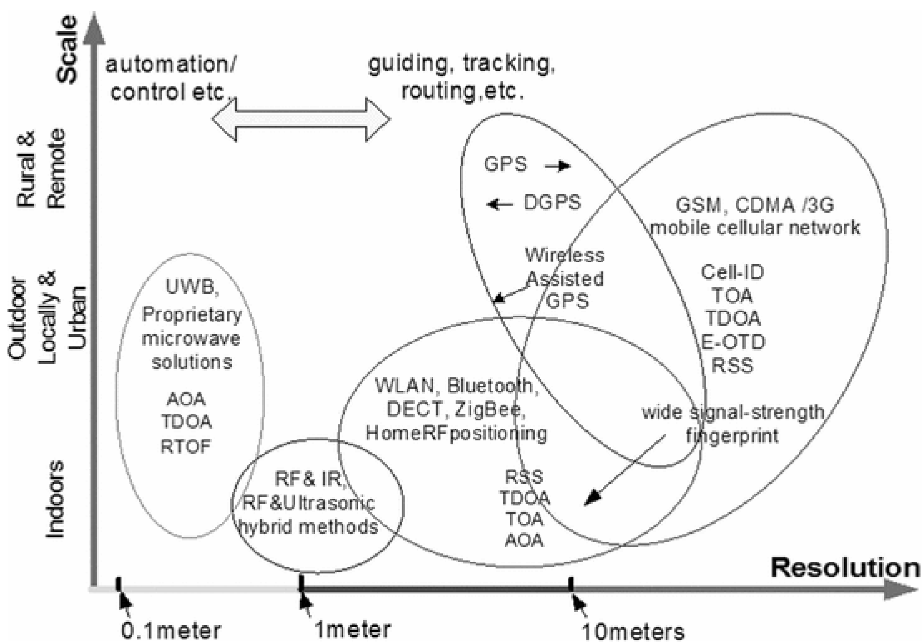


Figure 2-2 Outline of Current Wireless Positioning Systems (Liu et al., 2007)

Table 2-1 includes a summarized comparison of different RTLS types and highlights the UWB as a technology which can provide a continuous monitoring with a reasonable cost for up to 250 m in indoor areas. UWB's high accuracy together with its granularity level makes it suitable for the enterprises to leverage in their critical mission applications and tracking personnel.

Table 2-1 Comparison of Technology Types in RTLS Market (RTLS Market Overview, 2006)

RTLS Type	Wi-Fi	UWB	Passive RFID	IR
Cost	$$- $$$	$$$- $$$$	$	$$$
Power Requirement	High	High	Low. Magnetic Induction	high
Battery Lifespan (years)	3-5	3-5	Does not require a battery	4-7
Range	Indoors: 60-100 m Outdoors: 100-200 m	Up to 250 m	10-15 m both indoors and outdoors	IR spotter range is 15 m, convergence with RF up to 250
Granularity/ Resolution/ Accuracy	Some vendors suggest resolution can be 1-2 m but additional APs likely to be required. 5-10 m typically	Can be as low as in inches. Typically 1-5 feet	5-10 m typically	7-12 m typically
Continuous Monitoring	Yes beacon location available at real-time as tag is active based	Yes beacon location available at real-time as tag is active based	No tagged object will only be read and location provided in real-time when it comes into range of tracking unit	Yes beacon location available at real-time as tag is active based
Coordinate Position Provided	X and Y	X, Y and Z	X and Y	X and Y
Tracking Suitability	Best adopted for enterprises already having or planning to implement a Wi-Fi network as the tags leverage operations on the same network	Although can be adopted in any enterprise, UWB RTLS distinct advantage is enabling high level of granularity. It is expected to flourish in mission critical applications or for personnel tracking in dangerous environments	Passive RTLS will best suit applications that need a tracking solution for short to medium range applications. It is also suitable for end users with a limited budget but may require a basic but reliable tracking solution	Mostly applicable for indoors as IR in susceptible to NLoS

2.3.1.1 Ultra-Wideband (UWB) RTLS

According to the National Institute of Standards and Technology (NIST), compared to the other locating technologies, UWB has particular advantages for developing ad-hoc robust networks in hostile electromagnetic environments where jamming and interception attempts are assumed (Miller, 2003). UWB signal is defined to possess an absolute bandwidth of larger than 500 MHz or a relative bandwidth of larger than 20%. Commonly, Impulse Radio (IR) systems which transmit very short pulses with a high frequency are leveraged to implement the UWB systems (Gezici et al., 2005). UWB is a None-Line-of-Sight (NLoS) technology with a range of a few ten meters in the open spaces (Steggles & Gschwind, 2005). The bandwidth is perhaps the most prominent characteristic of the UWB communication systems, as the name declares. The wide bandwidth implies the potential of the UWB for high-resolution indoor positioning as its lower frequencies penetrate the walls, where the narrowband technologies fail because of the shadowing and multipath reflections. However, enjoying the wall or ground penetration, while achieving the same data rate, is totally scenario-dependent. In applications that wall or ground is considered as an obstacle, penetration is advantageous and useful (Miller, 2003). Moreover, thanks to the high bandwidth of UWB signals and consequently the much higher temporal and spatial resolution of them, AoA, ToA, and TDoA (in addition to the RSSI method) can be applied for precise object positioning in small areas such as a single room (Mandeljc et al., 2013). Compared to the conventional narrow-band RF systems, using the higher bandwidth (the narrower pulses) in UWB makes it possible to better distinguish the direct path signal from the reflections, as demonstrated in Figure 2-3.

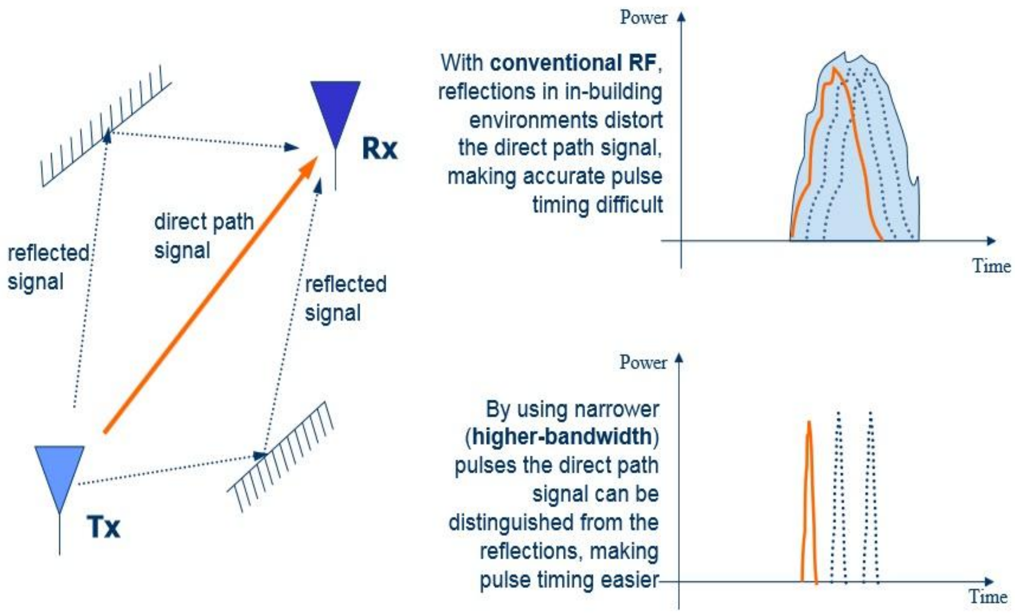


Figure 2-3 Overcoming the Multi-path Effect via UWB technology (Ward, 2007)

Besides, the wide bandwidth makes UWB efficient for the high-rate and multiple access communications, i.e. it can accommodate many users (high granularity). Nonetheless, the RTLS technologies whether UWB or conventional narrowband depend on the individuals' active cooperation to succeed. Hence, for intruder identification and tracking purposes, still surveillance cameras are the most appropriate technologies.

2.3.2 Video Surveillance and Visual Tracking

Surveillance generally refers to observing an area for activities, behavior or any other changing information by intention of managing, directing, security or safety protection of people or assets. Surveillance is performed in different forms by different means, depending on the intention and the target area. Video surveillance, aiming to ensure the physical security through detecting and tracking the intruders, has had a widespread use for many years. Video surveillance is common practice of observing an area by video cameras and analyzing the captured video for the purpose of human detection, abnormal activity detection, *intruder identification and tracking*.

2.3.2.1 Taxonomy and Evolution of Video Surveillance Cameras

There are different classifications for the surveillance cameras based on different characteristics. The surveillance cameras can be either: (1) *Analog* which turn the video signals to a format receivable by televisions, Video Cassette Recorders (VCRs) or monitors; or (2) *Digital* which digitizes the video signal by using a specialized encoder. Digital IP cameras offer the ability of expansion for much broader ranges and analytics as they can send and receive data via a computer network and the Internet. Both IP and analog cameras can transmit signals. However, the analog systems require different cables for transmitting the video, audio and alarm streams as well as the power supply cable, while they can all be included in cat-5 cable in an IP system. Unlike using the IP cameras, potential interference problems may occur in case of establishing a wireless surveillance system with the analog cameras. Besides, the signals captured by means of an analog camera cannot be encrypted. Moreover, long distance applications will be troublesome using the analog cameras. On the other hand, digital cameras offer better wireless reception, network design flexibility and simplicity, as well as easier remote access (NewJerseySolutions, 2013). Moreover, the analog cameras lack certain advanced features such as the digital zoom. An RS-485 cable is also required if an analog PTZ camera is to be controlled through the PTZ control panel. Typically a CCTV solution is based on analog cameras, Digital Video Recorder (DVR), control panel and a TV screen (Axis, 2008).

Different emerging technologies have gone through a long evolutionary path, resulting in the smart cameras. The evolution of smart cameras falls into three major paths: (1) Single Smart Cameras which focus on integrating the on-board processing with the essential sensing ability of the camera; (2) Distributed Smart Cameras (DSC) which introduce the collaboration of smart cameras in a networked camera system to solve the problems such as multi-camera tracking, behavior monitoring, and abnormal behavior detection; and (3) Pervasive Smart Cameras (PSC) which integrate the adaptability and autonomy into the DSCs to provide a service-oriented easy to deploy and operate a network that can adapt to the environmental changes and provide the customized services to the users (Rinner et al., 2008).

2.3.2.2 PTZ Cameras

PTZ cameras are a kind of stationary but rotating (horizontally and vertically) and zooming cameras which are being extensively deployed for wide-area surveillance. The unique characteristic of PTZ cameras, which makes them advantageous over other camera technologies, is enabling surveillance of a much wider area while being able to closely look at the points of interest and capture high degree of details (VideoSurveillance, 2013). Therefore, PTZ cameras can extensively contribute in simplifying the operation and management of the surveillance networks by decreasing the number of the required cameras for wide areas. The state-of-the-art PTZ cameras allow up to 360˚ horizontal rotation or panning and up to 220˚ vertical rotation or tilting (Axis, 2013). High zooming capability of the PTZ cameras is an enabler of the applications which aim at solving distant object identification and recognition, since they can capture very large number of pixels from the far distance objects (Choi et al., 2010). Moreover, IP PTZ cameras can be connected to central sophisticated video analytic software which implements further application-specific analytics (Oncam, 2013).

2.3.2.3 Challenges of Video Surveillance

The major challenging issues of video surveillance which have spawned extensive research endeavors are as follows:

- Cameras and recording devices are relatively expensive (Security Cameras, 2014).

- Without sophisticated hardware and software tools (which manage the video capture, analysis, archiving and retrieval), video surveillance is a time-intensive and labor-consuming task (Freeman et al., 2013).
- Camera footage grows drastically which makes it practically impossible to search for the events of interest within the huge volume of the video footage. By advent of the motion sensors, video footage growth is radically controlled and reduced; as the system can be set to record only when a motion occurs. Video analysis has been made easier by using automated software that organizes digital video footage into a searchable database (Lefèvre et al., 2003).
- Automated video analysis demands a high level of software intelligence and some heavy computations, because of the followings: (1) bad lighting and noisy video increases the complexity of object detection; (2) object identification becomes more complicated in the uniformed environments with similar clothing of the persons (Mandeljc et al., 2013); (3) locations must be translated from the image plane (pixels) into the real-world Cartesian coordinates; (4) person re-identification across the wide trajectories which involve multiple disjoint views is too complicated, time consuming and CPU-intensive due to the changes in the orientation, lighting and other features while the target is moving across the area; (5) Moreover, in the multi-camera systems, the cameras must be carefully calibrated in relation to one another in order to provide a consistent representation of the entire space.

In (Zelniker et al., 2008) a framework has been proposed for object tracking across camera views, based on the unsupervised learning and probabilistic abnormality inference. In (Prosser et al., 2008) a novel method is proposed for multi-camera object association, based on adapting a learned inter-camera illumination mapping function without the need for a manual training stage and using the new foreground objects. Another work tackling difficulty of the global anomaly detection by (Loy et al., 2009) proposes a novel approach based on learning the time-delayed dependencies between the activities across the camera views. In this model, different nodes represent the activities in different semantically decomposed regions and the directed links between those nodes encode the causal relationships between the activities. The work of (Hongyu et al., 2011) proposes a network-based algorithm for multi-camera abnormal activity detection. Another method is also proposed by (Chen et al., 2011) for multi-camera person tracking which extracts the persons' images from the captured video and labels those images with the owners' color vectors. An algorithm for automatic optimum camera placement is proposed by (Yabuta & Kitazawa, 2008) to minimize the required number of cameras for efficient monitoring of a complex area. This algorithm calculates the camera locations to cover all the essential regions and as many other less important regions as possible, through scene segmentation into regions, weight association to the regions, and visibility test between each two regions. This work formulates the problem as a set covering problem and uses CPLEX solver for it (IBM, 2013).

On the other side, PTZ cameras offering a wide dynamic FoV have gained increasing attraction for research to avoid the problems of modeling the behavior of the objects in a network of cameras. For example, the work of (Keni et al., 2007) adopts a single PTZ camera for tracking a person moving freely in a room. However, due to the frequent changes in the camera orientation, motion and color invariant, the frontal face detectors are used in each video frame in order to initialize and continuously update the person models. Therefore, PTZ cameras introduce other computational complexities as listed below:

- The orientation changes and the movement speed in the PTZ cameras make video analysis more difficult.
- While most of the CV algorithms work with the Cartesian coordinates, the PTZ cameras' movement control takes place in the spherical coordinate system. Therefore, a model is required for fast and accurate mapping of the pixel to the pan and tilt rotation angles of the PTZ camera. The dynamic rotation angles of the camera must be calculated from the extracted world coordinates of the target object via video frame analysis. The work of (Sankaranarayanan & Davis, 2011) proposes a real-time automatic model with minimum necessary formulation for mapping any point on the focal plane of a PTZ camera to its corresponding pan-tilt orientation based on the current pan-tilt value of the camera.
- While the PTZ camera is zoomed in a small area of its FoV to capture faraway details, the wider picture is not recorded and vital information could be lost.

2.3.2.4 Video Processing Techniques and Algorithms

The captured video in security surveillance may be watched in real-time or archived for future search, review and analysis. Without appropriate search techniques, the lengthy video footage would only leads to laborious and time-intensive searches, reviews and analysis. In fact, the cameras are only the observing eyes and an automated security system needs a processing brain to analyze the rich spatiotemporal video data in order to decide about the appropriate reactions to the events. Otherwise, the surveillance system becomes an “after the fact” forensics tool. Even the smart cameras, with on-board processing, are not capable of advanced application-specific analysis to deduct the environmental incidences and support the decision making process.

Common Video Codecs for Surveillance

One major challenge of digital video is the huge amount of data that need to be transferred and stored, different image and video compression techniques have been developed. Different compression techniques provide tradeoffs between many factors such as: resolution, color depth, frames per second, etc. Therefore, for different applications depending on the source content and target display, available CPU, and latency tolerance, a suitable codec (pair of video encoding and decoding algorithms) must be adopted. Surveillance digital video recorders often support encoding multiple channels simultaneously and performing intelligent transmission based on image analysis features. The International Standards Organization (ISO) is focused on consumer applications and has defined JPEG standards for still image compression and MPEG standards for moving pictures compression (Golston, 2004).

JPEG (Intra-frame compression) was the first widespread image compression standard (Peennebaker & Mitchell, 1993) which can achieve 10:1 compression without introducing serious effects on the image. Therefore, JPEG standard is not optimized for video in terms of bandwidth and storage providence and now is widely used for Internet web pages and digital still cameras. However, Motion JPEG uses JPEG for independently coding the frames of a video sequence. MPEG-1 was the first video codec developed by ISO which generates effectively 25:1 compression ratio. MPEG-2 was developed for digital televisions and became the most successful video compression algorithm so far with a ratio of around 30:1 compression. MPEG-4 was initiated as a follow-on to the success of MPEG-2 (ISO/IEC, 1999). MPEG-4 has higher complexity and coding efficiency than MPEG-2. A major breakthrough happened with the introduction of H.264 standard jointly promoted by ISO and International Telecommunication

Union (ITU). H.264 delivers a significantly better compression ratio around 2x reduction (60:1 compression) of bit rate versus MPEG-2 and MPEG-4 (Golston, 2004).

To transform the surveillance system into an intelligent active security agent, apart from the installation of the smart cameras and the transfer and storage of the compressed video, some sophisticated video analytic tools are required to perform the appropriate CV functionalities. Video surveillance has spawned large research projects in the fields of CV, pattern analysis, and artificial intelligence for detecting and tracking moving objects, classification of the detected objects, motion analysis, and activity understanding (Collins et al., 2000). Different tools have been developed which can detect the objects and persons in the video, analyze and identify the behaviors, send timely alerts to security officer, and facilitate the process of video content retrieval. The major categories of the video analysis techniques are introduced as follows.

Motion Detection

Background subtraction is a widely used approach for detecting the moving objects in the videos coming from the static cameras. This technique is based on determining the difference between the current frame and a reference frame called the "background model". The background model must be a representation of the scene without any moving object. Several methods have been proposed for effectively estimating the background model from the temporal sequence of the frames. A review and performance analysis of the background subtraction methods can be found in (Piccardi, 2004).

Object Detection and Localization

The state-of-the-art Probabilistic Occupancy Map (POM) algorithm is used for the frame-by-frame object detection and localization (Fleuret et al., 2008). The POM algorithm discretizes the ground plane of the region of interest into a rectangular grid of a specific resolution, typically 20 cm, and iteratively estimates the probability of the occupancy in each cell of the grid, based on the input binary images. The binary images are usually obtained through the background subtraction technique. For each frame, a set of anonymous detections is obtained including the cells with the high probability of occupancy. These sets can be linked into the trajectories, using the *k*-Shortest Path (*k*-SP) tracker (Berclaz et al., 2011).

Multi-Object Tracking

The classic multi-object tracking formulations emphasize the data association problem (Cai et al., 2006; Choi & Savarese, 2010; Israd & MacCormick, 2001). Some approaches (Jiang et al., 2007) rely on the manual track initialization and/or a fixed predetermined number of the objects, which are not realistic assumptions. Kalman filtering is an efficient multi-target tracking method which is suitable for the real-time applications as long as the number of the targets remains small (Black et al., 2002; Mittal & Davis, 2003; Iwase & Saito, 2004; Xu et al., 2004; Magee, 2004). By increasing the number of the targets, the identity switch becomes frequent and difficult to correct due to the recursive nature of the Kalman filtering method. The mean-shift algorithm used by (Wu & Nevatia, 2006) also suffers from the same problem. Particle filtering, applied in (Okuma et al., 2004) and (Du & Piater, 2007), can address some of the limitations of the Kalman filtering approach through exploring some hypotheses.

A more general spatiotemporal grouping framework is proposed by (Pirsiavash et al., 2011) which treats the multi-object tracking problem as a cost function (which requires estimating the number of the tracks, as well as their birth and death). A family of efficient, greedy but globally optimal multi-object tracking algorithms is described in the paper, including the common approaches for estimating the number of the unique tracks and the spatiotemporal extent of each track.

The *k*-SP algorithm determines the *k* shortest paths between a pair of nodes in a network (Martins & Pascoal, 2003). Multi-object tracking can be performed through frame-by-frame object detection and then linking the detections across the frames. However, the linking step results in a difficult optimization problem for the multi-target tracking applications. As proposed in (Berclaz et al., 2011) the complexity of the multi-object tracking problem can be reduced to a tiny fraction of the original LP problem through formalizing the object motions as the flows along the edges in a graph of the spatiotemporal locations and then applying the *k*-SP algorithm. The resultant algorithm is much simpler than the other state-of-the-art approaches and its convexity assures that the global optimum can be found.

Video Image Retrieval and Analysis Tools (VIRAT)

The VIRAT program is funded by the Information Processing Technology Office (IPTO) of the Defense Advanced Research Projects Agency (DARPA). This program aims at content-based video searching by organizing and storing the video footage within a searchable database and development of the video analysis tools that can retrieve the video frames which include the objects or activities of interest. Conventional video search is limited to the metadata queries, manual annotations and fast-forward examinations. Content-based video retrieval eliminates the need for the manual fast-forwarding searches, metadata queries, or adding annotations to retrieve the content of interest. By result, the human operator will be able to questioning the software agent to find all the footage where for example three or more persons are standing together. Content-based video retrieval, on the other hand, firstly requires segmenting the video stream into clips, then extracting the features of each clip, and finally retrieving the clips that match the similarity/distance metric of the query. Altogether, the automated video analysis and the intelligent alerting will take off a large portion of operators' burden (DARPA, 2014).

PoS Transactional Video Indexing

Indexing the video captured in the Point of Sales (PoSs) with the transactions made at different timestamps is a state-of-the-practice example which enables searching the video based on the suspicious transactions which can reveal the face of suspected person and the scene and ongoing activities at that time. Software tools are developed to integrate with the existing DVRs and PoSs to generate intelligent alarms and enable event-based video search and retrieval to help reduce shrinkage and property damage. The Corporation de Technologie d'Intelligence et d'Automatisation Internationale (CTIAI) is a center specialized in security analytics which develops tools for retails sales, banks, etc. CTIAI has a tool for retail stores PoSs which synchronizes and indexes the PoS transactions and the recorded video. Its GUI allows searching for the specific transactions, data and time, product description, bar code data, employee code, dollar value, or other customized detail which can be included in the transaction information extracted by the system (CTIAI, 2013).

2.4 Data Fusion Techniques for a Multisensory System Design

According to the Joint Directors of Laboratories (JDL), data fusion refers to a "multi-level, multi-faceted process which handles the automatic detection, association, correlation, estimation, and combination of data and information from several sources" (White, 1991). The data sources could be homogenous or heterogeneous sensors, databases, information gathered by human senses, etc. (Elmenreich, 2002). Data fusion aims at achieving higher quality information by using redundant, complementary, or timelier data (Luo et al., 2002). Multi-Sensory Data Fusion (MSDF) is a well-known technique for the synergic combination of the sensory data from multiple sensors to overcome the physical limitations of the individual sensing systems and to provide more reliable and accurate information for inference about a physical event, activity or situation (Elmenreich, 2002; Luo et al., 2002). MSDF requires interdisciplinary knowledge of signal processing, artificial intelligence, probability and statistics, etc. MSDF has application in robotics, biomedical systems, equipment monitoring, remote sensing, transportation systems, and so forth. MSDF algorithms can be broadly classified into: estimation methods, classification methods, inference methods, and artificial intelligence methods.

Estimation method is one of the simplest data fusion methods which takes a weighted average of the redundant information as the fused value. Despite the suitability of the weighted average method for real-time processing of low-level data, the Kalman Filtering is predominantly preferred because of its optimal (fused data) estimates in a statistical sense, with nearly the same processing requirements. The multidimensional feature space can be partitioned into some distinct sub-spaces, using a similarity measure which is determined by the classification methods, and each observation must be compared against *a priori* classes. Cluster analysis is a powerful multisensory data classification tool. Learning Vector Quantization (LVQ), K-means clustering, and Kohonen Feature Map are unsupervised or self-organized learning algorithms which can be used for classification. Baysian inference updates the probabilities of the alternative hypotheses based on the observational evidence. New information is used for updating the *a priori* probability of the hypothesis. Table 2-2 contains the classification of the MSDF algorithms according to (Luo et al., 2002).

Table 2-2 MSDF Algorithms Classification (Luo et al., 2002)

Estimation Methods	Non-recursive: • Weighted Average • Least Squares Recursive: • Kalman Filtering • Extended Kalman Filtering
Classification Methods	• Parametric Templates • Cluster Analysis • Learning Vector Quantization (LVQ) • K-means Clustering • Kohonen Feature Map • ART, ARTMAP, Fuzzy-ART Network
Inference Methods	• Baysian Inference • Dempster-Shafer Method • Generalized Evidence Processing
Artificial Intelligence Methods	• Expert System • Adaptive Neural Network • Fuzzy Logic

There are some challenging issues regarding data fusion which arise from: the data to be fused, imperfection and diversity of the sensor technologies as summarized in Figure 2-6. The fusion algorithm must treat any of them with special care to mitigate its effects (Khaleghi et al., 2011).

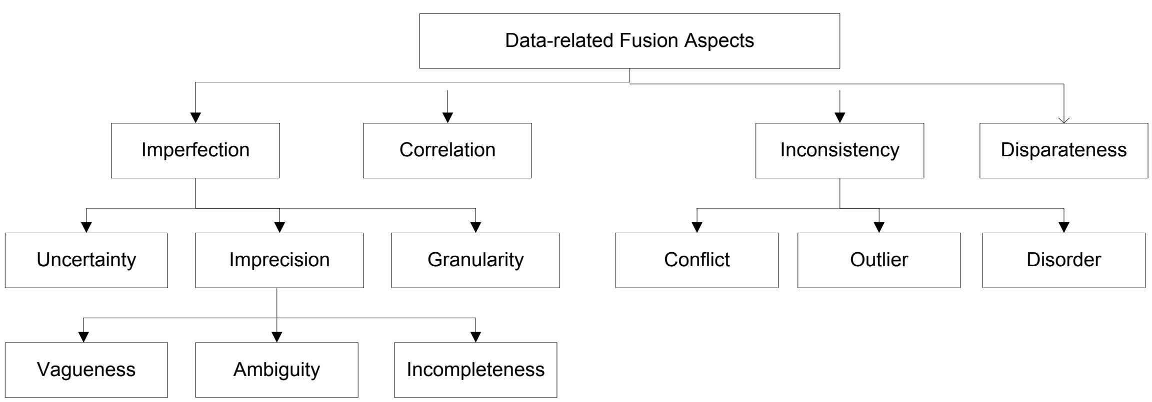


Figure 2-4 Data Fusion Challenges (Khaleghi et al., 2011)

2.5 Related Work

To the best of our knowledge, the task of *Automatic Intruder Identification and Tracking* has not been fully addressed in researches. However, the involved sub-tasks such as human detection, identification, locating, and tracking by identification have a long literature in both of the RF-based and CV-based approaches. The state-of-the-art approaches focus on fusing data from both of the RTLS and Video Surveillance Systems for various combinations of the abovementioned tasks in security, safety or other applications. The present review roughly divides the related literature into three groups: (1) using UWB; (2) using PTZ surveillance cameras; and (3) using the fused radio-video data from the aforementioned technologies.

2.5.1 Real-time Identification and Tracking with UWB RTLS

RF-based tracking, compared to the challenging and heavy CV-based face recognition and person re-identification along the trajectories, is advantageous in terms of accurate real-time identification (Mandeljc et al., 2013). However, UWB technology cannot provide as much locating accuracy as the CV-based locating, due to the scattered behavior of the radio signals.

Advanced Baysian filtering techniques are proposed by (Denis et al., 2005) for the UWB tracking systems to deal with the biased **ToAs** in the NLoS indoors. An accurate **TDoA**-based UWB localization approach is developed by (Zhang et al., 2006) which achieves a sub-cm accuracy by suppressing the indoor multipath effects. The developed system is claimed to be superior to the Frequency Modulated Continuous Wave (FMCW) systems and in compliance with the FCC UWB.

The excellence of the UWB technology for indoor positioning makes it suitable for numerous applications such as elderly care, as employed by (Stelios et al., 2008), in order to provide them with timely and valid medical services. Indoor robot navigation is addressed by (Krishnan et al., 2007) using UWB positioning. An underground localization system, composed from the UWB technology and ANN, is presented in (Taok et al., 2009) which uses the location fingerprinting technique to avoid NLoS, reverberation, and multipath effects of the AoAs of the signals.

2.5.2 Video Surveillance and Tracking with PTZ Cameras

Video surveillance and analysis, unlike the RF-based approaches, enable not only for human detection and positioning but also for visual analysis of their behavior for detecting the abnormalities and reconstructing the events' scenes. Moreover, video surveillance enables unobtrusive detection and recovery of the trajectories without the need for wearing the RF-based tags. Therefore, the untagged (unauthorized) persons can be captured through video surveillance along their trajectories. Another limitation of the RF-based positioning which is improved by the video surveillance is the update rate which is only limited by the camera's frame rate while the update rate of the RF-based positioning is more limited due to the nature of the radio signal and the time-slotting, and is decreased as the number of the tagged persons increases (Mandeljc et al., 2013).

Conventionally, multi-target tracking approaches depend on the use of multiple cameras with overlapping FoVs and can be reviewed within two groups: (1) Detection-by-Tracking which relies on sequential detection, and usually apply the Kalman filter (Xu et al., 2004; Iwase & Saito, 2004) or the Particle filters (Otsuka & Mukawa, 2004; Kristan et al., 2009); (2) Tracking-by-Detection which employs a robust frame-by-frame detection (Fleuret et al., 2008; Khan & Shah, 2009) and the global optimization methods for tracking (Berclaz et al., 2011).

The trackers, used for Detection-by-Tracking, are causal and only consider the previously processed frames. Although this makes them suitable for real-time tracking, the trackers may result in irrecoverable errors when the detection fails in a frame or when the detections are incorrectly linked. These errors are propagated in the subsequent frames and eventually cause in tracker failure. This problem is mitigated in the Tracking-by-Detection approaches. However, relying on the identity propagation with none or limited appearance-based validation, Tracking-by-Detection is prone to propagation of the identity switches along the trajectories. To reliably distinguish the individuals, a much larger assortment of visual cues is required (Mandeljc et al., 2013). Another approach for reliably distinguishing the detected individuals in the video is data fusion and augmenting the anonymous video detections with the real-time RF-based identity information. For example combined RFID and video information are used for human behavior analysis by (Hsu et al., 2006).

To avoid complexity in large-scaled surveillance networks, the PTZ cameras must be employed to obtain higher degree of continuity and accuracy for people tracking. Several research works such as (Luca et al., 2005) and (Hampapur et al., 2003) experimented high-resolution face image acquisition by the combined use of the PTZ and static cameras. The research of (Choi et al., 2010) proposes a novel calibration method for coordinating the views of the static and PTZ cameras, as well as a PTZ control framework for smooth face tracking.

2.5.3 Fusing Radio and Video Data

With respect to the complexities of video analysis for human Tracking-by-Identification and the cluttered behavior of the RF-based positioning, sensor integration and data fusion is promising in terms of increasing the performance and accuracy. Compared to the CV-based human Tracking-by-Identification, a hybrid system will be improved using the accurate real-time RF-based identification. Unlike the CV-based detection which suffers from false positives and negatives, RF-based detection and identification is very reliable and practically without no false positive or negative. Furthermore, RF-based identification is real-time and eliminates the need for heavy video analysis. On the other hand, compared to the RF-based positioning, which is limited to the tagged persons and is cluttered across the radio coverage, the hybrid system can achieve more accurate localization by using CV techniques (Mandeljc et al., 2013).

To the best of our knowledge the closest work to our research has been done by (Dibitonto et al., 2011) to fuse UWB and video data for more reliable and robust tracking services and more accurate context understanding. They leveraged UWB RTLS and a fixed video camera and applied a simplified version of the cross-tracking algorithm (Collins et al., 2001) for automatic association of the trajectories observed by the two sensor technologies. They carried out some tests to check the feasibility of this automatic data association and the results demonstrated a coarse matching level for their prototype system. This result is mainly due to subsystems' bias and the matching algorithm. Connecting the system to the other sensors, as a depth sensor, can present a better understanding of the ambient observed. They also did not evaluate locating accuracy; for this purpose, different metrics are used by different authors as follows.

Assessment of 2D localization error distribution along with any relevant dependencies is suggested by (Hightower & Borriello, 2001). However, this information does not include information about missing and phantom detections. Among the CV-based approaches, the work of (Fleuret et al., 2008) counts the number of the detected individuals and the number of false detections in the video frames to report precise error rates. In (Delannay et al. 2009), sport players recognition is validated for the videos coming from a set of loosely synchronized cameras in order to determine the number of false and missing detections.

A personnel tracking system which integrates active RFID and a web camera is proposed by (Wang & Cheng, 2011). In this system, background subtraction method is used for detecting the personnel in the captured video. The coordinates of the detected personnel are calculated and cross referenced with the RFID positioning information to add the IDs of the detected persons. Another proof-of concept application which fuses UWB and CV-based location data and is capable of monitoring and tracking individuals in multiple areas is named RVid. It also includes Intelligent Face Logger which provides the support for recognizing the individuals (Tronci et al., 2013).

In (Mandeljc et al., 2013), a hybrid Tracking-by-Identification system is developed comprising an UWB RTLS and multiple calibrated and time-synchronized video cameras with overlapping FoVs. The UWB tags are positioned with combination of the AoA and TDoA measurements, each tag having the update frequency 4.6 Hz. The state-of-the-art POM algorithm (Fleuret et al., 2008) is applied for frame-by-frame person detection and localization, and *k*-SP algorithm (Berclaz et al., 2011) is applied for linking the independent detections into the trajectories.

2.6 Summary and Conclusions

The literature review of this chapter focused on the state-of-the-art technologies for real-time, accurate, and complementary data collection from wide indoor areas. The technologies introduced in this chapter, which are BIM, UWB RTLS and PTZ video cameras are integrated in the design of AIITS with maximum accuracy, reliability and added-value. BIM can assist for optimum system deployment, better access control to the restricted areas, localization validation and so forth. Real-time identification by UWB enables distinguishing the tagged persons from the untagged persons and PTZ video camera enables tracking any person of interest along his/her trajectory.

We found numerous researchers have undertaken experiments in fusing the radio and video data from various sensor technologies for different purposes such as: workers’ safety, elderly care, employee tracking and so forth. However, none of them aim at increasing security automation and reliability. Also, BIM is not used in the design, deployment and operation of the security systems as a source of geospatial insight over the static objects of the environment.

With respect to the labor-demanding and time-consuming task of security surveillance, and the shortcomings of the present systems, we propose a simple but capable multi-sensory design for automation of the wide indoor areas security. We believe the joint use of the RF-based human identification and tracking with the CV-based human detection and locating can reliably accomplish automatic intrusion detection and intruder identification. The strength of the above-mentioned technologies for indoors, in addition to the richness of the BIM can be complementary for accomplishment of the *Automatic Intruder Identification and Tracking,* by applying an accurate data fusion method.

CHAPTER 3 REQUIREMENTS ANALYSIS AND PROPOSED METHODOLOGY

3.1 Introduction

Ensuring indoor security, as discussed in Section 2.1, necessitates modeling, perception and analysis of different environmental aspects via appropriate means. Integrated frameworks are promising to overcome the technical limits of the individual technologies and to increase the automation level of the isolated security solutions. In this regard, an appropriate data fusion model is required to accurately correlate the results of the components and to infer the most suitable response.

This chapter includes the requirement analysis and the technology adoption and integration criteria in Section 3.2 for the intruder identification and tracking approach which is introduced in Section 3.3 and is elaborated in Section 3.4. A robust design which is easily scalable for wide indoor areas is proposed in Section 3.5 based on the discussed criteria and requirements for the AIITS. This chapter concludes with summary and conclusions in Section 3.6.

3.2 Requirement Analysis and Technology Adoption Criteria

"Good applications are those that achieve an adequate equilibrium between system requirements, technological advantages, and associated costs" (Muñoz et al., 2009). As discussed in Section 1.2, the objective application of this thesis, intruder tracking, is divided into three functions including: (1) intrusion detection, (2) intruder identification and locating, and (3) intruder tracking by PTZ camera. The requirements of a reliable AIITS which implements the above-mentioned functions are discussed in the following sections in order to determine the technology adoption and integration criteria which yield a good system design.

3.2.1 High-Level Requirements of AIITS

Regardless of our proposed approach based on the joint use of UWB RTLS and PTZ camera, any architecture for an applicable AIITS must satisfy the following requirements:

(1) A person may be authorized to access a private room only during a limited period and may not be authorized for all the private spaces in the building. There must be some well-defined and accurate mechanism to exert the space constraints. The building spaces' usage (public or private) and ACL must be defined for the AIITS to allow authorization control.

(2) For each private space, the coverage of the RTLS and the camera must be examined, measured and documented. The IDs of the sensors and cameras must be bounded to the covering space for XQuery processing and video retrieval.

(3) Intrusion detection is only achievable in the areas covered simultaneously with the RTLS and video camera. Also, for identifying and locating the intruder, who is an untagged person, the camera must cover the whole area that the RTLS sensors cover. In the partially overlapped areas, the AIITS functionalities cannot be accomplished as they are based on comparisons and inferences on the fused radio and video data.

(4) The authorized persons must be registered with the RF-based tags which they are supposed to carry during their access to a private room. The usability, as an important factor in the acceptance of the AIITS, must be considered in the tagging approach. The size and weight of the tags must not be a negative factor in the system's usability and should not disturb people's mobility.

(5) Real-time data collection is an essential prerequisite of any real-time application. To ensure no data loss occurs during the monitoring, density of the tags must be controlled to not overpopulate a RTLS cell. Otherwise, data collection would be problematic when all the tags are present because of the limited update rate of the cell.

(6) The update rate of the RF-based tags (Hz) must be (optimally) equal or as close as possible to the video update rate (fps) for correct data correlation during data fusion. The maximum video frames which can be recorded vary depending on the camera and its different codec types, for example Sony ER-580 provides maximum of 30 fps for H.264 codec, 20 fps for MPEG-4 and 16 fps for JPEG (Sony SNC-ER580, 2012). The RTLS platforms usually have controllable update rates for the tags. For example, Ubisense provides as fast as 40 Hz and as slow as one update every 14 minutes (Ubisense RTLS, 2012).

(7) AIITS must be designed for 24/7 continuous operation, near real-time intrusion detection with optimally no false positives and negatives. It should be crash tolerant and should have recoverable services. Also, the AIITS must provide the possibility of logging the RTLS data and recording the captured video.

(8) Storage management is important because of the continuous surveillance. Some policies must be defined for the AIITS to control the size of storage, e.g. deleting the video footage after 5 days and the RTLS logs after 1 month.

3.2.2 Environment Modeling Requirements

As the focus of this research is not on the building modeling and only a data rich building model is required, an standard and easy to develop approach must be adopted which is easily updatable to reflect the area changes. The overall requirements are listed as follows:

(1) **Simulation options:** A digital model is required which allows simulating the sensors' coverage by examining different views.

(2) **Easy information retrieval about the spatial relationships:** An object-oriented modeling and data storage must be employed to enable easy information retrieval about the spaces, their constraints, and their installed equipment. This is also helpful for linking the ACLs to the corresponding building spaces in the model.

(3) **Easy model updating:** the building model should not be static or hard to manipulate for facilitating simulation, updating, and future system scaling.

Advantages of BIM in the Context of AIITS

The investigation of different building modeling technologies in Section 2.2 highlighted BIM's strength for satisfying the requirements of the AIITS by offering the following advantages:

(1) **Spatial relationships vs. only geometry:** Encompassing the depth as the 3rd dimension and the object-oriented modeling, which contains the relationships of the model elements, BIM provides better representation of the monitoring environment and gives a better insight for the enhanced decision making. This is useful in choosing the optimum installation points for the sensors and cameras which achieves the maximum-coverage arrangement scheme, validating sensory location data, directing the PTZ camera for target tracking, and routing the security officers through the shortest path to the located intruder.

(2) **Different views and simulation options:** The BIM allows for viewing and zooming in/out the area from different angles. The BIM enables simulating the sensors' coverage; for example, inserting a virtual camera in the desired points within the model to simulate its FoV without physical tests.

(3) **Easy information retrieval about the spaces and facilities:** BIM enables various data inquiries about the modeled elements because of its object-oriented nature. The BIM creates and maintains a database storing the geometry, relationships and attributes of the model elements. This is advantageous for query processing and applying the usage constraints to the AIITS (e.g. public vs. private space and its ACL).

(4) **Interoperability and cooperation:** Using the open standards and non-proprietary, neutral data structures such as Industry Foundation Classes (IFCs) and aecXML, BIM widely supports the interoperable projects among the building stakeholders. Therefore, it is advisable to align the security with other BIM-based systems of the building, such as energy analysis, etc.

(5) **Access to the most updated space model and information:** Since the plan and/or the usage constraints of the spaces may get changes after establishment of the AIITS, using BIM ensures the AIITS has access to the most up to date environment model as its most essential awareness source. Using BIM, as a shared knowledge resource which is frequently exchanged between stakeholders, guaranties consistency of the AIITS with the surveillance area.

3.2.3 RTLS Requirements

As the proposed AIITS is expected to be deployed for wide indoor Tracking-by-Identification, a technology must be adopted as its RTLS which has the following characteristics:

(1) **Accuracy:** for the location-aware services to be reliable, the essential requirement is accuracy. UWB, as the most accurate indoor positioning technology, requires different number of sensors for different location calculation methods in order to provide accurate

results. Table 3-1 shows different combinations of methods and sensors together with their positioning results.

Table 3-1 Sensor Requirement for Accuracy of UWB RTLS (Ward, 2007)

System Architecture	Number of Sensors Detecting Tags	Other Information Required	Result
TDoA only	4+	None	3D position
TDoA + AoA	2+	None	3D position
Single AoA	1	Known height of a tag	2D position

(2) **Real-time response:** If the number of tags in a RTLS cell exceeds a specific limit for a pre-determined update rate, locating latency would appear. Also, applying the filtering algorithms to remove noisy location data or to respond steadily results in locating latency.

(3) **Range:** for the indoor applications, including security Tracking-by-Identification, the RTLS must be usable between fixed and potentially mobile receivers and be functional in metallic and furnished environments with populous interference, which cause the signal multipath problem. UWB, being immune to the multipath effect, offers the highest accuracy for the maximum indoor range.

(4) **Visibility:** to increase the visibility between the tags and sensors and to decrease the errors, placement of the sensors must avoid the blind spots and achieve the optimum coverage.

(5) **Continuous monitoring:** As continuous and real-time location data is required for object detection and tracking, active RF-based tags must be adopted so that they automatically broadcast radio signal using their internal battery power.

(6) **Granularity:** since the tracking applications mainly address numerous targets, a finely granular sensor network must be adopted for the RTLS to accommodate many targets, as one or potentially several tags could be assigned to a single target.

(7) **Node density:** Since the used commercial RTLS uses only one channel with time-division mode to communicate with the tags, only one tag can be located at a time in each RTLS cell (Zhang, 2010). Therefore, it is not efficient to overpopulate a cell by tags and the density must be controlled according to the application's required update rate.

(8) **Size and weight:** as the tags are supposed to be carried with the mobile assets or persons, they must be as small and light as possible not to be annoying or hindering the smooth operation.

(9) **Equipment and Installation cost:** must be justifiable with respect to the importance of the application which the RTLS is deployed for and the value of the target assets.

(10) **Interoperability:** as the RTLS may need to interface or coexist with other wireless communication technologies, it must stand against the other signals in the RF spectrum.

3.2.4 Video Surveillance and Analysis Requirements

The general and AIITS-specific requirements of video surveillance and analysis are listed as:

(1) **High resolution and capture rate:** For CV-based video analysis, it is required to capture video with high resolution and update rate so that the detection and recognition algorithms have rich data to analyze. The resolution is essential for accurate analysis and the high update rate provides redundant data (with tiny changes in two consecutive frames) for result optimization by noise reduction.

(2) **Wide FoV:** The FoV is an important factor as it determines the size of a surveillance network in terms of the number of the required cameras to cover a wide area. As the complexity of management, handover, and analysis radically increases for large size camera networks, it is critical to adopt the cameras offering wide FoVs.

(3) **Camera orientation:** Inclination of the camera increases the complexity of video analysis, as the image plane and a target scene will have angular position to each other. As the 2D image plane does not contain sufficient data to represent the real-world scene, additional information is required to construct a 3D scene model for some of the analysis, e.g. through stereo vision. However, for the sake of simplicity and considering the optimum possible FoV, the camera can be mounted on the ceiling with a downward vertical orientation. This orientation is not unrealistic for our AIITS application which does not require any processing further than detecting the human bodies, locating and counting them. Moreover, this top view, besides yielding in wider FoV (for high ceilings) and simplified video analysis, prevents video occlusion. Video occlusion, which means covering a target (partially or entirely) by another moving target, cannot happen in ceiling camera installation that provides top view of the area.

(4) **Image plane vs. world coordinate system:** In addition to the inclination complexity which is relaxed by using 2D coordinates for the AIITS, the calculated coordinates by the CV object locating algorithms are in pixels (not the real-world measuring units, e.g. cm). Therefore, a coordinate translation step is required to map the video locations (pixel coordinates) into the real-world coordinate system, i.e. *BCS* (Mandeljc et al., 2013).

3.2.5 Technology Adoption and Integration Criteria

To provide the necessary real-time data for the AIITS, this research proposes integrating the RTLS and CV technologies. However, with respect to the diversities of the aforementioned technologies and also the shortcomings of the similar frameworks in the related researches, further comparison is carried out in this research to adopt the most competent technologies which demonstrate the highest interoperability. Dividing the RTLS technologies into the traditional narrow-band vs. UWB, and the surveillance cameras into the fixed and PTZ, six possible combinations are investigated in Table 3-2 for usage in AIITS.

Table 3-2 Technology Adoption Review

			Camera Types		
			Fixed Camera	**PTZ Camera**	**Fixed + PTZ Cameras**
			Continuous observance over a limited FoV (Suitable for lossless-data applications)	Wider FoV achievable in exchange of loss of continuity during camera rotations (Suitable for target tracking & behavior analysis)	Continuous observance over critical & confined areas (e.g. entrance) with fixed cameras + wide-area on-demand surveillance with PTZ cameras
RF-based Technologies For RTLS	**Traditional Narrowband RFIDs**	Tag detection & identification + chokepoint locating	Steps 1, 2 and 3 are infeasible across wide areas	Although PTZ camera is available for performing step 3, steps 1 and 2 are infeasible across wide areas	Although PTZ camera is available for performing step 3, steps 1 and 2 are infeasible across wide areas
	UWB	RFID capabilities + precise wider range locating	Step 3 is doable only within a very limited area	Steps 1, 2 and 3 are feasible	Fixed camera is assigned to collaborate with UWB in steps 1 and 2. PTZ camera is assigned to step 3

The traditional narrowband RFID, having the multipath problem (as a NLoS technology) and therefore inaccurate positioning, is not promising for wide indoor tracking applications. Instead, the narrowband technology is suitable for access control of the chokepoints (RFID tag identification for granting access to the authenticated person). By contrast, the UWB, being more immune to the multipath problem and able to achieve high accuracy in positioning, is suitable for wide indoor identification and tracking applications rather than getting confined to the chokepoint access control. Figure 3-1 illustrates the capabilities of different positioning technologies in terms of accuracy versus coverage. As the figure shows, the UWB offers the best equilibrium of accuracy and coverage while an ideal technology is defined to be a system that can achieve the precision of less than 0.3 m for coverage of more than 100 m. Therefore, the

UWB-based RTLS is superior to the traditional narrowband RFIDs to meet the AIITS's coverage and accuracy requirements.

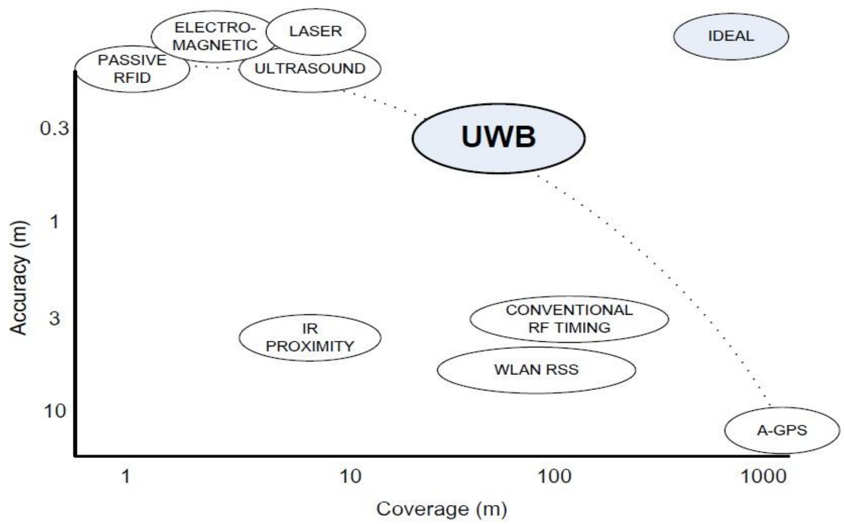


Figure 3-1 Comparison of Locating Technologies (adapted from Ward, 2007)

On the other side, visual intruder tracking across a wide area as another AIITS requirement is doable either by several fixed cameras or considerably less number of PTZ. A single fixed camera has a very limited ability for tracking a mobile target along his trajectory as it cannot pan or tilt. In contrast, a single PTZ camera can replace multiple fixed cameras by offering the rotation capability and achieving a considerably wider FoV. Therefore, the PTZ cameras are preferable for the AIITS as the decreased number of the surveillance cameras is advantageous in terms of mitigating the complexity of the camera coordination for tracking handover and video analysis.

The fixed camera however, having continual monitoring over its FoV, is suitable for monitoring the confined but critical regions of area, e.g. a gate or entrance, to not lose any incidences. The video captured from the entrance is useful for detecting the intrusions, step 1 of the proposed AIITS approach, through the analysis for counting the number of detected human bodies entering and exiting the restricted area. Lossless video capturing and analysis are also required for the

AIITS to identify and locate the intruder among the tagged persons identified by the RTLS, step 2 of the proposed AIITS approach.

The main advantage of adopting PTZ cameras, considering their higher cost, for the AIITS is their excellent wide area tracking ability which matches the requirement of step 3. Fixed cameras as discussed cannot efficiently handle intruder tracking and impose high cost and complexity for a large number of cameras or they quickly loose the target for a smaller number of cameras. On the other hand, although the dynamic FoV may result in video loss, a high (ceiling) installation of a single PTZ camera can mitigate this problem. This installation provides a considerably large FoV. On the other side, the rotation of the camera to capture the intruder along his trajectory is assumed to be less significant in this case and not cause loosing of a large part of the scene, if the area is not very large. However, the height increase results in the resolution decrease and zooming into the details will result in losing the overall view of the area.

With respect to the above-mentioned arguments, a PTZ camera is assumed to be capable of accomplishing all the required tasks, in steps 1, 2, and 3, in a relatively larger area compared to the fixed cameras. Therefore, this research focuses on the integration of the UWB RTLS with the PTZ camera for *Automatic Intruder Identification and Tracking* across wide indoor areas. However, the currently adopted architecture can be expanded to a network of fixed and PTZ cameras in the future to provide wider surveillance without losing resolution.

3.3 Proposed Approach

In this approach, we propose the deployment of an UWB RTLS and ceiling installation of a PTZ camera for simultaneously monitoring a wide restricted area, named space S_i. This space must be defined as private in the BIM and must be linked to an Access Control List (ACL) which contains the name and authority schedule of the persons who are authorized to access this space. The individuals registered as authorized are assigned with the UWB tags for identification and tracking purpose and their assigned tag ID is added to their ACL record. The authorized persons are asked to attach their assigned UWB tag before entering the area in order to enable the AIITS to detect, position, and count the number of the present authorized persons in the space.

Another complementary sensor technology is required for capturing the untagged persons (intruders), in order to detect the intrusions, and identify and locate the intruders. Video is a suitable source of complementary data for RTLS positioning and identification and is a conventional method for security surveillance. Advanced CV methods have enabled detecting, locating, and tracking mobile objects, e.g. human bodies, in the video frames. Therefore, this research proposes fusing the video data with the real-time location and identification data coming from the UWB tags to detect intrusions, and identify and track the intruders. The RTLS sensors and the PTZ camera are installed in the area with an entirely overlapping coverage and a simultaneous operation as illustrated in Figure 3-2, to detect intrusions through distinguishing the tagged and untagged individuals.

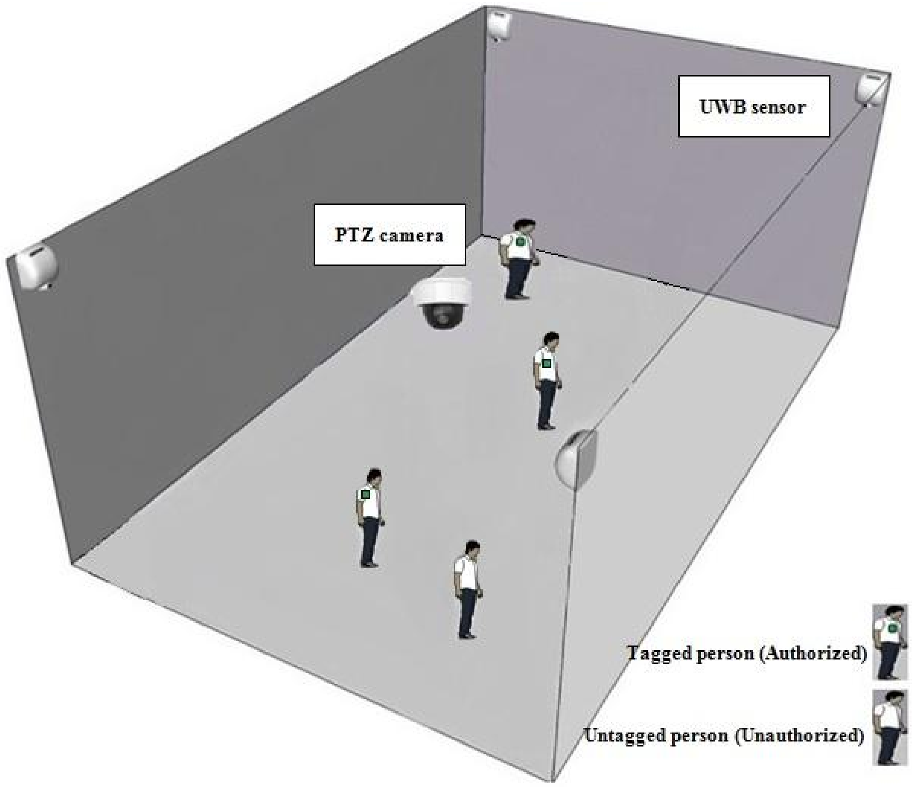


Figure 3-2 A Sample Scenario of AIITS Deployment and Operation

Environment modeling via BIM, environment perception via the sensor technologies, and data fusion are elaborated in Section 3.3.1, 3.3.2 and 3.3.3, respectively.

3.3.1 Environment Modeling via BIM

According to the requirements determined in Section 3.3.2, the BIM is used to provide the AIITS with the most up to date and enriched model of the environment which is going to be monitored. The coverage simulation is divided into two categories of: simulating the cameras' FoV and simulating the signal coverage of the RTLS sensors. The cameras' FoV simulation is possible via the BIM software through the following procedure (Asen, 2012):

(a) Adding a virtual camera into the 3D model of the target area using the location coordinates and the orientation of the real-world camera,

(b) Adjusting the FoV of the virtual camera according to the FoV and focal length of the actual camera.

However, simulating the coverage of a single UWB sensor is pointless as at least four sensors are required to build a RTLS cell for our proposed UWB positioning methodology. In other words, the field of radio coverage is considered as the area surrounded by four sensors. Nonetheless, the ultimately achieved coverage is affected by the installation point and orientation of the sensors. Therefore, best recommended practices, which are found in the RTLS set up user guide, must be taken to avoid the radio obstructions and reflections. The radio coverage field, known as RTLS cell, is created, shown, and saved within the RTLS software which can also be copied in the BIM's 3D model so that can be compared with the visual FoV of the AIITS.

Apart from the modeling advantages, the BIM can assist in storing the usage constraints of each room in the building. The object oriented nature of the BIM enables to define the following two properties for the "room" objects:

(a) If the room is "public" or "private".

(b) A list of persons who are authorized to access to each "private" room along with their authorization schedule, called ACL.

The ACLs of the private areas are then used as an authority check during tag detection and identification within that area.

3.3.2 Environment Perception via UWB RTLS and a PTZ Camera

To automate the intrusion detection, intruder identification and locating, and finally visual intruder tracking, the AIITS must collect the 2D location coordinates of all present persons as well as the number of all tagged and untagged persons in a real-time manner. In the proposed methodology, the real-time x and y coordinates of the detected persons are collected in two separate 2D vectors, defined in the Euclidean space, at each update. The vectors are named as r_t and v_t, to store respectively the RTLS positioning results (from the detected tag owners) and the video locating results (for the detected human bodies) at time t. The coordinates of the i^{th} element in the vector r_t, $i=1,\dots,n$, are stored as p_{it}^{r} and the coordinates of the j^{th} element in the vector v_t, $j=1,\dots,m$, are stored as p_{jt}^{v}.

3.3.3 Data Fusion for Intrusion Detection and Intruder Identification/Locating

The sensor modules actively percept events within their overlapping coverage and provide the fusion module with the number of the detected persons and their coordinates. As Figure 3-3 illustrates, the fusion module determines whether an intrusion has been occurred or not by comparing the number of elements in r_t and v_t and generates an alert if an intrusion occurred.

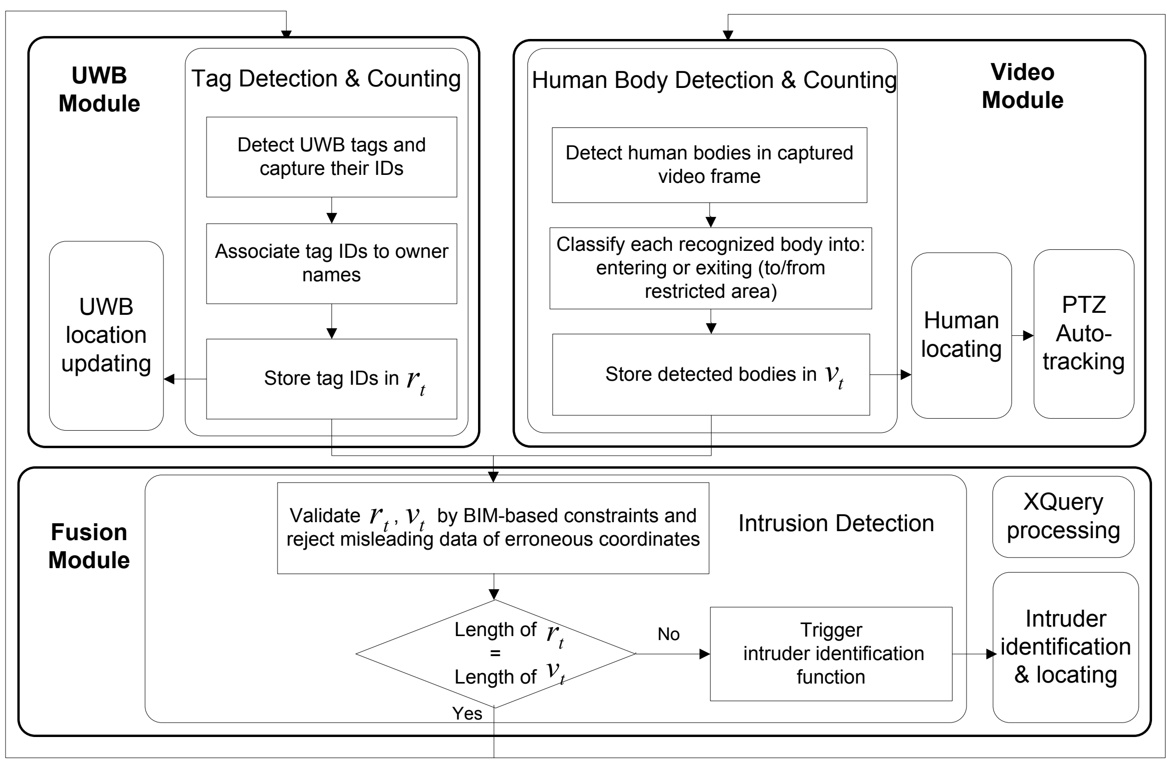


Figure 3-3 Intrusion Detection via Data Fusion Module

Figure 3-4 illustrates the flowchart of intruder identification and locating function which is triggered in case of intrusions to accomplish the intruder tracking goal. In this function, the

Euclidean distances ($\delta_{i,j}$) between p_{it}^{r} and p_{jt}^{v} are calculated (Equation 3-1) and saved in a metric space *Eud*.

$$\delta_{i,j} = \sqrt{\left(p_{it}^{r}(x) - p_{jt}^{v}(x)\right)^{2} + \left(p_{it}^{r}(y) - p_{jt}^{v}(y)\right)^{2}} \tag{3-1}$$

Then, the nearest elements (neighbors) from r_t and v_t are searched using *k*-NN (Li & Cheng, 2009; Chávez et al., 2001), to find the corresponding p_{jt}^{v} for each p_{it}^{r}. In this case, the *k* value is set to be 1. At the end, the unmatched v_t element, e.g. p_{jt}^{v}, refer to the position of the intruder.

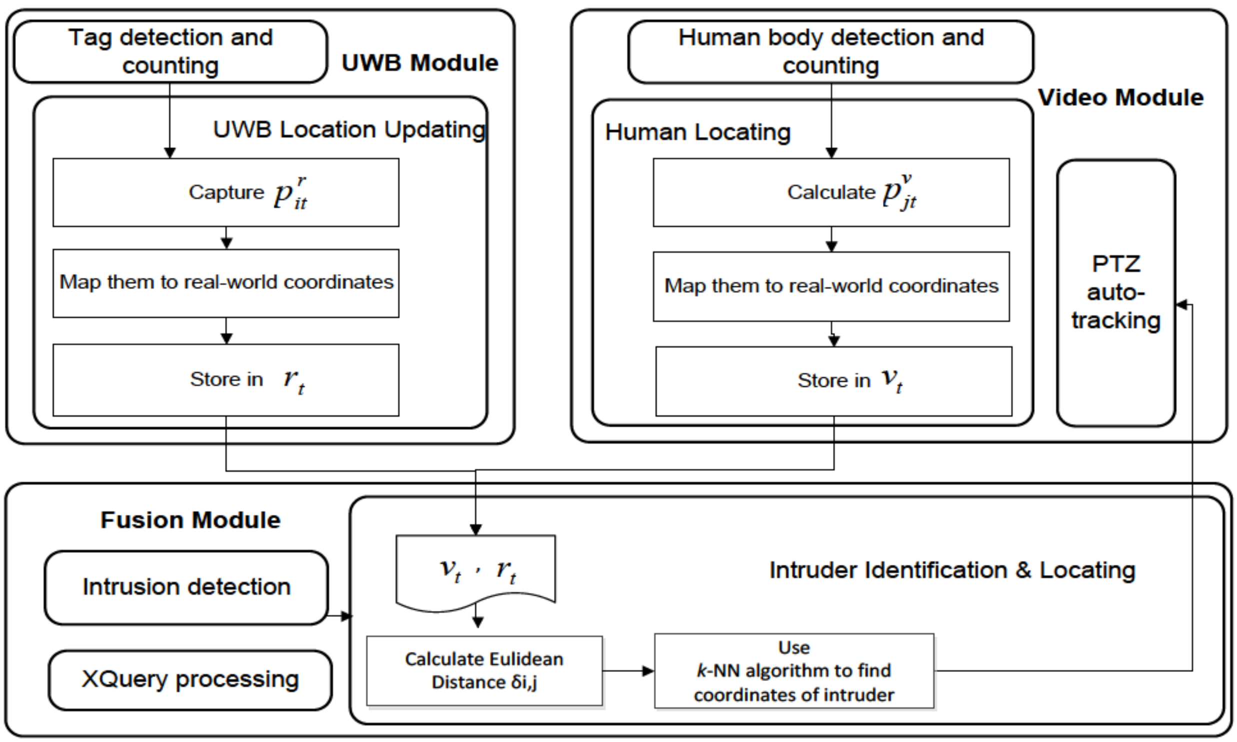


Figure 3-4 Intruder Identification and Locating via Data Fusion Module

The unmatched v_t element, as shown in Figure 3-5, is passed to the PTZ auto-tracking application in the video module for tracking purpose.

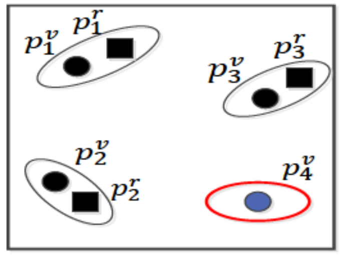


Figure 3-5 Intruder Identification via *k*-NN Algorithm

As Figure 3-6 illustrates, the location coordinates of the identified intruder is received by the auto-tracking application from the fusion module to be converted to the rotation angles for the camera; i.e. pan and tilt parameters. Then, the calculated angles are sent to the IP camera to capture the position of the intruder, i.e. the point p_{jt}^v. One possibility of expanding the proposed system is that the location changes of the identified intruder can be continuously extracted from the video and sent to the auto-tracking application for capturing high-resolution video of the intruder along his trajectory. This video is archived as an authentic forensics evidence which can be indexed by intrusion event IDs for easier search and retrieval.

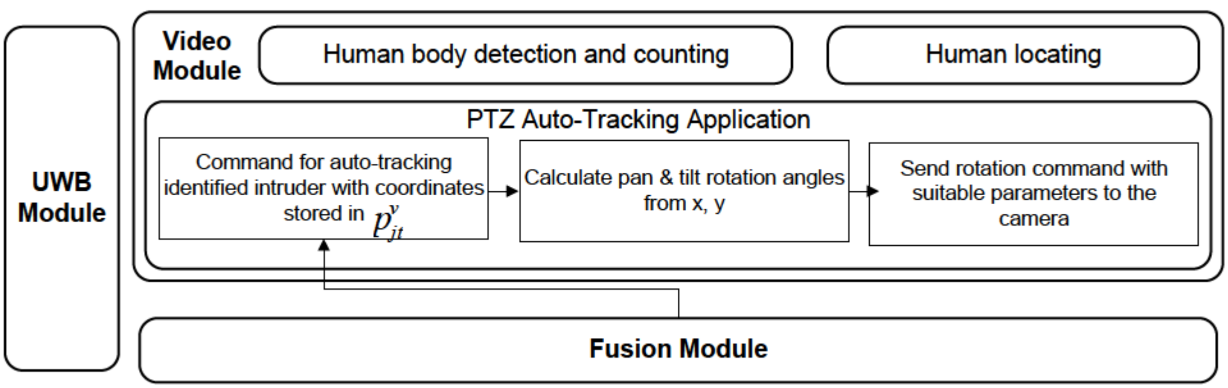


Figure 3-6 Intruder Auto-Tracking via PTZ Camera

3.4 Details of Proposed AIITS Methodology

The methodology introduced in the previous section involves the following tasks:

3.4.1 Defining Coordinate Systems for Target Locating and Tracking

In our proposed approach, the room r_i has been modeled in the BIM and a Cartesian coordinate system is defined for this model, named *BCS* short for *BIM Coordinate System*. The UWB RTLS also needs a Cartesian coordinate system for its positioning which in deployment may or not may be aligned to *BCS*. We name the *UWB Coordinate System* as *UCS* with the origin O_U which is the reference point of the RTLS measurements; i.e. the locations of the UWB tags are found with respect to this origin. The third Cartesian coordinate system is defined for the video image, named as *Image Coordinate System* or *ICS* with the origin O_I which is used for calculating the locations of the detected bodies in the video. In the proposed approach, the UWB and video location data are projected on the *UCS* for correlation and data fusion purpose. Nonetheless, the locations need to be mapped from the *UCS* into the *BCS* for location validation. For example, by using the *BCS,* the RTLS location data which are within the walls, columns, or any other unaccepted areas of the building can be removed. Figure 3-7 illustrates the relationship between the three Cartesian coordinate systems defined for the leveraged data sources (BIM, RTLS, and video). Moreover, a spherical coordinate system is required for the rotations of the PTZ camera using its internal parameters, the pan and tilt angles.

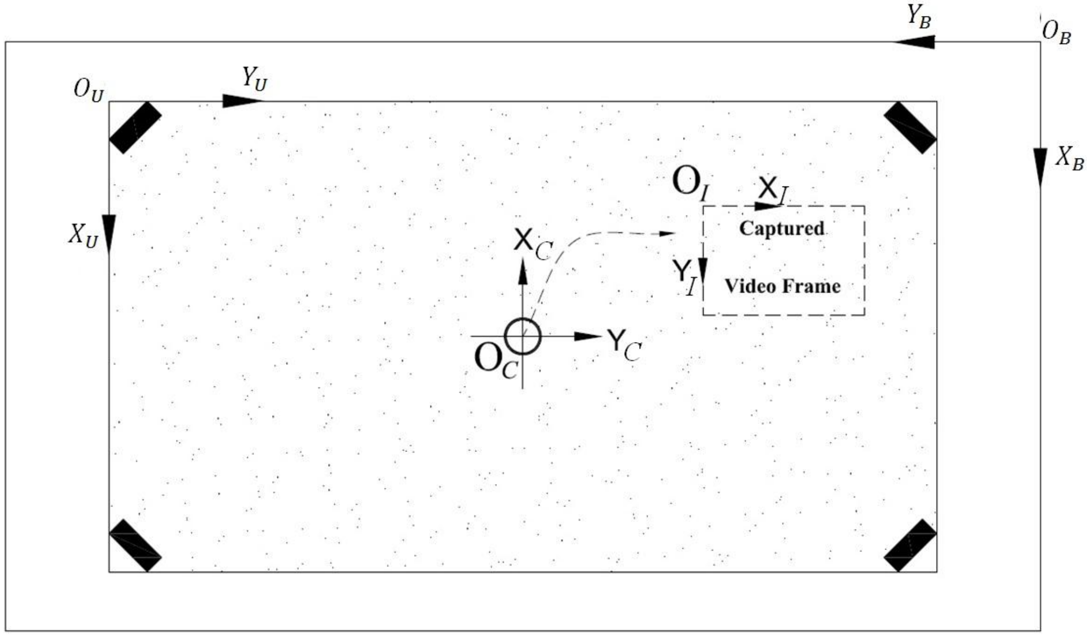


Figure 3-7 Relationship of the Three Cartesian Coordinates System in the Proposed AIITS

A spherical coordinate system is defined for dealing with the rotation angles of the camera and is named as *Camera Coordinate System* or shortly *CCS*. As illustrated in Figure 3-8, the position of a point in a spherical coordinate system for 3D space is specified by three numbers: (1) the "radial distance" of that point from a fixed origin, called *r*; (2) its "polar angle" measured from a fixed *zenith* direction, called θ; and (3) the "azimuth angle" of its orthogonal projection on a reference plane that passes through the origin and is orthogonal to the zenith, φ, measured from a fixed reference direction on that plane (Wikipedia, 2013a).

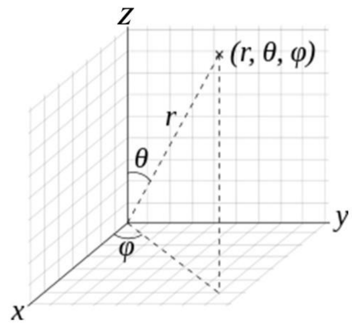


Figure 3-8 Spherical Coordinates of a Point in 3D Space (Wikipedia, 2013a)

3.4.2 Registering Authorized Persons

The proposed approach for operation of the AIITS includes the registration of the authorized people in the ACLs which in this context refers to the lists of permissions for different restricted spaces in the building. The ACLs are linked to the spaces in the BIM (ACL_i for space S_i) for authorization control. One RTLS tag is assigned to each authorized person and the ID of this tag is bound to the space IDs the owner is authorized to access. When an authorized person enters the space S_i, his tag is detected and identified via the RTLS and the tag ID is searched within the ACLs' records to find its owner ID, the spaces and the schedules that he is authorized to access. If the space ID is not found in his records or his access schedule for this space does not include current date and time, an intrusion alert can be issued. In short, linking the ACLs to the BIM spaces assist in detecting the tagged but unauthorized persons, since a tagged person may not be necessarily authorized to access each and every restricted space or at any date and time but this cannot be controlled by the RTLS, because neither the RTLS nor the ACL mechanism can restrict the access as can be done using the biometric or access card mechanisms.

3.4.3 Identification and Tracking of Authorized Persons

The RTLS detects the presence of the tags' owners in the restricted area, which enables identifying and locating the persons who are considered authorized. Location changes of each tag are calculated in near real-time which updates the position of the owner and allows for logging and further analysis such as calculating the trajectories.

3.4.4 CV-based Person Detection and Locating

Independent from the wireless sender-receiver mechanism of detection, which is limited to the tagged persons, a surveillance camera allows capturing live video from the area and ongoing events, which enables detecting everybody whether tagged or untagged. Analyzing the high-resolution video for detection, positioning, and counting the human bodies provides another source of data which can be compared against the RTLS data for accomplishing intruder identification and positioning. Mostly, CV algorithms split the video stream to its frames and analyze the frames for object detection and positioning, using the pixels coordinates (X_I, Y_I). The first coordinate (X_I) is the horizontal address of any pixel in a raster image and the second coordinate (Y_I) is its vertical address. The calculated location therefore must be translated from pixel dimensions to the real-world dimensions. As the detected bodies are usually contoured with rectangles surrounding the bodies, the center points of these rectangles can be computed and used as the location information. Equations 3-2 and 3-3 are used for translating the pixel coordinates to the real-world FoV location coordinates.

$$x = \left(\frac{X_I}{NumberOfPixelColumns}\right) \times WidthOfFoV \tag{3-2}$$

$$y = \left(\frac{Y_I}{NumberOfPixelRows}\right) \times HeightOfFoV \tag{3-3}$$

Number of pixel columns in a digital image is its first resolution number and the number of pixel rows is its second resolution number when the resolution is specified with a set of two positive integer numbers, where the first number relates to the image width and the second number relates to its height.

3.4.5 Identification of the Untagged Intruder amongst the Located Persons

Finally, the video location data are compared against the RTLS location data of the detected persons to identify the (untagged) intruders detected in the video but not the RTLS. A simple implementation of the k-NN algorithm, k = 1, can be used for associating the RTLS and CV location data by searching for the nearest neighbor of any RTLS coordinates pair in the CV location dataset. The unassociated CV location data is considered as the intruder's location and is used for intruder auto-tracking by the video camera.

3.4.6 Visual Auto-Tracking of the Identified Intruder

This research proposes using a PTZ video camera which enables tracking a mobile target (in this context, the intruder) along his trajectory in a significantly wider FoV compared to the fixed cameras. For this purpose, the location coordinates of the target must be given to the camera in terms of its pan and tilt rotation angles.

3.4.6.1 Calculation of Camera Rotation Angles from the Cartesian Target Coordinates

In order to track a target at point P using a PTZ camera, 3D Cartesian coordinates of two points are required; the center of the camera sphere and the P. The center of the camera sphere is the origin of the CCS (O_C) and the P in intruder tracking scenario is extracted from the fusion results for intruder identification and locating. The O_C and P are positioned in the UCS as: $O_C(x_{O_C}$, $y_{O_C}, z_{O_C})$ and $P(x_P$, y_P, $z_P)$. However, P must be first represented in the CCS as illustrated in Figure 3-9.

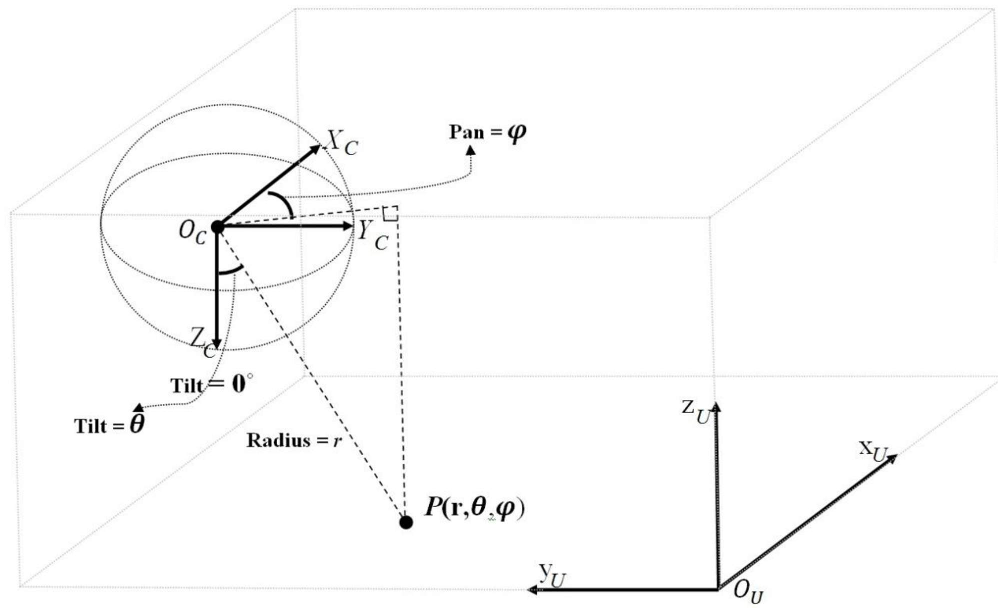


Figure 3-9 Translating Coordinates from (*UCS*) Cartesian to (*CCS*) Spherical

Equations 3-4 to 3-6 are used for calculating the coordinates of the vector $\overrightarrow{O_C P}$ in the *UCS*.

$$\Delta x = (x_P - x_{O_C}) \tag{3-4}$$

$$\Delta y = (y_P - y_{O_C}) \tag{3-5}$$

$$\Delta z = (z_P - z_{O_C}) \tag{3-6}$$

Then, the spherical coordinates (i.e. (*radius r*, *inclination θ*, *azimuth φ*)) of *P* are calculated from the origin O_C by using the Equations 3-7, 3-8, and 3-9.

$$r = \sqrt[2]{\Delta x^2 + \Delta y^2 + \Delta z^2} \tag{3-7}$$

$$\text{Pan} = \varphi = \arctan\left(\frac{\Delta y}{\Delta x}\right) \tag{3-8}$$

$$Tilt = \theta = \arccos\left(\frac{\Delta z}{r}\right) \tag{3-9}$$

In addition to the above-mentioned translation, other angular shifts may be required for the PTZ cameras having an oriented internal spherical coordinate system, as it will be described in our case study, Section 4.7.

3.4.7 XQuery Processing via Binding the Detected Events to the Cameras

For a surveillance network deployed throughout the building, we propose registering the cameras in the BIM's spaces. By registering the cameras to their covering spaces, the search for any event of interest can be limited to a drastically narrower scope in the video footage related to the space or the camera ID selected in the query parameters. By taking a similar registration approach for the RTLS cells, the cameras can be also bound to the overlapping UWB cells. Therefore, an RTLS cell which is covered by more than one surveillance camera will be registered with multiple camera objects in the BIM. This can be used for binding the UWB detected events to the cameras that captured those events. The UWB events include the ID and location updates of the detected tags. Binding the events to the cameras simplifies the practice of finding the exact video frames recorded for the detected events by narrowing down the video search scope. The recorded timestamps for the detected events helps to slide in the corresponding video frame, assuming the camera and RTLS are well-synchronized. Suppose the camera C_i is installed in the space S_i and captures the tag T_j during the time t_1 to t_2. The user can easily retrieve the video of interest by selecting different search parameters including: tag ID, camera ID, building space, and time interval.

Further analysis or off-line monitoring is also possible through linking the detected UWB tags to the video frames captured during their detection interval. For this purpose, first the update rate of the RTLS must be unified with the capture rate of the camera then the UWB detections must be synchronized with the video frames to be accurately linked to each other. The proposed approach enables the AIITS to process the following XQueries:

- Which authorized persons were present while the event "x" was occurring?

- When exactly the event “x” has happened?
- Retrieve the video frames among time t_1 to t_2 in which the tagID = “e” has been detected.

3.5 System Design

The present research aims at improving indoor security through the automation of intruder identification and tracking via the proposed AIITS. Fusing data from BIM, UWB RTLS, and PTZ camera, the proposed AIITS is superior to the similar security solutions by: (1) the decreased number of the cameras for surveillance, thanks to the PTZ capabilities of the used camera technology and the top-view which avoids video occlusion; (2) the decreased RTLS errors through the BIM validation and fusion with the CV-based location data; and (3) effective event-based video search and retrieval.

Figure 3-10 illustrates the proposed AIITS architecture. In this architecture, the UWB RTLS collects the real-time position and identification data from the tagged (authorized) persons. An IP PTZ camera is used for capturing the video from the scene to analyze for human body detection, locating and counting. The PTZ camera is essentially useful for tracking the intruder which is identified and located via the fusion module, as a part of the core software, through radio and video data association. The server machine runs the server applications of the RTLS and the camera, as well as the developed AIITS core software. The BIM software is also installed on the server machine to support interoperations with the BIM.

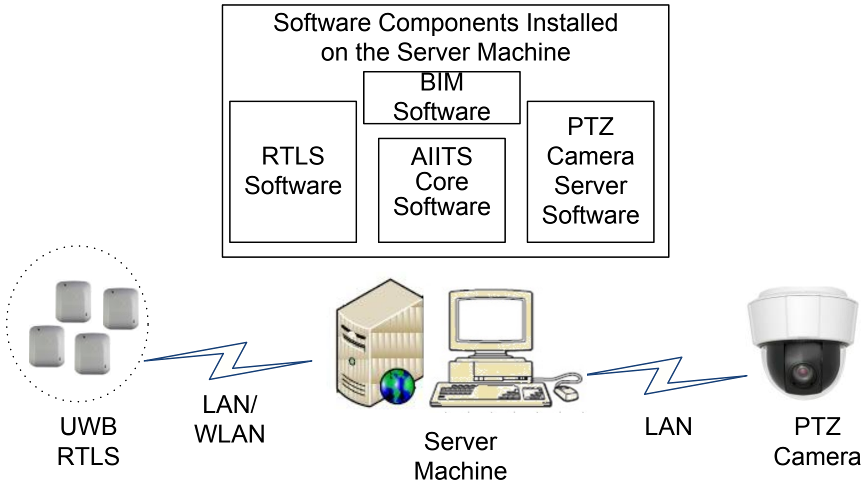


Figure 3-10 AIITS Design

In addition to automatic intruder identification and locating, the proposed MSDF-based AIITS delivers some added values to the state-of-the-practice security solutions, such as XQuery processing and event-based video retrieval. For example, the sensor data fused in the AIITS, containing the correlated and meaningful information about the security events, can be stored as forensics evidence or for further analysis (Kolias et al., 2010). Although the present research is carried out with a simple AIITS architecture having a single PTZ camera, the AIITS architecture is easily scalable to leverage more PTZ cameras. Also, a combination of fixed and PTZ cameras can be adopted as discussed in Table 3-2.

3.6 Summary and Conclusions

This chapter presented our proposed approach for the development of the AIITS covering different topics including: analysis of the AIITS requirements, technology adoption and integration criteria to meet the discussed requirements, proposed methodology, and finally the suitable system design to implement the methodology and meet the requirements of AIITS.

Our proposed *Automatic Intruder Identification and Tracking* approach takes advantage of the MSDF concept to provide required supplementary data for performing intrusion detection, intruder identification and locating, intruder auto-tracking, and XQuery processing. We proposed applying data fusion on three complementary data sources namely: BIM, UWB RTLS, and PTZ camera. The superiority of the afore-mentioned technologies has been proved in the literature (as reviewed in Subsections 2.2.1, 2.3.1, and 2.3.2.3) and has been investigated in Section 3.2 for AIITS requirements.

The proposed AIITS delivers some added values to the state-of-the-practice security solutions, such as XQuery processing for event-based video search and retrieval. Nonetheless, the simple architecture proposed in the present research which leverages a single camera cannot track more than one intruder at a time but it can be easily expanded to involve multiple cameras to accomplish multi-target tracking.

CHAPTER 4 IMPLEMENTATION AND CASE STUDY

4.1 Introduction

Based on the system design elaborated in Chapter 3, this chapter presents the technology integration and the core AIITS software development approach. The hardware and software components of the AIITS are introduced in Sections 4.2 and 4.3, respectively. The development of the core AIITS software is explained in the rest of this chapter as follows: the programming platform and language are introduced in Section 4.4; the third-party APIs are introduced in Section 4.5; and the features and implementation limitations of the currently implemented modules are described in Section 4.6. Our case study for validation of the AIITS functionalities and evaluation of its performance is presented in Section 4.7. The summary and conclusion are provided in Section 4.8.

4.2 Hardware Components of AIITS

4.2.1 UWB RTLS

The Ubisense 2.1 platform is adopted as the UWB RTLS to satisfy real-time location awareness and visualization requirement. Objective of the Ubisense system, was delivering a system that could provide accurate location data (1 m or better accuracy) in very difficult environments (such as manufacturing facilities) with high reliability. To achieve this purpose, they chose UWB which is the only wireless solution capable of meeting all of these requirements simultaneously. The Ubisense RTLS employing both time-of-arrival and angle-of-arrival measurements delivers the best accuracy and reliability.

Figure 4-1 illustrates the block diagram of the Ubisense Location System linking four sensors spaced in the area to be covered with a switch or hub to send real-time measurements to a platform server which performs further processing and visualization. Ubisense tag locations are sent via standard Ethernet cable or wireless LAN to the Location Engine which processes the data and passes the information via an industry-standard API to applications.

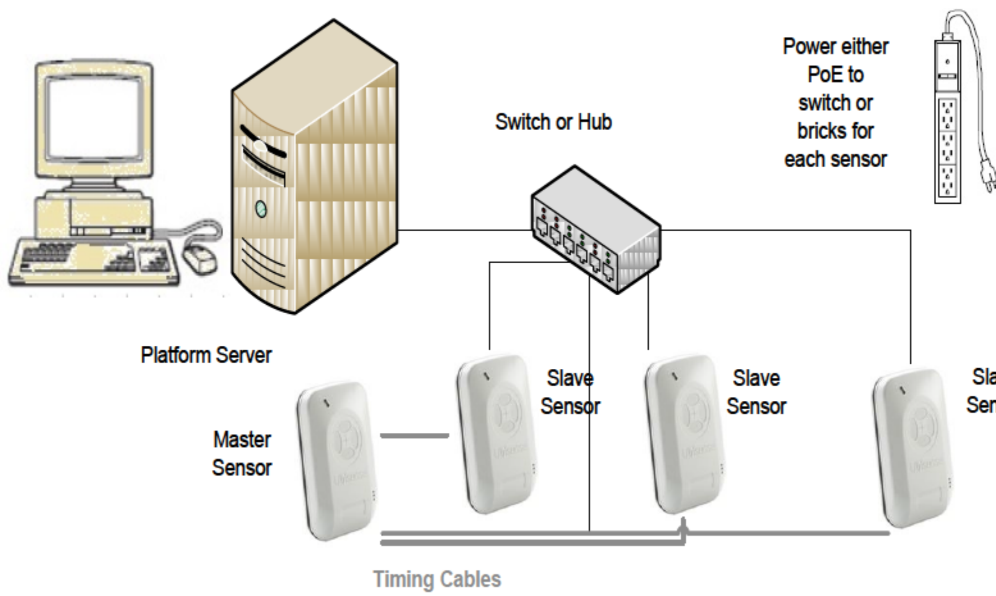


Figure 4-1 Ubisense Location System Block Diagram (Ubisense, 2008)

The key physical components of the location system are: Tags, Sensors, Network components, and Platform Server which are briefly described as follows.

(a) Ubisense Tags

Ubisense tags can be found in a number of types such as *Slim* and *Compact*, illustrated in Figure 4-2. The basic functionality of all the types are the same and only the features vary. *Slim* tags are more comfortable to be worn by people whereas the *Compact* tags are especially designed for harsh industrial environments (Ubisense, 2012). In Ubisense RTLS, the tags are assigned to arbitrary owner objects in the Ubisense Site Manager application for reporting their real-time position. The locating can be performed in 3D and up to the accuracy of 15 cm (dependent on system configuration and environment). Besides the tracking capabilities, they have features like LEDs for easy identification, a buzzer to provide basic messaging abilities, a motion detector to instantly activate a stationary tag and two push buttons to trigger events. The size and other specifications of the *Slim* and *Compact* tags can be found in Appendix A.


a. Slim Tag


b. Compact Tag

Figure 4-2 Series 7000 Ubisense Tags (Ubisense, 2012)

Each tag is registered with its containing location engine cell and is inserted into the schedule for that cell. The schedule determines when the tag should emit pulse. Each tag is allocated an appropriate schedule of time slots, in which it is active, and the schedule can be changed on the fly in response to the application's requirements (Steggles & Gschwind, 2005). When a tag emits UWB pulse, the signal is picked up by one or more sensors in the cell. The slave sensors decode the signal and send its angle of arrival and timing information to the master sensor by Ethernet. The master sensor accumulates all sensed data and computes the position of the tag. The position data is delivered to the configured sink address. In a full Ubisense platform, the sink address is automatically set for each cell.

The tags use two separate radio channels to operate: a bidirectional conventional telemetry channel and a transmit-only UWB channel. Since there is only one UWB channel in each cell, only one tag can be located at a time. The location engine cell divides time into time slots and allocates appropriate time slots to the tags according to their requested update rate. The conventional radio is used to manage this scheduling and the UWB channel is only used when a location data is generated.

(b) Ubisense Sensors

Sensor units are placed above the area the tags are to be tracked. They must be provided with power, networking and timing cable connections. Sensor are arranged to cooperate in location engine cells, each cell having a single master sensor and a number of slave sensors. The master and slave sensors are physically the same. The sensors are measurement devices containing an array of antennas and UWB radio receivers to detect UWB pulses from Ubisense tags, allowing the Location System to position tags in 3D by up to 15 cm precision. The series 7000 Ubisense sensors are illustrated in Figure 4-3 and their specifications can be found in Appendix A.

Figure 4-3 Ubisense Series 7000 UWB Sensors (Ubisense, 2012)

A simple sensor configuration to create a cell is a square having sides in the range of 10-30 m. the sensors are mounted at the corners of the square, near the ceiling to provide a good line of sight. The sensors should be pointed towards the floor in the middle of the square and having no roll. The installed sensors are connected to the Ethernet switch with a network cable. One sensor is configured to be master and the other three will be slaves. A timing cable should connect each slave sensor to one port on the master sensor. Sensors can be powered from a power brick or using Power over Ethernet (PoE).

(c) Ubisense Network Components

Sensors are connected by standard 100BASE-TX Ethernet to the master and to the platform server. A switch or a hub connects the sensors and the platform server together and if sensors are to be supplied via PoE, the switch must be a PoE capable switch.

(d) Ubisense Location Platform

The Ubisense location platform is meant to bridge location sensors, business systems, and activities and is a suite of software components that enable setup, calibration and configuration of location sensors and tags. Figure 4-4 illustrates the architecture of the Ubisense Location platform. The Location Platform has a service-oriented architecture and can scale seamlessly from a laptop to a cluster of multiple CPUs.

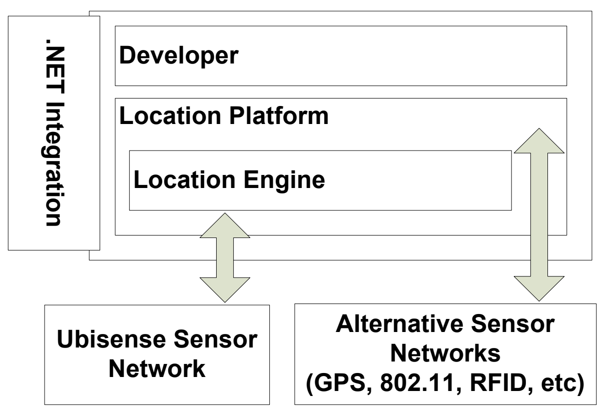


Figure 4-4 Ubisense Location Platform Architecture (Ubisense, 2013)

The Ubisense employs a combination of TDoA and AoA localization algorithms and requires at least two readings in order to generate a 3D position for the detected tag. An individual time slot is over 26 ms duration that leads to an update rate of 39Hz per cell (the maximum update rate of 160 Hz is also available for upgraded systems); each individual tag has a maximum update rate of 10Hz, though. In a typical open space, a location accuracy of 15 cm can be achieved across 95% of readings (Steggles & Gschwind, 2005).

4.2.2 PTZ Camera

The Sony SNC-ER 580 camera is adopted as the PTZ video camera to satisfy reliable target tracking requirements including high resolution video capture and wide dynamic FoV with PTZ capabilities. The name SNC stands for Secured Network Communication which is an application level solution for security, used to facilitate connection encryption for Session Announcement Protocols (SAPs) like DIAG and RFC. The SNC can provide security in terms of client-server authentication and provides the client with data confidentiality (Sony, 2011).

The highlight features of the adopted camera can be listed as follows:

- High quality live images can be monitored at a maximum frame rate of 30 fps,
- It supports three video compression modes (video codecs): JPEG is the best choice for high-quality still images; MPEG-4 provides clear video images over limited-bandwidth networks; and H.264 is the alternative for severely limited-bandwidth networks, with twice the efficiency of MPEG-4,
- It is capable of high-speed (maximum 300° rotation/second) panning (up to 360°) and tilting (maximum 210°),
- Up to 5 users can view images from one camera at the same time, and
- Date/time can be superimposed on the image.

Optical zoom of 20x, digital zoom of 12x, intelligent motion detection, camera tampering detection and alarm functions are further interesting features that can be mentioned (Sony, 2011). The complete camera specifications can be found in Appendix B. Figure 4-5 illustrates the simple configuration that enables connecting to the PTZ IP camera from the server machine which runs the camera server software, called Realshot Manager, for the purpose of setting and monitoring. The camera can be powered either via power brick or through HPoE (High-Power over Ethernet), i.e. it can be powered through the same cable that it uses for data transfer, resulting in simplified physical infrastructure. The camera is connected to the switch via standard 100BASE-TX Ethernet cable. The server machine must be connected to the same network segment via the switch to detect, register and access to the IP camera.

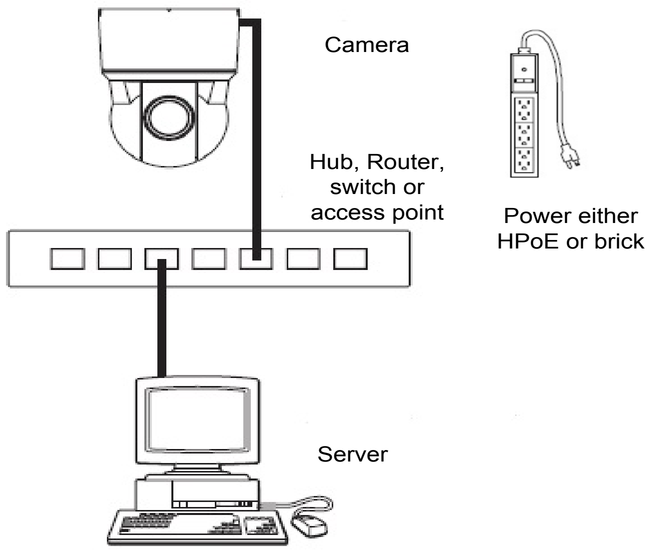


Figure 4-5 Sony IP Camera Network Configuration (adapted from Sony, 2008)

To connect the camera to a network after installation, first a new IP address must be assigned to the camera. For this purpose, the SNC toolbox application must be installed on the server machine via its installation wizard. The system requirements for server machine of the camera are provided in Table B-1 and the initial camera setup using SNC Toolbox application is explained in Appendix C.

4.2.3 Server Machine

The server is a desktop computer with Intel(R) Xeon(R) CPU X5550 @ 2.67 GHz and 6.00 GB RAM which is operating Windows 7 Enterprise 64-bit.

4.3 Software Components of AIITS (Installed on the Server Machine)

Server machine runs the server software for the UWB RTLS and the IP PTZ camera, as well as the BIM software. The Ubisense .Net API, Sony Network Camera SDK (x86), and Autodesk Revit Architecture 2012 API are installed on the same machine for the purpose of integrated interface development. The integrated application is meant to implement interfaces to the core functionalities of the third-party sensor platforms. Data fusion and linked data storage are also implemented in the application.

4.3.1 BIM Software

The Autodesk Revit Architecture 2012 is installed as the BIM software. Autodesk Revit software is specifically built for BIM, offering a coordinated and consistent model-based approach for design and construction. Autodesk Revit is a single application that provides features for architectural design, MEP and structural engineering, and construction such as: using intelligent parametric building components to improve the accuracy of design; bidirectional associativity which automatically reflects any design change throughout the whole model; work sharing among multiple users on the same intelligent model; and providing better insight into constructability of the building elements (Autodesk, 2013).

4.3.2 UWB RTLS Software

The Ubisense 2.1 platform is installed on the computer as explained in Appendix E. Ubisense Location Platform supports: (1) spatial event detection that is detecting relationships between objects and turning location events into data to be used by application programmers; (2) visualization including 2D and 3D views in smart client programs, web browsers and hand-held devices; and (3) open integration with a suite of applications for vertical markets.

4.3.3 PTZ Camera Software

Sony Realshot Manager is the software application used to setup and monitor Sony IP cameras in a multipoint monitoring system. The installation procedure of the Sony Realshot Manager Software could be found on Appendix F. By installing the Sony Realshot Manager on the server machine, initial setups including video codec, frame rate, resolution, etc., are applied on the SNCER580 PTZ camera. The Sony Realshot Manager provides three different codec types for monitoring and recording the video which includes: JPEG, MPEG4, and H.264 which were introduced in subsection 2.3.2.5. Among those, the JPEG is the easiest and least CPU-intensive method for compression and decompression of video which requires higher bandwidth and storage compared to the other codecs. The MPEG4 option enables for more compressed video transmission and storage resulting in less bandwidth and memory requirement but higher computation. The third option, H.264, is the ideal codec in the surveillance industry offering the most appropriate tradeoff among CPU requirement and bandwidth/storage reduction (Sony, 2008).

In Realshot Manager, the user can monitor up to 16 IP cameras at a time by selecting the appropriate layout. Video search and recorder setting options enable the user to search in the recorded video files and to setup the recording. The user can switch among LIVE and PLAYBACK modes, adjust the zoom or control the rotation of camera, record the captured video from each camera, playback the recorded files, and export them to be converted from .cam to .avi, which is more compatible with different players and analytic software.

4.3.4 AIITS Core Software

This section explains the five programming modules that are developed by using the third-party APIs and the data fusion algorithm, as the core software for automatic Intruder Identification and

Tracking. Figure 4-6 illustrates the architecture of the software comprising: (1) BIM module which is developed via Autodesk Revit Architecture 2012 .Net API; (2) UWB Module which is developed via Ubisense 2.1 .Net API; (3) Video Module which is developed via Sony Network Camera SDK (for x86) in order to connect to Sony IP PTZ camera; and (4) Data Fusion Module which is simple implementation of *k*-NN algorithm; and (5) GUI which is a Windows form designed and developed in C# as the source of events and is handled through different event-handler methods scattered over the aforementioned modules.

Simulation and sensor deployment design are performed via the third-party BIM software. A BIM plug-in is developed to import the UWB location data to the BIM software and visualize them in the area model. In order to infer intrusions, both of the Video and UWB modules have an equal role in locating persons; however, video data are more comprehensive as the camera captures both tagged and untagged persons. Visual identification of persons is not needed in this application and the intruder identification is sufficient through correlating detected tags and bodies via UWB and video to find the intruder. Location of the (untagged) intruder person is extracted from video to be used for tracking by the PTZ camera. The data fusion module exchanges data with all sensor modules to correlate data in order to infer intrusions, identify and locate the intruder, and process the XQueries.

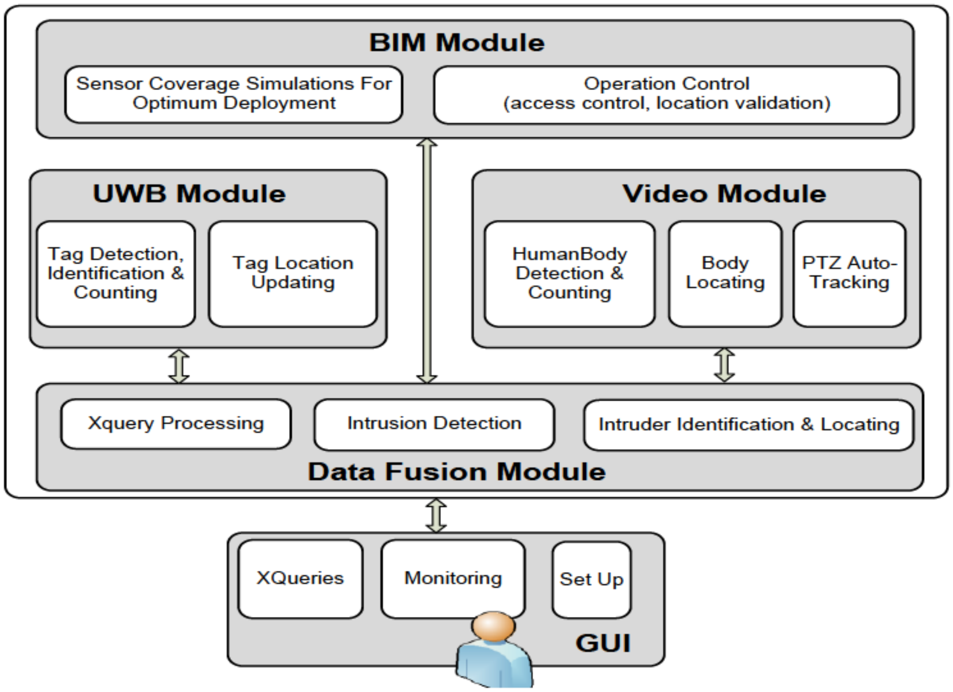


Figure 4-6 Modules of the AIITS Core Software

4.4 Programming Platform and Language

In this research, .Net framework 3.5 is used to develop the GUIs of the three components of the proposed system. The .NET Framework is a popular platform which provides a comprehensive and consistent programming model for building applications that have visually stunning user experiences and seamless and secure communication (Microsoft, 2013).

Microsoft Visual Studio 2010 is used as the Integrated Development Environment (IDE) to develop the GUIs. Visual Studio supports many different programming languages for code editing and debugging. The Windows Form Designer is used to build GUI applications and the GUI is linked with code using an event-driven programming model. In this research, C# is used as the programming language supported by the three third-party software modules.

4.5 Third-party APIs

The third-party platforms that are adopted for BIM, UWB RTLS, and PTZ camera provide their own APIs in C# to allow the users to extend the core functionalities of the system by their own business need and integrate desired functionalities into their application.

4.5.1 Revit Architecture .Net API

Autodesk Revit provides a rich .Net API that can be used to extend the core functionality of Revit in simulation, design and other construction phases. For the purpose of this project, the API is mainly used to develop Revit plug-in that visualizes UWB location data in the building model by importing them from UWB log file.

4.5.2 Ubisense .Net API

Ubisense .Net API is provided by Ubisense to allow users to extend the core functionalities of the Ubisense to fit their business requirements. For example, users can implement functions to detect presence of tags, get/set/remove objects' names, get list of object types and names, get location/spatial events, send data to tags, etc. In this research a software piece is developed by using the Ubisense API which provides programming interface to key Ubisense components (Jian, 2011). Using this API enabled us to get the list of the owner types of UWB tags, as well as the list of the names of the detected tags for each owner type. Also, the real-time location data of selected tags are logged in text files. Moreover, by using the API, a visualization map is created for this application which allows viewing tag movements in the area in 2D or 3D.

4.5.3 Sony Network Camera SDK

The Sony Network Camera SDK is composed of the following libraries, which are applicable for C++ language but have similar .Net APIs in *.Net Class Library*: (1) *SNC Core Library* which is the most significant library in the SDK as it manages the camera handle and communicates with the camera over the network. The camera handle is used to identify a camera by giving the belonging information such as the IP address, port number, proxy, and authentication info; (2) *SNC CGIWrapper Library* which operates the values of CGI-parameter in the camera. This library uses COM (standing for Component Object Model) and requires initializing COM before using it; (3*) SNC Stream Library* which provides API to play and record video and audio; (4) *SNC Joystick Library* which can be used to implement pan, tilt and zoom of the camera so that users can perform PTZ control by joystick; (5) *SNC Version Up Library* enables to integrate camera's firmware upgrade function to the application; (6) *SNC Audio Upload Library* which enables integrating audio upload function to the application; (7) *SNC Automatic Discovery Library* which enables integrating the function of automatic camera discovery on the connected network to the application. Besides the above-mentioned libraries which are used based on the required functionality to implement, there exist two more libraries in the SDK: (1) *SNC SDK Common Define Library* which includes the definition and usage of the structures and errors that the camera uses in common such as: the camera handle used in the SDK, the structure that keeps basic information to access the camera and is called *NETINFO*, the structure which keeps video and audio codec information and is called *CODECINFO*, and so forth; (2) *SNC .Net Library* which includes the functions corresponding to C++ libraries' functions.

The SDK includes some sample applications that have implemented the functionalities of provided libraries in C++ and .Net (C#) projects. The documentation of each library together with its sample applications help to understand the procedure of implementing each functionality through the required APIs.

4.6 Features and Limitations of the AIITS Software

The AIITS software consists from three main applications: (1) BIM visualization application which is a plug-in, Class Library, developed for Autodesk Revit Architecture 2012 to visualize the location data of the RTLS from the log files in the Building model; (2) RTLS interface, a Windows form application developed to log data about the selected tags, view the RTLS cell and the tags' positions and movements in a 2D or 3D map of the area, and process a sample XQuery; and (3) Camera interface, a Windows form application that allows the user to connect to the SNC-ER 580 IP cameras, view live video, perform PTZ control, record and playback the video, and perform auto-tracking for a selected UWB tag. The two Windows forms applications are implemented in a single Visual Studio project but are introduced separately after the BIM visualization plug-in application.

4.6.1 BIM Module

The BIM module used in this project is a plug-in application developed for Autodesk Revit Architecture in the master thesis of (Jian, 2011) to visualize the logged location data within BIM as shown in Figure 4-7. Further implementation details are available in Appendix D. Although Ubisense API provides visualization functionality for user applications, a Revit plug-in is preferred as it enables taking advantage of the viewing options in the Revit BIM, e.g. different colors for different tags, views from different angle and better zooming. In addition, Ubisense visualization function is applied for real-time monitoring of the tags' positions in the UWB cell which is not sufficient for constructing the trajectories of the tags. For this purpose, an additional analysis step is required to be applied on a sequence of tag's location changes in order to generate the trajectory line.

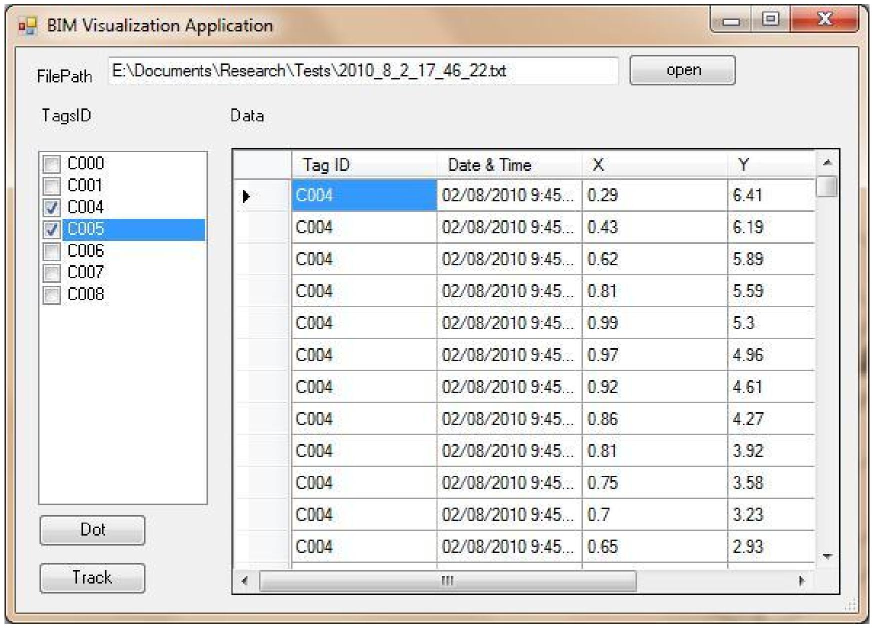


Figure 4-7 BIM Visualization Application Window (Jian, 2011)

This plug-in executes within Revit Architecture and visualizes the location data of a log file which the user imports in it. The location data in our project is captured and collected by another logger application, which is developed for Ubisense RTLS and is presented in next section, in the form of (x, y, z) coordinates, and is fed into the BIM Visualization plug-in in the format of text file. The text file is parsed and the tag IDs are listed in the left hand side column which is a checklist box for selecting the tags the user wants to visualize their trajectories. The coordinate and timestamp information of the selected tags will also be displayed in text format in the right hand side Data Grid View. The result of clicking *Track* button of the illustrated plug-in window is shown in Figure 4-8. Each green dot represents the position of the selected tag in a specific timestamp and linking the consecutive dots generates the trace of that tag during the location logging. However, the current implementation of the BIM plug-in has some limitations, e.g. it does not allow zooming or different viewing options of Revit.

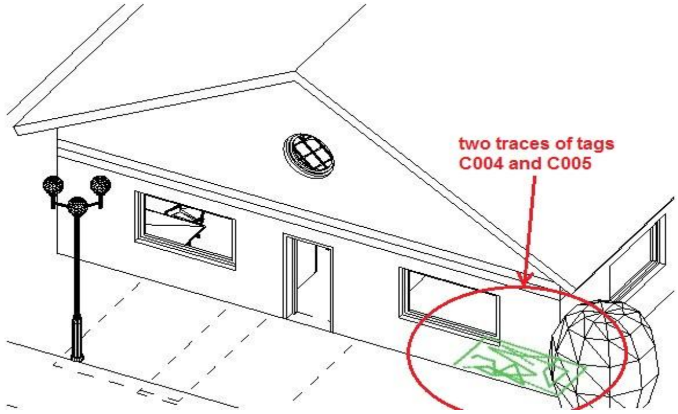


Figure 4-8 Result of Running BIM Visualization Plug-in within Revit Architecture (Jian, 2011)

4.6.2 UWB Module

The UWB interface includes two main parts: (1) An UWB (location) Logger for collecting the real-time location data of the user-selected tags for the purpose of off-line analysis; and (2) XQuery Processing of the tags of interest which searches over both the UWB log files and the video recordings. An area map is also drawn for visualizing the connected Ubisense cell and the detected tags movements, in 2D or 3D. However, this visualization does not support different viewing and zooming options or distinct coloring for different tags. In a client machine, the visualization functionality of the BIM visualization plug-in introduced in Section 4.6.1 and the visualization map in the UWB Logger application can be used complementarily to obtain both real-time vision and non real-time trajectories of the detected tags. Figure 4-9 illustrates the UWB RTLS interface which lists the detected tags for the user and he can select any of them for logging purpose. Also, the user can see tags' relative positions and their movements in the area map. A proof of concept XQuery function is also implemented which enables retrieving video frames which have been recorded during detection of a tag of interest.

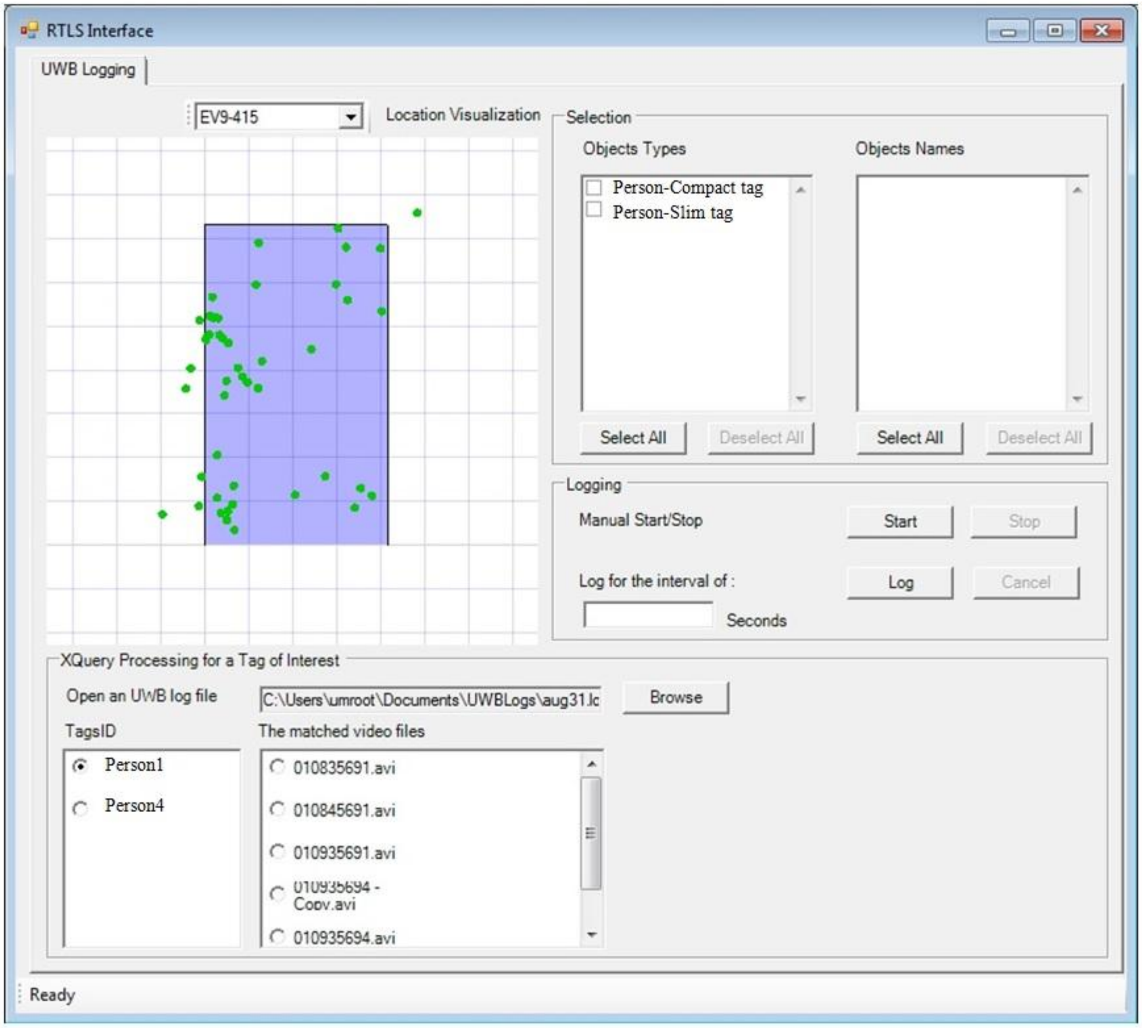


Figure 4-9 AIITS RTLS Interface

The current implementation of the UWB module has some limitations, which includes:

(1) The visualization map does not include ID and location information of the visualized tags in the cell.

(2) The XQuery processing is implemented only for a single query template. Future work must be done on implementing the other types of XQuery processing based on frame indexing with arbitrary UWB or other data.

4.6.3 Video Module

Our video module consists of two separate parts: (1) camera application, and (2) CV program. Camera application is an interface for monitoring and controlling the camera. The CV program handles all the tasks for human detection, locating, and counting in the captured videos as explained below.

4.6.3.1 Camera Application

The primary objective of implementing this module is to provide the user with a GUI that enables controlling and monitoring the SNC-ER580 camera for the purpose of intruder tracking. Although the user can find some of the features scattered through the Realshot Manager and SNC toolbox applications, this interface is specifically designed to be integrated with UWB RTLS interface, by gathering the required features of camera through its API for intruder tracking purpose. The major issue in Realshot Manager that motivated us to develop a similar application via Sony SDK is the special recording file format (.cam) which is used in the Realshot Manager and is not supported with any well-known image processing library and platform. The Realshot Manager however has the option for converting the .cam files to .avi which is more supported with video analysis libraries but it is time consuming and not user-friendly. The record function in the developed GUI is implemented so that it directly records .avi file format.

Figure 4-10 illustrates the AIITS camera control interface which consists of four different features group boxes for: (1) camera connection asking the user for camera's IP address, port number, and the account credentials. Upon successful connection, live video capture of the camera is displayed on the display part of the interface; (2) manual PTZ control including: Directional control, Absolute/Relative move, and Continuous move with adjustable rotation speed. The speed range is 0-100 for directional and absolute/relative moves but expanded towards the -100 for continuous moves which is meant for selecting the direction of movement by selection of positive or negative numbers; (3) video record and playback allow recording the live video into .rec or .avi files has for playback and off-line processing; and (4) Auto-tracking implemented for automatically following a user-defined tag owner by the camera. In this phase, auto-tracking feature is not implemented for real-time tracking, and is working only for location data extracted from UWB log files.

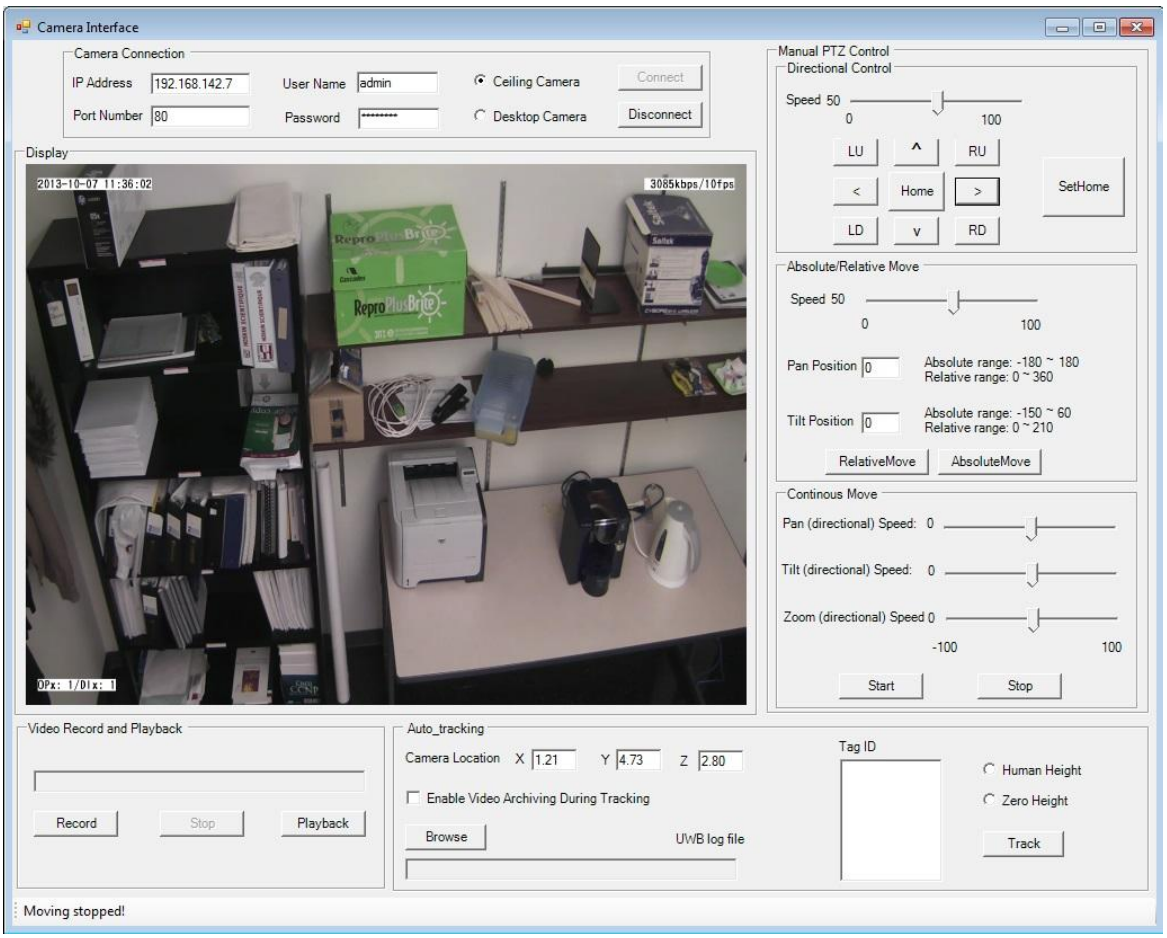


Figure 4-10 AIITS PTZ Camera Interface

4.6.3.2 CV Program

To analyze and extract the desired features from the captured video, additional functions must be implemented including: human detection, locating, and counting. We explored the possibility of implementing the required functionalities using the OpenCV library and due to the problems we faced in connecting to the SNC-ER 580 camera and also the detection weakness from top view, we used an open source Matlab program (Felzenszwalb et al., 2010) for image processing.

(a) Human Detection

For this purpose, we used a Matlab program developed based on the work of (Felzenszwalb et al., 2010) which can basically detect any object of interest after a good training procedure. Using this program demands: (1) splitting the captured video stream to images; (2) sampling the images for training and testing purpose; (3) annotating the objects of interest (human bodies) in the training images to feed into the trainer program; (4) inputting the resulting model into the detector program; (5) applying the detector on the testing images; and finally (6) reassembling the test images (superimposed with the detection rectangles) as a video file. This CV procedure is a big bottleneck for our real-time application which must be improved in future work. Figure 4-11 is the result of running the program for one frame of a video captured during one of our tests and it shows two redundant (false) detections.

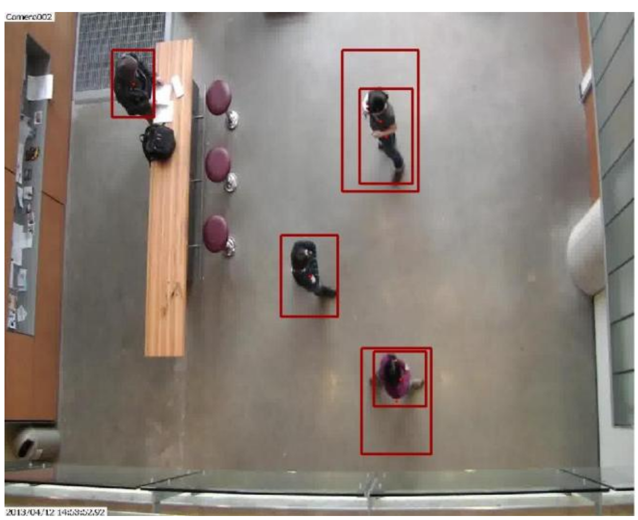


Figure 4-11 Preliminary Human Detection Results in One Single Video Frame

(b) Human Locating

We implemented human locating by calculating the center point of the detected rectangles and translating the pixel coordinates to the real-world coordinates by using Equations 3-2 and 3-3 discussed in Section 3.4.4. The corresponding codes can be found in Appendix F. We also improved the Matlab code to eliminate the false positives which result in false intrusion detections. For this purpose, we involved a threshold value to remove one of the rectangles placed in another rectangle through locating the rectangles' center points and checking their distance against the threshold value. Finally, the filtered and translated center coordinates are exported into a text file as the human locating result.

(c) Human Counting

By filtering out the redundant central points of the detected rectangles, the number of the remaining rectangles is considered as the number of the detected human bodies and is exported in the above-mentioned text file.

Implementation Limitations of the Video Module

(1) If a tag owner moves fast, UWB can better reflect his speed than the PTZ auto-tracking application. This problem is caused by the camera's limited rotation speed and not the processing. Also, for high update rates of the UWB tags, because of the large number of small-size rotations, PTZ cameras perform poorly (although, our adopted camera is claimed to have 300 degrees per second performance. In fact the camera can rotate 300 degrees per second in one shot, but if it is commanded for a large number of small rotations the tracking speed decreases).

(2) Fast UWB updates and therefore quick camera rotations cause some jitters in the video which could be smoothed by future software enhancements.

(3) The present camera application is not programmed to track a real-time target (intruder) and is only tested for replaying the trajectory of a captured tagged person. The future work must utilize this programming logic to address the real-time moving target.

(4) 2D location coordinates from the image plane cannot be translated into 3D real-world coordinates unless some extra information or assumptions exist. For example an additional view of the same scene at the same time (stereo vision) can be applied to reconstruct a 3D point from the pixels (OpenCV Answers, 2013). Powerful machine learning techniques are applied to learn the 3D structure of a scene as a function of the (single) image features in (Saxena & Y. Ng, 2013). To avoid applying additional cameras or CV techniques, we used a camera attached to the ceiling with downward orientation to neglect the Z (depth) dimension which is unnecessary for our application.

4.6.4 Data Fusion Module

As discussed in Section 3.3.3, this AIITS module is responsible for intrusion detection and intruder identification/locating through associating the CV and UWB data. In a case of intrusion, the fusion module finds the intruder's location through associating the CV and UWB location data and detecting the CV location which cannot be associated to any UWB location. The input data for the fusion module are the results of the CV and UWB modules which have been synchronized and aligned in the *UCS*, as it will be explained in our case study, for association purpose.

To perform data association, the *k*-NN algorithm has been implemented in Matlab to find the nearest (k=1) neighbor of an item in the CV dataset through comparing it against all the UWB data items. This program first calculates the Euclidean distance between each UWB-measured location point and each CV-measured location point (Equation 3-1). Then it associates the nearest neighbor (or the minimum Euclidean distance) of each CV data item from the UWB dataset. After finding all the minimum-distance UWB and CV-based location points for association, if any CV-based point remains unassociated with any UWB data, it will be considered as the location of the intruder. This location is in fact the object in the video with the maximum distance from all UWB-measured points.

4.7 Case Study

A preliminary test was performed in a laboratory of Concordia Institute for Information Systems Engineering, with dimensions of 7.32 m × 4.16 m × 3 m, to investigate the feasibility of the proposed AIITS by checking its sensor components and collecting some real data for fusion. Four UWB Sensors (Ubisense RTLS, 2012) were installed at the four corners of the lab and one PTZ Camera (Sony SNC-ER580, 2012) was mounted at the ceiling of the room which could only cover an area of 1.57 m × 2.87 m within the RTLS cell (almost laboratory wide). Some reference angles of the camera were measured as shown in Figure 4-12 which helped us to interpret camera behavior for different input angles and to implement the auto-tracking application for it. As this figure illustrates, the PTZ camera is centered at point O_C and can rotate in two perpendicular circle planes: panning plane (parallel to the floor) and tilting plane (perpendicular to the former one). The upper rotation limits of the adopted camera for this downward installation are marked as the 60˚ and -150˚ angles which define the 210˚ tilting range of this camera. Also, the tilting origin and the vertical-to-the-bottom orientation are marked respectively as the points 0˚ and 45˚. Moreover, this figure illustrates the relationship between the corresponding pan and tilt angles (in the *CCS)* of an arbitrary point P with its Cartesian coordinates.

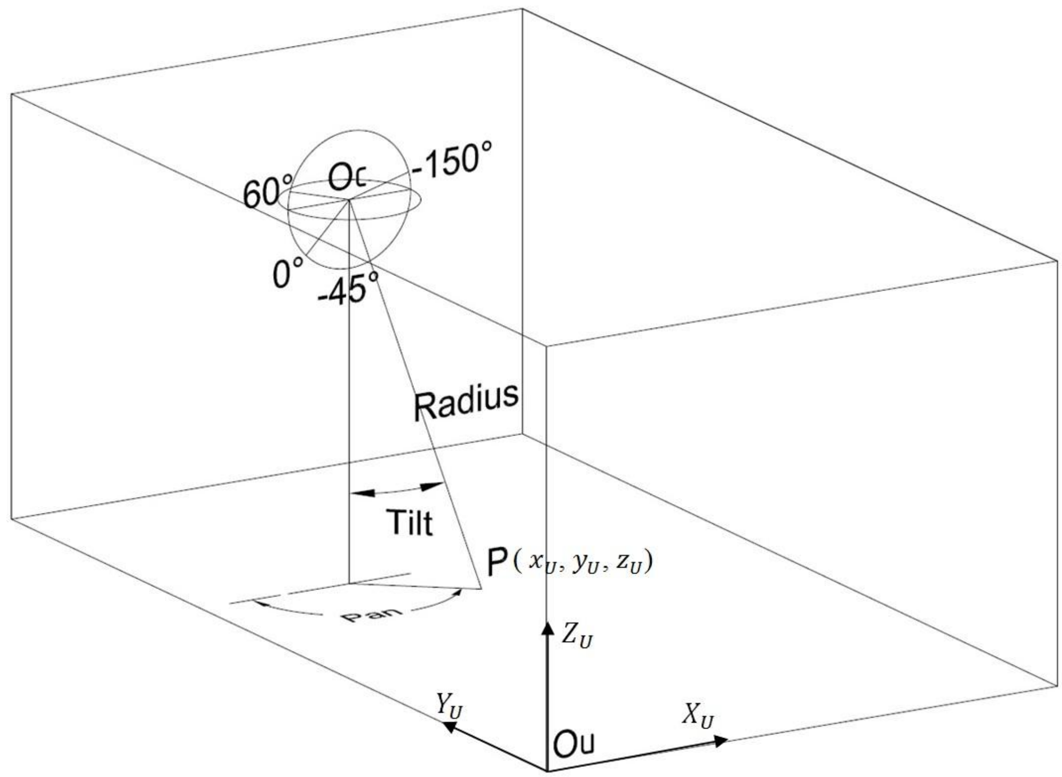


Figure 4-12 Measurement of Camera's Rotation Angles

Through the experiments and as shown in Figure 4-12, we realized that the camera's tilting origin has 45˚ deviations with respect to the downward vertical orientation that must be involved in the Equation 3-9 for calculating the correct tilt angle of rotation. Therefore, a modification is applied to Equation 3-9 in the auto-tracking application of our case study to compute the tilt angle of a point of interest as follows:

```
targetTilt = Math.Acos(targetZinCameraSys / radius) * (180 / Math.PI) - 45;
```

This code takes the shifted (using Equation 3-6) Z coordinate of the target *P* as well as the computed radius (using Equation 3-7) to calculate their *arccos.* As the C# Math.Acos function generates the result in radian, the angle is then converted to degrees and also is adjusted to the tilting origin of the camera in use with the constant -45˚.

In this preliminary test, the scenario included two tagged and one untagged persons to be monitored by the camera and the RTLS. Some video frames including all the three persons were processed using the EmguCV Library (EmguCV, 2013) for human body detection and locating. The human detection results were not good because of the top camera view. Hence, we decided to choose another open source object detection program which is learning-based and allows us for training it by some sample images from our test. Moreover, in this test the camera FoV was too limited because of the lab's low ceiling. Therefore, our main case study has been performed in another site as follows.

Site Configuration

A larger area with a higher ceiling as illustrated in Figure 4-13, located at the atrium of the 5th floor in EV building of Concordia University is used for the main case study. A wireless Ubisense cell has been set up by four Ubisense sensors which are connected to a laptop through Wi-Fi, for calibration and data collection purposes using the installed Ubisense server software (Ubisense Location Engine). A Sony PTZ camera has been mounted at the height of 9.3 m which achieves a FoV of 8.9 m × 4.87 m, where multiple persons can freely move. The FoV can be calculated from the camera's angles of view. We calculated the camera's angles of view from the first test as follows and the inverse method is used for calculation of new FoVs.

$$\propto = 2\,(\tan^{-1}(\tfrac{a}{2} \div h)) = 2\,(\tan^{-1}(\tfrac{1.57}{2} \div 3)) = 2\,(14.62) = 29.24^{\circ}$$

$$\beta = 2\,\tan^{-1}(\tfrac{b}{2} \div h) = 2\,(\tan^{-1}(\tfrac{2.87}{2} \div 3)) = 2\,(25.54) = 51.08^{\circ}$$

The camera is connected to a laptop, placed at the terrace of 7th floor and was running the camera server (Sony Realshot Manager), via an HPoE switch.

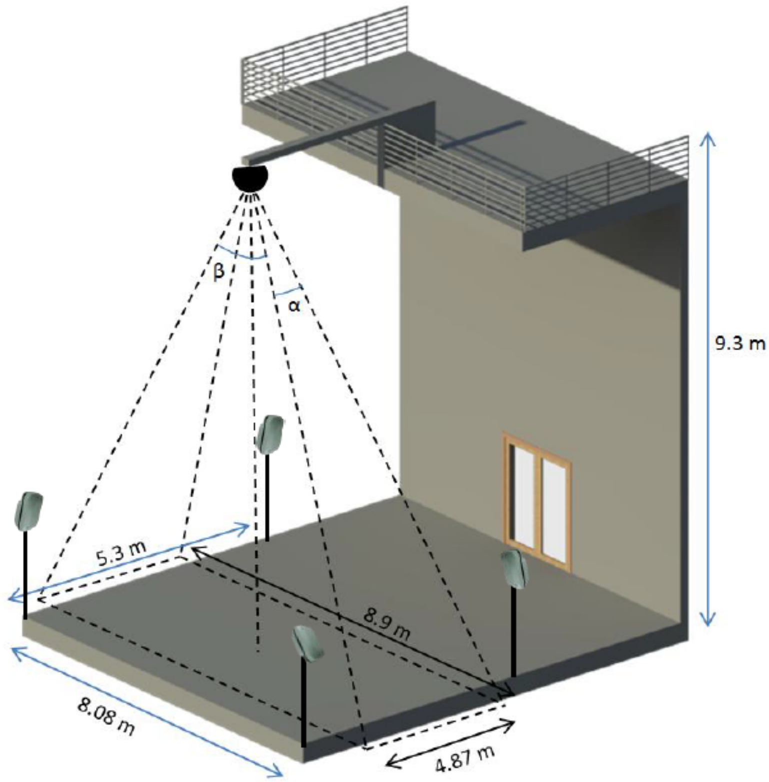


Figure 4-13 Site Configuration

Data Acquisition

The test has been carried out for three minutes of data acquisition from four persons wearing both slim and compact UWB tags and moving in and out of the overlapped (radio and video) coverage. Totally eight tags with an individual update rate of 16 Hz are registered in the Ubisense Location Engine and then are selected in the developed logger application for logging. The camera is set up for recording video with 4096 kbit/s bit rate, 30 fps frame rate, 1920 × 1080 image resolution and fixed downward orientation.

Pre-processing Video and UWB Data

Before fusion, video and UWB data must be separately processed to provide the detection and localization data that can be fused for inference about intrusions, intruders identification and locating. In the present research, as the UWB logger application gathers neat time-stamped location data of the identified tags, the above-mentioned step mostly refers to the CV-based processing for extraction of similar data from the images. Furthermore, usually some other pre-processing is required to provide the UWB and video datasets which are aligned in terms of time, frequency, and coordinates such as: data synchronization, sampling, averaging, and filtering. The pre-processing assures providing reliable and worthy input for fusion from the collected sensory data.

Video Analysis and Pre-processing

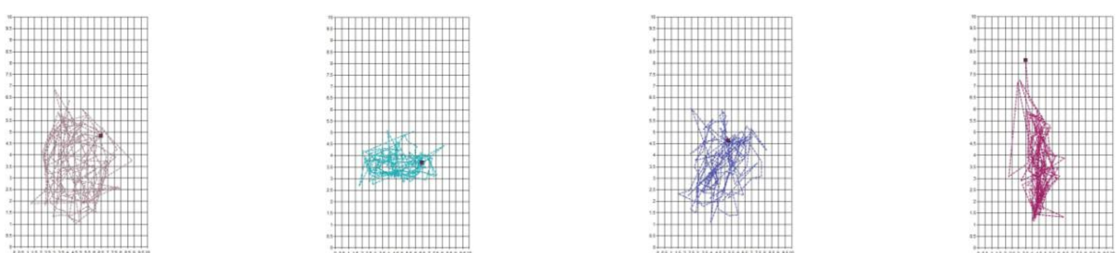
In the present case study to investigate the feasibility of the proposed approach and considering the latency of the CV analysis, we sampled video frames for an interval of fifteen seconds. Furthermore, to avoid time-consuming CV analysis for the too similar video frames, the video frames of each second are sampled by the rate of 1:5, i.e. 6 frames out of 30 are sampled from each second and totally a collection of 90 images is provided for image processing and data fusion. The images of the sampled collection are processed using a CV-based human detector which was trained with another image set from the captured video. In the training image set all the human bodies are manually annotated which generate an xml file per image containing the coordinate information of the annotated bodies. Also, some negative samples (images without any human) are added to the training image set. Then, the images and their annotation (xml) files are jointly used for training. In the present case study, the total number of positive and negative samples was 950 and this image set was divided into two sub-sets: (a) images for training and (b) images for validating the trained detector. Also, the images were resized into the 375 × 250 resolution for decreasing the training time. Upon successful validation of the trained model, a human model is generated to be used in the detector program. The case study image set (collection of 90 images) is tested with this detector for human detection purpose. The detected bodies are located within each image (with the pixel coordinates (X_I,Y_I)) and are mapped into the real-world FoV coordinates (X_U, Y_U) using Equations 3-2 and 3-3, as follows.

$$X_U = \left(\frac{X_I}{375}\right) \times 8.9\ m$$

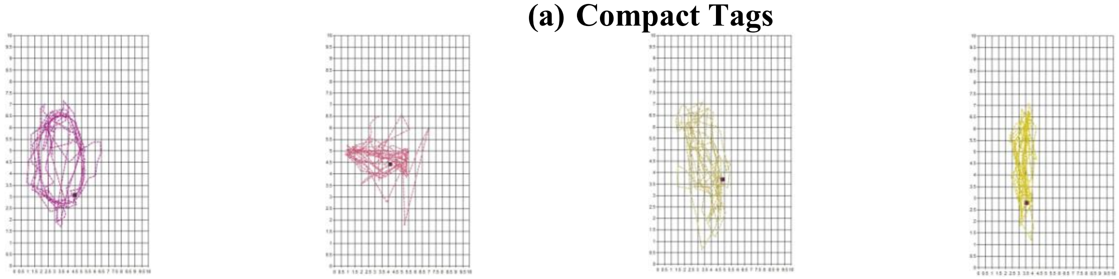
$$Y_U = \left(\frac{Y_I}{250}\right) \times 4.87m$$

UWB Data Analysis and Pre-processing

On the other hand, the same size UWB data set (90 consecutive timestamps) must be provided and pre-processed to be fused with the CV-based results. We analyzed the UWB log file in the MS Excel for identifying the tags accuracy and update rate. According to our Excel visualization results shown in Figure 4-14, the slim tags demonstrated fewer location errors and therefore their data are selected for pre-processing and fusion. Moreover, among the four slim tags, the tag with the most erroneous data is omitted. Therefore, its owner is assumed to be an untagged person who is supposed to be identified as the intruder of this case study using the data fusion program.



(a) Compact Tags



(b) Slim Tags

Figure 4-14 UWB Visualization Results for (a) Compact and (b) Slim Tags

Also, we noticed the average update rate of each individual tag was 11 Hz during the data collection phase which must be reduced to 6 Hz to provide the same-size dataset as video for the synchronized interval of 15 seconds. To sample 6 readings out of 11, it is sufficient to randomly pick one of the two consecutive readings. However, the better approach is averaging the two consecutive UWB readings and using the averaged value for fusion. This strategy not only satisfies our sampling goal (11Hz vs. 6 Hz) but also improves the UWB location accuracy. In addition, the UWB data must be sampled from the interval which is synchronized with the analyzed video sample. Synchronization is an essential step for accurate data fusion, especially if the sensor platforms are installed on different computers with different system times. After synchronizing the UWB and video streams, the sensory data must be aligned if measured in different coordinate systems and must be tailored if gathered from different size coverage fields. In our experiment, due to the limited camera FoV, the UWB coverage was bigger and its data must be filtered out so that the items falling out of the camera's FoV are omitted.

In this case study, an Excel add-in is used for UWB data averaging and animating (i.e. replaying tags' movements) together with video playback which enables finding the most accurate timestamp for synchronization with the sampled video stream. The averaged data from the synchronized stream are selected for fusion. At the end, in the selected dataset, the UWB data out of the camera FoV are filtered. The resulting UWB dataset is ready to be jointly used with the CV results for fusion.

Data Fusion Remarks

Before testing the datasets with the fusion program, we visually compared the corresponding UWB and video location coordinates using the MS Excel "scatter" plot. The results of this visual comparison are collected in Appendix G and are used later for validating the results of the fusion program. In this visual comparison, we identified different conditions which can result in different inferences and correlation errors and categorized them as follows. Also, by ranking their correlation complexity we predicated the error types that can occur. Furthermore, a few suitable strategies are applied and suggested to mitigate the errors. Two different sources of inference and correlation errors include:

(1) **A tag which is out of the camera's FoV has been erroneously positioned within the FoV by the RTLS.** This issue leads to inability of the fusion module to detect the intrusion as an erroneous tag location would be matched with a CV- located point. For example, if the CV has located three human bodies in a frame and on the other hand, due to UWB positioning error, three tags are detected within the FoV, no intrusion will be detected as all the detected persons in video are assumed to be tagged and authorized. This condition is referred as a false negative case and is illustrated in Figure 4-15.

Figure 4-15 UWB Positioning Error Causing False Negative Intrusion Detection

(2) **A tag which is within the camera's FoV is erroneously positioned out of the FoV by the RTLS.** This issue leads to false intrusion alerts since a tagged person will be considered untagged. This condition is referred as false positive case. Moreover, a more

severe condition, when multiple tags are erroneously positioned out of the FoV, leads to the identification of incorrect number of untagged persons as intruders as illustrated in Figure 4-16.

Figure 4-16 UWB Positioning Error Leading to Wrong Intruder Identification

Another inference error which can happen through the fusion is incorrect association and intruder identification due to a complex distribution of the location points. Two issues can create a complex arrangement of points in the FoV: one is when the persons are too close to each other, e.g. passing by each other, and the second is due to the positioning errors that show the location points closer than their real distance as illustrated in Figure 4-17.

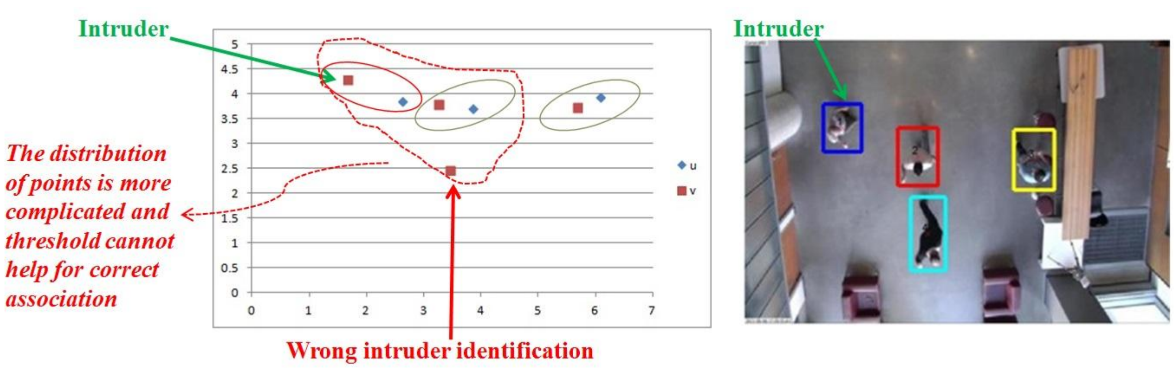


Figure 4-17 Wrong Data Association and Intruder Identification Due to UWB Positioning Errors

In these conditions the fusion module fails to correctly associate the data and find the intruder. By identifying the above-mentioned error sources and types, the following treatments are suggested for increasing the success rate of data fusion for this case study which can be applied or customized for similar conditions:

(1) Considering the fact that for this case study we had designated a single person as the intruder, any case with (length of v_t) > (length of r_t) +1, where v_t and r_t are respectively the Euclidean space vectors of video and UWB RTLS location points, is filtered to avoid generation of incorrect results about number of intruders.

(2) Excluding the severe erroneous cases, some cases can be effectively correlated by the fusion module if a rational threshold value is involved as the maximum acceptable distance among the points for being considered as the nearest neighbors of each other and being associated. By rejecting the data associations which bypass this threshold criterion, better association results will be generated. Finding a suitable threshold according to the range of the positioning errors in the dataset enables the fusion program to avoid the false negative inferences and the incorrect intruder identification errors.

Data Fusion and Sensitivity Analysis with Different Threshold Values

Data fusion in this case study was performed via the k-NN algorithm as explained in Section 4.6.4 to infer about the intrusions and identify the intruders. Correlation of the synchronized UWB and video data was performed four times using the fusion module: without threshold and with three different correlation thresholds. By definition, any correlation of the points with a distance exceeding the threshold is refused. This new criterion can help avoiding the false negative intrusion alerts, also correct identification of the intruder when only one pair of such points exists in the v_t and r_t (the video point of the refused correlation pair is identified as the intruder). However, for some cases the threshold criterion results in refusing more than one data correlation which is another type of fusion error. In these cases, still we can maintain the intrusion alerts generated for of the cases with (length of v_t) > (length of r_t) as reliable as before; i.e. true positive intrusion alert. Nonetheless, the present implementation of the data fusion module is incapable of identifying the intruder for these cases.

We performed a sensitivity analysis on our data fusion methodology to achieve the optimum success rates through choosing the best correlation threshold. Our sensitivity analysis consists of comparing and drawing the optimum results from the four different conditions as summarized in Table 4-1. The best success rates in intrusion detection and intruder identification were achieved by the threshold = 1.2 m through this sensitivity analysis. Better results may be drawn through a larger study for adopting the best threshold value. According to Table 4-1, with the most suitable threshold value (1.2 m), the success rates of 76.66% and 47.77% are achieved for intrusion detection and intruder identification, respectively. These results are validated in Appendix G using MS Excel scatter plot for visual comparison of the location points and identification of the erroneous cases.

Table 4-1 Sensitivity Analysis of Success Rate of Data Fusion

case	**Intrusion Detection**		**Intruder Identification**	
	Out of 90	**percentage**	**Out of 90**	**percentage**
Without Threshold	50	55.55%	33	36.66%
Threshold = 1.0 m	63	70.00%	34	37.77%
Threshold = 1.1 m	69	76.66%	42	46.66%
Threshold = 1.2 m	69	76.66%	43	47.77%

Furthermore, the intruder identification results can be improved by ignoring the minority errors and generalizing a correct identification (obtained for a timestamp of a second) to the whole second. As long as the intrusion identification function does not report two or more conflicting intruder identifiers (i.e. different intruders are not identified in different milliseconds of the input sampled from one second), an identification in any millisecond of one second can be considered for the whole second period. In this case study, having 90 frames and UWB detections sampled from 15 seconds of our experiment, by considering the above-mentioned amendment we further improved our identification results and achieved the success rate of 60%.

Intruder Auto-Tracking with the PTZ Camera

Intruder auto-tracking requires real-time identification of the untagged intruder, i.e. real-time fusible video and radio data which are not currently accessible due to the latency of the used CV program and the deficiency of the present prototype for fusing real-time data. Therefore, the prototype must be enhanced for being able to accomplish intruder auto-tracking as an AIITS. However, auto-tracking at present is implemented for the tagged persons' data, collected in the UWB log files. This function in fact replays the selected tags' movements, still without reflecting their actual speed; since the replaying speed is affected by the number of the coordinate transformations (from Cartesian to spherical). The future work should expand the prototype system's capabilities for on-line data fusion, using a near-to real-time CV program.

XQuery Processing

At present, the XQuery processing is implemented as a function of searching the recorded video files directory for finding those recorded during the interval that a user-selected UWB tag has been detected. In other words, although different XQueries can arise, the current prototype allows the user only to choose a tag ID from the UWB log files and for this ID calculates the total logged interval of detection. Then, the function finds the video files with the names matching the calculated interval. The function also calculates the starting and ending video frame number and appropriate media players can be used to jump into the determined frames.

4.8 Summary and Conclusions

This chapter introduced the hardware and software components of our prototype AIITS and also described the features and implementation limitations of the current prototype which consists of: the BIM visualization add-in, the UWB logger application, the camera interface and the CV program, and the data fusion module. Our case study including real-world data acquisition, data pre-processing, and data fusion demonstrated the feasibility of the proposed AIITS for indoor security. However, further enhancements are required to achieve a real-time and reliable performance, considering the following conclusions drawn from our case study:

- Data fusion accuracy can be affected by two error sources, including: (1) false positive or negative human detection of the CV program; and (2) inaccuracy of the UWB in human positioning. The CV-based human detection can be improved by training a strong model via a large image set. Also, some distinction threshold can be used for rejecting the too close detections to avoid or decrease the false positives. However, the UWB positioning errors are intrinsic and we cannot improve in the AIITS.
- The quality of the fusion results is dependent on the synchronization and alignment of the data, and it can also be improved by applying some association criteria, such as a suitable closeness threshold. Our intrusion detection and intruder identification results are improved by 21.11% and 11.11%, respectively, using the threshold = 1.2 m (which is adopted with respect to the average positioning error observed in the captured UWB data). The success rates of the current AIITS prototype are 76.66% and 47.77% for

intrusion detection and intruder identification, respectively. Further improvement was achieved in intruder identification by accepting a correct identification for the whole second when the identification failed in the rest of that second's sampled data. The final intruder identification success rate was 60% in this case study.

- The CV program used in the development of this prototype is not suitable for real-time applications and is only leveraged to investigate the applicability of the proposed AIITS.

CHAPTER 5 CONCLUSIONS AND FUTURE WORK

5.1 Summary of Research

The focus of the present research was on developing a MSDF-based methodology for *Automatic Intruder Identification and Tracking*. This methodology proposes real-time identification and tracking of the authorized persons via UWB tags which are attached to them before entering a private area of a building. In addition, a PTZ camera is used in the same area to capture video from ongoing activities and to detect the present persons who can be untagged. The number of UWB tag detections and video human body detections are compared to infer about intrusions. If an intrusion is detected, the *k*-NN algorithm is applied on the location data, reported by the UWB and CV modules, to associate the detected tags to the detected human bodies in the video in order to find the video data that cannot be associated to any UWB data. The unassociated data are marked as the intruders which will be tracked by the PTZ cameras. In addition to the sensory data fusion, using BIM is proposed in this research for several reasons. BIM has the geospatial model of the building, has the information about the usages of the rooms whether they are public or private, can store an ACL for each room and can import access constraints to the security system. Besides, BIM provides a rich simulation environment to test the coverage of the sensor networks before deployment. It can also assist in filtering out noisy location data of RTLS.

5.2 Research Contributions and Conclusions

Our main contribution in this research was developing a MSDF-based security solution which integrates the tasks of intrusion detection, intruder identification and locating, and finally intruder auto-tracking into a hybrid system. The objective of this integration is multifold:

increasing accuracy of individual sensor technology, expanding system functionality to address multiple security tasks and therefore increasing the automation level of the current security systems, and finally processing queries regarding the recorded incidents which demands content-based video indexing and retrieval (XQueries).

The following conclusions can be stated from the present research:

(1) The state-of-the-art building modeling and sensor technologies have been adopted for designing the AIITS. BIM, an UWB RTLS, and a PTZ camera are leveraged in the proposed design because of their complementary nature in implementing the proposed methodology.

(2) In this design, data fusion accuracy can be affected by two error sources, including: (1) false positive or negative human detection of the CV program; and (2) inaccuracy of the UWB in human positioning. The CV-based human detection can be improved by training a strong model via a large image set. Also, some distinction threshold can be used for rejecting the too close detections to avoid or decrease the false positives. However, the UWB positioning errors are intrinsic and we cannot improve in the AIITS.

(3) The quality of the fusion results is dependent on the synchronization and alignment of the data, and it can also be improved by applying some association criteria, such as a suitable closeness threshold. Our intrusion detection and intruder identification results are improved by 21.11% and 11.11%, respectively, using the threshold = 1.2 m (which is adopted with respect to the average positioning error observed in the captured UWB data). The success rates of the current AIITS prototype are 76.66% and 47.77% for intrusion detection and intruder identification, respectively.

5.3 Limitations and Future Work

According to the conclusions drawn from our research and the present limitations, the following work is suggested for future research:

- CV-based detection's performance must be improved, using a most efficient program, to enhance the overall results of AIITS.
- Data fusion module must be further developed for processing real-time continuous data.

REFERENCES

AGC. (2006). *The Contractors' Guide to BIM* . The Associated General Contractors of America.

Asen, Y. (2012). *Building Information Modeling Based Integration and Visualization for Facilities Management, Master thesis, Concordia University.* Montreal.

Autodesk. (2003). *BIM in Practice.* San Rafael, CA, USA: Autodesk Inc.

Autodesk. (2007). *BIM Parametric Building Modeling.* Autodesk Inc.

Autodesk. (2013). *Autodesk Revit Overview*. Retrieved from Autodesk: <http://www.autodesk.com/products/autodesk-revit-family/overview>

Axis. (2008). *IP-Surveillance Design Guide.* Axis Communications.

Axis. (2013). *q60_series*. Retrieved from Axis Communications: <http://www.axis.com/products/q60_series/index.htm>

Bahl, P., & Padmanabhan, V. (2000). RADAR: An In-Building RF-Based User Location and Tracking System. *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM, Volume:2*, (pp. 775-784).

Bekkali, A., Sanson, H., & Matsumoto, M. (2007). RFID Indoor Positioning Based on Probabilistic RFID Map and Kalman Filtering. *Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communication (WiMOB)*, (pp. 21-21). White Plains, NY.

Berclaz, J., Fleuret, F., Turetken, E., & Fua, P. (2011). Multiple Object Tracking Using K-Shortest Paths Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, (pp. 1806-1819).

Black, J., Ellis, T., & Rosin, P. (2002). Multi View Image Surveillance and Tracking. *Motion and Video Computing, 2002. Proceedings of the Workshop on Motion and Video Computing*, (pp. 169-174). U.K.

Cai, Y., de Freitas, N., & Little, J. (2006). *Robust Visual Tracking for Multiple Targets.* Springer.

Cerrada, C., Salamanca, S., Perez, E., Cerrada, J., & Abad, I. (2007). Fusion of 3D Vision Techniques and RFID Technology for Object Recognition in Complex Scenes. *IEEE International Symposium on Intelligent Signal Processing (WISP)*, (pp. 1-6). Madrid.

Chen, C., Chen, T., Lin, J., & Wang, D. (2011). People Tracking in the Multi-Camera Surveillance System. *Second IEEE International Conference on Innovations in Bio-inspired Computing and Applications (IBICA)*, (pp. 1-4). Kaohsiung, Taiwan.

Choi, H., Park, U., & Jain, A. (2010). PTZ Camera Assisted Face Acquisition, Tracking & Recognition. *IEEE Biometrics: Theory Applications and Systems (BTAS)*, (pp. 1-6). Seoul, South Korea.

Choi, W., & Savarese, S. (2010). *Multiple Target Tracking in World Coordinate with Single, Minimally Calibrated Camera.* Berlin Heidelberg: Springer.

Collins, R., Lipton, A., & Kanade, T. (2000). *Introduction to the Special Section on Video Surveillance.* IEEE Transactions on Pattern Analysis and Machine Intelligence.

Collins, R., Lipton, A., Fujiyoshi, H., & Kanade, T. (2001). Algorithms for Cooperative Multisensor Surveillance. *Proceedings of the IEEE*, (pp. 1456-1477). Pittsburgh, PA, USA.

CTIAI. (2013). Retrieved from CTIAI: <http://www.ctiai.com/en/about.html>

DARPA. (2014) Retrieved from DARPA: <http://www.darpa.mil/NewsEvents/Releases/2011/2011/06/23_DARPA_advances_video_analysis_tools.aspx>

Delannay, D., Danhier, N., & De Vleeschouwer, C. (2009). Detection and Recognition of Sports (wo)men from Multiple Views. *Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on Distributed Smart Cameras*, (pp. 1-7). Louvain-la-Neuve, Belgium.

Denis, B., CEA, L. G., Ouvry, L., Uguen, B., & Tchoffo-Talom, F. (2005). Advanced Bayesian Filtering Techniques for UWB Tracking Systems in Indoor Environments. *IEEE International Conference on Ultra-Wideband*, (pp. 6-6). Z r ich.

Dibitonto, M., Buonaiuto, A., Luca Marcialis, G., Muntoni, D., Maria Medaglia, C., & Roli, F. (2011). *Fusion of Radio and Video Localization for People Tracking.* Berlin Heidelberg: Springer.

Du, W., & Piater, J. (2007). *Multi-Camera People Tracking by Collaborative Particle Filters and Principal Axis-Based Integration.* Berlin Heidelberg: Springer .

Elmenreich, W. (2002). *An Introduction to Sensor Fusion.* Austria: Vienna University of Technology.

EmguCV. (2013). Retrieved January 1, 2013, from Emgu: http://www.emgu.com/wiki/index.php/Main_Page

Felzenszwalb, P., Girshick, R., McAllester, D., & Ramanan, D. (2010). Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (pp. 1627-1645).

Fleuret, F., Berclaz, J., Lengagne, R., & Fua, P. (2008). Multicamera People Tracking with a Probabilistic Occupancy Map. *IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume:30 , Issue:2*, (pp. 267-282).

Freeman, R., Moore, L., Garcı´a A´lvarez, L., Charlett, A., & Holmes, A. (2013). Advances in Electronic Surveillance for Healthcare-Associated Infections in the 21st Century: A Systematic Review. *Elsevier, Journal of Hospital Infection, Volume:84, Issue:2*, (pp. 106-119).

Gezici, S., Kobayashi, H., Poor, H., & Molisch, A. (2005). Performance Evaluation of Impulse Radio UWB Systems with Pulse-Based Polarity Randomization in Asynchronous Multiuser Environments. *IEEE Transactions on Signal Processing, Volume:53, Issue:7*, (pp. 2537-2549).

Golston, J. (2004). Comparing Media Codecs for Video Content. *Embedded Systems Conf.* San Francisco.

Hammad, A., Setayeshgar, S., Zhang, C., & Asen, Y. (2012). Automatic Generation of Dynamic Virtual Fences as PArt of BIM-Based Prevention Program for Construction Safety. *In proceeding of IEEE Winter Simulation Conference.* Berlin, Germany.

Hampapur, A., Sharat, P., Senior, A., Ying-Li, T., Brown, L., & Bolle, R. (2003). Face Cataloger: Multi-Scale Imaging for Relating Identity to Location. *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance*, (pp. 13-20). Hawthorne, NY, USA.

Hariharan, S. (2006). *Improving Dynamic Decision Making Through RFID: A Partially Observable Markov Decision Process (POMDP) for RFID-enhanced Warehouse Search Operations.* Oklahoma State University.

Hensworth, S. (2011). Building Information Modelling and Security Design. *The Australian Building Services Journal, Volume:2*, (pp. 44-45).

Hightower, J., & Borriello, G. (2001). Location Systems for Ubiquitous Computing. *Computer, Volume:34, Issue:8*, (pp. 57-66).

Hongyu, G., Weiyao, L., Xiaokang, Y., Hongxiang, L., Ning, X., Jun, X., & Yan, L. (2011). A New Network-Based Algorithm for Multi-Camera Abnormal Activity Detection. *IEEE International Symposium on Circuits and Systems (ISCAS)*, (pp. 361-364). Shanghai, China.

Hsu H., Cheng Z., Huang T., Han Q., (2006). Behavior Analysis with Combined RFID and Video Information. *Ubiquitous Intelligence and Computing* (pp 176-181).

IBM. (2013). *CPLEX Optimizer*. Retrieved from IBM: http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/index.html

Ingram, S., Harmer, D., & Quinlan, M. (2004). UltraWideBand Indoor Positioning Systems and Their Use in Emergencies. *IEEE Position Location and Navigation Symposium*, (pp. 706-715).

ISO/IEC. (1999). *Information technology – Generic Coding of Audio-Visual Objects- Part2: Visual.* Vancouver.

Israd, M., & MacCormick, J. (2001). BraMBLe: A Bayesian Multiple-Blob Tracker. *Proceedings of the Eighth IEEE International Conference on Computer Vision (ICCV)*, (pp. 34-41). CA, USA.

Iwase, S., & Saito, H. (2004). Parallel Tracking of All Soccer Players by Integrating Detected Positions in Multiple View Images. *IEEE Proceedings of the 17th International Conference on Pattern Recognition*, (pp. 751-754). Yokohama, Japan.

Jia, S., Sheng, J., Chugo, D., & Takase, K. (2007). Human Recognition Using RFID Technology and Stereo Vision. *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, (pp. 1488-1493). Tokyo.

Jian, J. (2011). *Development of a Web Service for Visualizing the Data of Ultra Wideband Real-Time Location System, Master thesis, Concordia University.* Montreal, Canada.

Jiang, H., Fels, S., & Little, J. (2007). A Linear Programming Approach for Multiple Object Tracking. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07)*, (pp. 1-8). Vancouver, Canada.

Keni, B., Florian, V., & Rainer, S. (2007). Automatic Person Detection and Tracking using Fuzzy Controlled Active Cameras. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* , (pp. 1-8). Minneapolis, MN, Germany.

Khaleghi, B., Khamis, A., O. Karray, F., & N. Razavi, S. (2011). Multisensor Data Fusion: A Review of the State-of-the-Art. *Information Fusion*, (pp. 28-44).

Khan, S., & Shah, M. (2009). Tracking Multiple Occluding People by Localizing on Multiple Scene Planes. *IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume:31, Issue:3*, (pp. 505-519).

King, T., Kopf, S., Haenselmann, T., Lubberger, C., & Effelsberg, W. (2006). COMPASS: A Probabilistic Indoor Positioning System Based on 802.11 and Digital Compasses. *Proceedings of the 1st ACM International Workshop on Wireless Network Testbeds, Eperimental Evaluation & Characterization (WiNTECH)*, (pp. 34-40). New York, NY, USA.

Kolias, V., Giannoukos, I., Anagnostopoulos, C., Anagnostopoulos, I., Loumos, V., & Kayafas, E. (2010). Integrating RFID on Event-based Hemispheric Imaging for Internet of Things Assistive Applications. *3rd International ACM Conference on PErvasive Technologies Related to Assistive Environments.* New York, NY, USA.

Krishnan, S., Sharma, P., Guoping, Z., & Woon, O. (2007). A UWB based Localization System for Indoor Robot Navigation. *IEEE International Conference on Ultra-Wideband*, (pp. 77 - 82). Singapore.

Kristan, M., Perš, J., & Kovačič, S. (2009). Closed-world Tracking of Multiple Interacting Targets for Indoor-sports Applications. *Computer Vision and Image Understanding* (pp. 598-611). Elsevier.

Kulyukin, V., Gharpure, C., Nicholson, J., & Pavithran, S. (2004). RFID in Robot-Assisted Indoor Navigation for the Visually Impaired. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, (pp. 1979-1984). UT, USA.

Lee, H., Wicke, M., Kusy, B., & Guibas, L. (2008). Localization of Mobile Users Using Trajectory Matching. *Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, (pp. 123-128). New York, NY, USA.

Lefèvre S., Holler J., Vincent N. (2003). A Review of Real-time Segmentation of Uncompressed Video Sequences for Content-based Search and Retrieval. *Real-Time Imaging, Volume:9, Issue:1,* (pp 73-98).

Liu, H., Darabi, H., Banerjee, P., & Jing, L. (2007). Survey of Wireless Indoor Positioning Techniques and Systems. *Part C: Applications and Reviews, IEEE Transactions on Systems, Man, and Cybernetics, Volume:37, Issue:6* , (pp. 1067-1080).

Liu, Y., & Wang, L. (2004). RFID and RTLS Technology and Application. *Applied Science and Technology*, (pp. 51-53).

Loy, C., Tao, x., & Shaogang, G. (2009). Modelling Activity Global Temporal Dependencies using Time Delayed Probabilistic Graphical Model. *IEEE 12th International Conference on Computer Vision*, (pp. 120-127). London, UK.

Luca, M., Stefano, P., Andrea, T., Deborah, M., & Carlo, S. (2005). Cooperative Multisensor System for Real-Time Face Detection and Tracking in Uncontrolled Conditions. *Image and Video Communications and Processing.* San Jose, CA: SPIE Proceedings.

Luo, R., Yih, C., & Su, K. (2002). Multisensor Fusion and Integration: Approaches, Applications, and Future Research Directions. *Sensors Journal, IEEE, Volume:2, Issue:2*, (pp. 107-119).

Magee, D. (2004). Tracking Multiple Vehicles Using Foreground, Background and Motion Models. *Elsevier, Volume:22, Issue:2*, (pp. 143-155).

Mandeljc, R., Kovaˇciˇc, S., Kristan, M., & Perš, J. (2013). Tracking by Identification Using Computer Vision and Radio. *Sensors, Volume:13, Issue:1*, (pp. 241-273).

Martins, E., & Pascoal, M. (2003). A New Implementation of Yen’s Ranking Loopless Paths Algorithm. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies, Volume:1, Issue:2*, (pp. 121-133).

Microsoft. (2013). *Microsoft.Net*. Retrieved from Microsoft: http://www.microsoft.com/net

Miller, L. (2003). *Why UWB? A Review of Ultrawideband Technology.* Gaithersburg, Maryland: NIST.

Mittal, A., & Davis, L. (2003). A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene. *International Journal of Computer Vision, Volume:51, Issue:3*, (pp.189-203).

Motamedi, A. (2009). *Framework For Lifecycle Management of Facilities Components Using RFID Technology, Master thesis, Concordia University, Montreal, Canada.*

Munoz, D., Lara, F. B., Vargas, C., & Enriquez-Caldera, R. (2009). *Position Location Techniques and Applications*. Academic Press.

NewJerseySolutions. (2013). *IP Versus Analog Cameras*. Retrieved from New Jersey Solutions: <http://www.njsolutions.net/faqs-cctv/42-cctv/74-ip-versus-analog-cameras>

Ni, L., Liu, Y., Lau, Y. C., & Patil, A. (2003). LANDMARC: Indoor Location Sensing Using Active RFID. *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, (pp. 407-415). Fort Worth, TX.

Okuma, K., Taleghani, A., De Freitas, N., Little, J., & Lowe, D. (2004). A Boosted Particle Filter: Multitarget Detection and Tracking. *Springer, Volume:3021*, (pp. 28-39).

Oncam. (2013). *Oncam Grandeye IPZoom*. Retrieved from oncamgrandeye: <http://www.oncamgrandeye.com/29-oncam-grandeye-ipzoom.html>

OpenCV Answers. (2013). Retrieved from OpenCV: http://answers.opencv.org/question/13982/2d-pixel-coordinate-to-3d-world-coordinate/

Otsuka, K., & Mukawa, N. (2004). Multiview Occlusion Analysis for Tracking Densely Populated Objects Based on 2-D Visual Angles. *Proceedings of the 2004 IEEE Computer*

Society Conference on Computer Vision and Pattern Recognition, (pp. 1-90). Atsugi, Japan.

Papapostolou, A., & Chaouchi, H. (2011). RFID-Assisted Indoor Localization and the Impact of Interference on its Performance. *Journal of Network and Computer Applications*, (pp. 902-913).

Peennebaker, W. B., & Mitchell, J. L. (1993). *JPEG Still Image Data Compression.* Springer.

Piccardi, M. (2004). Background Subtraction Techniques: A Review. *IEEE International Conference on Systems, Man and Cybernetics, Volume:4*, (pp. 3099-3104). Sydney, NSW, Australia.

Pirsiavash, H., Ramanan, D., & Fowlkes, C. (2011). Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 1201-1208). Irvine, CA, USA.

Priyantha, N., Chakraborty, A., & Balakrishnan, H. (2000). The Cricket Location-Support System. *Proceedings of the 6th Annual International ACM Conference on Mobile Computing and Networking (MobiCom)*, (pp. 32-43). New York, NY, USA.

Prosser, B., Gong, S., & Xiang, T. (2008). Multi-camera Matching under Illumination Change Over Time. *Inria Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications - M2SFA2.* Marseille, France.

Rinner, B., Winkler, T., Schriebl, W., Quaritsch, M., & Wolf, W. (2008). The Evolution From Single to Pervasive Smart Cameras. *Second ACM/IEEE International Conference on Distributed Smart Cameras*, (pp. 1-10).

RTLS Market Overview. (2006). Retrieved from rfid.net: http://rfid.net/basics/rtls/246-rtls-market-overview

Sankaranarayanan, K., & Davis, J. (2011). PTZ Camera Modeling and Panoramic View Generation via Focal Plane Mapping. In *Computer Vision – ACCV 2010* (pp. 580-593). Columbus, OH, USA: Springer Berlin / Heidelberg.

Saxena, A., & Y. Ng, A. (2013). *Make3D*. Retrieved from Make3D: http://make3d.cs.cornell.edu/index.html

Song, J., Haas, C., & Caldas, C. (2006). Tracking the Location of Materials on Construction Job Sites. *Journal of Construction Engineering and Management, Volume:132, Issue:9*, (pp. 911-918).

Sony. (2008). *Intelligent Monitoring Software.* Japon: Sony Corporation.

Sony. (2011). *Network Camera.* Japon: Sony Corporation.

Security Cameras. (2014) Retrieved January 27, 2014, from Sony: <http://pro.sony.com/bbsc/ssr/cat-securitycameras/>

Sony SNC-ER580. (2012). Retrieved January 9, 2013, from Sony: http://pro.sony-asia.com/product/resources/en_BA/images/Brochure/Security/SNC-ER580.pdf

Steggles, P., & Gschwind, S. (2005). The Ubisense Smart Space Platform. *Adjunct Proceeding of the Third International Conference on Pervasive Computing*, (pp. 73-76).

Stelios, M., Nick, A., Effie, M., Dimitris, K., & Thomopoulos, S. (2008). An Indoor Localization Platform for Ambient Assisted Living Using UWB. *Proceedings of the 6th International ACM Conference on Advances in Mobile Computing and Multimedia*, (pp. 178-182).

Strauchs, J. (2012). *What Is BIM and Should You Care?* Retrieved from securityinfowatch: http://www.securityinfowatch.com/article/10672781/hot-trends-what-is-bim-and-should-you-care

Taok, A., Kandil, N., & Affes, S. (2009). Neural Networks for Fingerprinting-Based Indoor Localization Using Ultra-Wideband. *Journal of Communications, Volume:4, Issue:4*, (pp. 267-275).

Tronci, R., Murgia, G., Pili, M., Ristori, M., Sirena, N., & Roli, F. (2013). *The Ambient Intelligence Laboratory of the Sardegna District.* Pula, CA.

Ubisense. (2008). *Location Engine Config User Manual.* Ubisense.

Ubisense. (2012). Retrieved from Ubisense: http://www.ubisense.net/en/resources/factsheets/

Ubisense. (2013). Retrieved from Ubisense: http://www.ubisense.net/en/media/pdfs/factsheets_pdf/47329_ubisense_location_platform_en110131.pdf

Ubisense RTLS. (2012). Retrieved from Ubisense: http://www.ubisense.net/en/resources/factsheets/ubisense-real-time-location.html

VideoSurveillance. (2013). *Video Surveillance newsletter*. Retrieved from VideoSurveillance: <http://www.videosurveillance.com/newsletters/june-2010.asp>

Wang, C. S., & Cheng, L. C. (2011). RFID & Vision Based Indoor Positioning and Identification System. *IEEE 3rd International Conference on Communication Software and Networks (ICCSN)*, (pp. 506-510). Xi'an.

Want, R., Hopper, A., Flacao, V., & Gibbons, J. (1992). The Active Badge Location System. *ACM Transactions on Information Systems (TOIS)*, (pp. 91-102).

Ward, A. (2007). *Ubisense RTLS Accuracy Review.* IEEE.

White, F. (1991). *Data Fusion Lexicon, Joint Directors of Laboratories.* San Diego: Technical Panel for C3, Data Fusion Sub-Panel, Naval Ocean Systems Center.

Wikipedia. (2013a). *Spherical Coordinate System*. Retrieved from Wikipedia: <http://en.wikipedia.org/wiki/Spherical_coordinate_system>

Wikpedia. (2013b). *atan2*. Retrieved from Wikipedia: <http://en.wikipedia.org/wiki/Atan2>

Wu, B., & Nevatia, R. (2006). Tracking of Multiple, Partially Occluded Humans Based on Static Body Part Detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (pp. 951 - 958). California, USA.

Xu, M., Orwell, J., & Jones, G. (2004). Tracking Football Players with Multiple Cameras. *International IEEE Conference on Image Processing*, (pp. 2909 - 2912). UK.

Yabuta, K., & Kitazawa, H. (2008). Optimum Camera Placement Considering Camera Specification for Security Monitoring. *IEEE International Symposium on Circuits and Systems*, (pp. 2114 - 2117). Seattle, WA.

Zegelin C., (2003). *U.S. Patent* Application 10/648, 712.

Zelniker, E., Gong, S., & Xiang, T. (2008). Global Abnormal Behaviour Detection Using a Network of CCTV Cameras. *The Eighth International Workshop on Visual Surveillance (inria).* Marseille, France.

Zhang, C. (2010). *Improving Crane Safety by Agent-Based Dynamic Motion Planning using UWB Real-Time Location System, Ph.D Thesis, Concordia University.* Montreal, Canada.

Zhang, C., Kuhn, M., Merkl, B., Fathy, A., & Mahfouz, M. (2006). Accurate UWB Indoor Localization System Utilizing Time Difference of Arrival Approach. *IEEE Radio and Wireless Symposium*, (pp. 515 - 518). San Diego, CA.

Zhang, S., & Yuan, F. (2006). RFID Technique and its Application in Safety Management System for People and Vehicle in Mine Shafts. *Southern Metals*, (pp. 12-14).

Zhang, W., Cheung, S., & Chen, M. (2005). Hiding Privacy Information in Video Surveillance System. *Proceedings of the 12th IEEE International Conferance on Image Processing*, (pp. 868-871).

Appendix A: Specifications of Ubisense Series 7000 Tags and Sensors

Table A-1 Specifications of Ubisense Series 7000 Slim Tags

	Compact Tags	Slim Tags
Size & Weight	Height 40 mm (1.5 inches) Width 40 mm (1.5 inches) Depth 15 mm (0.6 inches) Weight30 g (1 Ounce)	Height 83 mm (3.26 inches) Width 42 mm (1.65 inches) Depth 10 mm (0.39 inches) Weight35 g (1.2 Ounce)
Operating Temperature	Standard -20 C to 60 C (-4 F to 140 F) Extended -30 C to 70 C (-22 F to 158 F)	Standard -20 C to 60 C (-4 F to 140 F) Extended -30 C to 70 C (-22 F to 158 F)
Mounting Options	Multiple including screw mounts, cable ties, self adhesive and mounting plate	Multiple including mounting plate and attachment to personnel badges
Operating Frequencies	Ultra-wide band: 6 GHz – 8 GHz Telemetry channel: 2.4 GHz	Ultra-wide band: 6 GHz – 8 GHz Telemetry channel: 2.4 GHz
Beacon Rate	Dynamically variable (from 1 per minute to 10 per second)	Dynamically variable (from 1 per minute to 10 per second)

Table A-2 Specifications of Ubisense Series 7000 Sensors

Size and Weight	
Dimensions	20cm x 13cm x 6cm (8” x 5” x 2.5”)
Weight	650g (23 oz)
Operating Conditions	
Temperature	0°C to 60°C (32°F to 140°F)
Humidity	0 to 95%, non-condensing
Enclosure	
IP30	
Location Performance	
Operating Range	Up to 160m (520ft) in open field conditions
Achievable Accuracy	Better than 30cm (12”) in 3D
Radio Frequencies	
Ultra-wideband	6GHz – 8GHz
Telemetry channel	2.4GHz
Certifications	
FCC Part 15 (FCC ID SEASENSOR20)	
EU CE	
Power Supply	
Power-over-Ethernet IEEE 802.3af compatible 12V DC @ 10W (optional)	
Mounting Options	
Adjusting mounting bracket (supplied)	
Ubisense Part Codes	
UBISENSOR7000, UBISENSPS (optional 12V power supply)	

Appendix B: SONY SNC-ER-580 Specifications

Table B-1 Sony SNC-ER-580 Specifications

SNC-ER580	
Camera	
Image device	1/2.8-type Exmor CMOS
Number of effective pixels	Approx. 3.27 Megapixel
Minimum Illumination	Color: 1.7lx (F1.6, shutter 1/30sec, AGC ON, 50IRE[IP]) B/W: 0.3lx (F1.6, shutter 1/30sec, AGC ON, Night Mode, 50IRE[IP]) Color: 1.2lx (F1.6, shutter 1/30sec, AGC ON, 30IRE[IP]) B/W: 0.18lx (F1.6, shutter 1/30sec, AGC ON, Night Mode, 30IRE[IP])
Electronic shutter speed	1/1 to 1/10,000 s
Gain control	Auto/Manual (-3 to 28 dB)
Exposure control	Full auto, Shutter priority, Iris priority, Manual
White balance mode	Auto, ATW, Indoor, Outdoor, One-push, Manual, Sodium vapour lamp
Lens type	Auto-focus Zoom Lens
Zoom ratio	Optical zoom 20x, Digital zoom 12x, Total zoom 240x
Horizontal viewing angle	55.4 to 2.9 degrees
Focal length	f=4.7 to 94.0 mm
F-number	F1.6 to F3.5
Minimum object distance	10mm (wide) to 800mm (tele)
Pan angle	360 degrees endless rotation
Pan Speed	300 degrees/s (max)
Tilt angle	210 degrees(with e-flip)
Tilt speed	300 degrees/s (max)
Preset position	256 positions
Tour	5
Camera Features	
Day/Night	Yes
Wide-D	Yes*1 (86 dB)
Noise Reduction	Yes
Image	
Codec image size (H x V)	1920x1080, 1680x1056, 1280x1024, 1440x912, 1280x960, 1376x768, 1280x800, 1280x720, 1024x768, 1024x576, 800x600, 800x480, 768x576, 720x576, 704x576, 720x480, 640x480, 640x368, 384x288, 320x240, 320x192
Video compression format	H.264, MPEG-4, JPEG
Codec streaming capability	Dual streaming
Maximum frame rate	30 fps (H.264) / 20 fps (MPEG-4) / 16 fps (JPEG)
Audio	
Audio compression	G.711/G.726
Scene Analytics	
Intelligent motion detection	Yes

*1 When using Wide-D technology, the maximum frame rate will be 15fps.

Network	
Protocols	IPv4, IPv6, TCP, UDP, ARP, ICMP, IGMP, HTTP, HTTPS, FTP (client/server), SMTP, DHCP, DNS, NTP, RTP/RTCP, RTSP, SNMP(MIB-2)
ONVIF conformance	Yes (Ver. 1.02)
Number of clients	5
Authentication	IEEE802.1X
Interface	
Ethernet	10BASE-T / 100BASE-TX (RJ-45)
Card slots	SD memory card x 1 (Compatible with the SD/SDHC standards)
Sensor input	x2
Alarm output	x1
External microphone input	Mini-jack (monaural)
Audio line output	Mini-jack (monaural), Max ouput level: 1 Vrms
General	
Mass	1.7 kg (3 lb 12 oz) (including sealing bracket)
Dimensions (ø x H mm)	ø147.4 x 190.9 mm (5 7/8 x 7 5/8 inches)
Power requirements	HPoE (IEEE802.3at compliant), AC24V
Power consumption	Approx. 25W
Operating temperature	-5 to +50 °C (23 to 122°F)
Starting temperature	0 to 50 °C (32 to 122°F)
Storage temperature	-20 to +60 °C (-4 to +140°F)
Safety regulation	UL2044, FCC 15B Class A, IC Class A, IEC60950-1, EN55022(A)+EN55024+EN50130-4, VCCI Class A, C-Tick Class A
System Requirements	
Operating system	Microsoft Windows XP(32bit) - Professional Edition Microsoft Windows Vista(32bit) - Ultimate, Business Edition Microsoft Windows 7 (32/64bit) - Ultimate, Professional Edition
Processor	Intel Core2 Duo 2.33 GHz or higher
Memory	2 GB or more
Web browser	Microsoft Internet Explorer Ver. 6.0, Ver. 7.0, Ver. 8.0 Firefox Ver.3.5 (Plug-in free viewer only) Safari Ver.4.0 (Plug-in free viewer only) Google Chrome Ver.4.0 (Plug-in free viewer only)
Supplied Accessories	
	CD-ROM (User's guide, supplied programs), Installation manual, Ceiling bracket, Screws(2), Template, 24V AC Connector, I/O Connector

*The SNC-ER580 includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

Appendix C: Initial IP Camera Setup via SNC Toolbox Application

The server must meet the system requirements mentioned in Table B-1. After installation, open the SNC toolbox and in the main window click on "search" to detect the connected cameras. Figure C-1 is illustration of main window of SNC toolbox program.

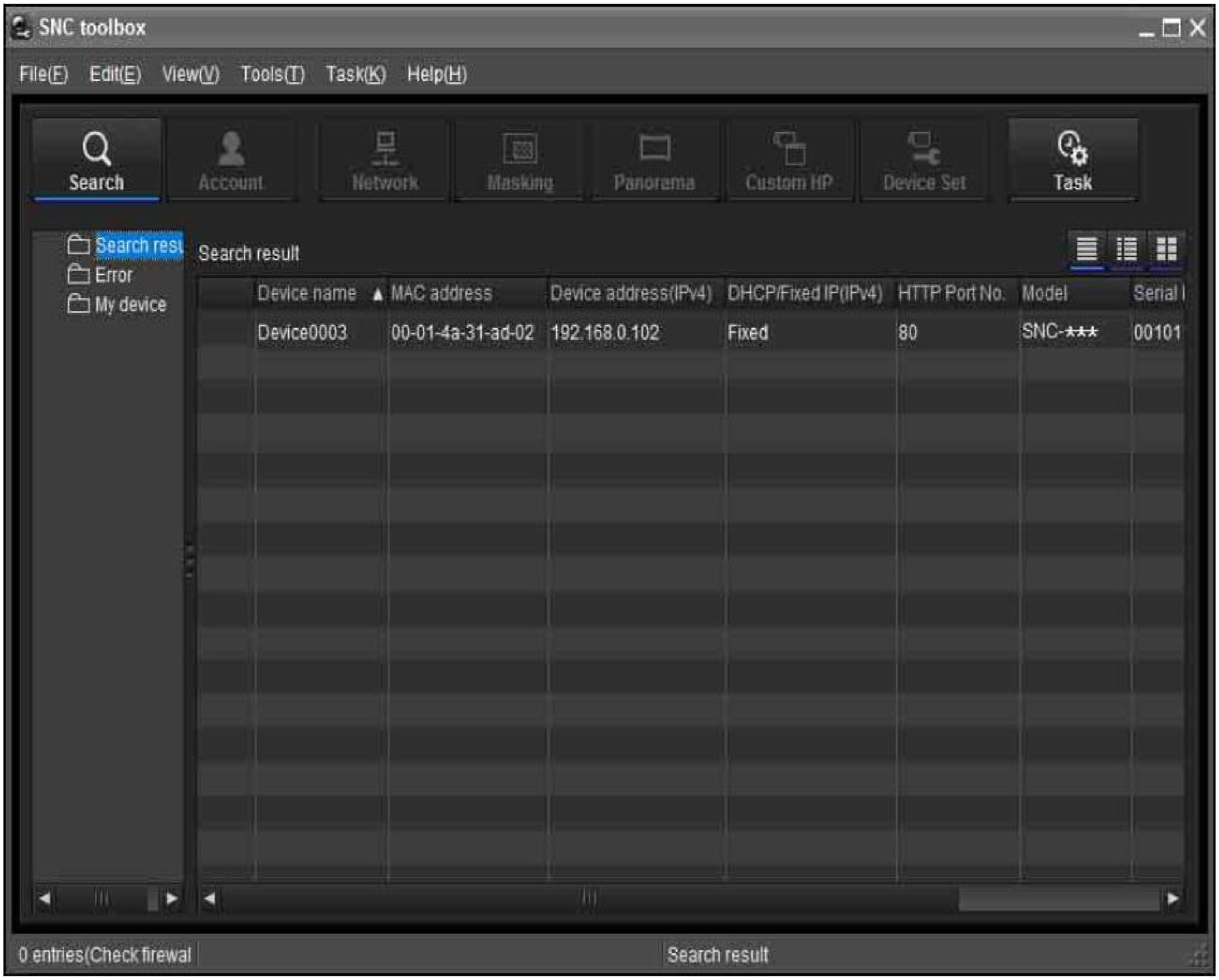


Figure C-1 SNC Toolbox Interface

By selecting the camera from the list and entering administrator username and password in the window displayed in Figure C-2.

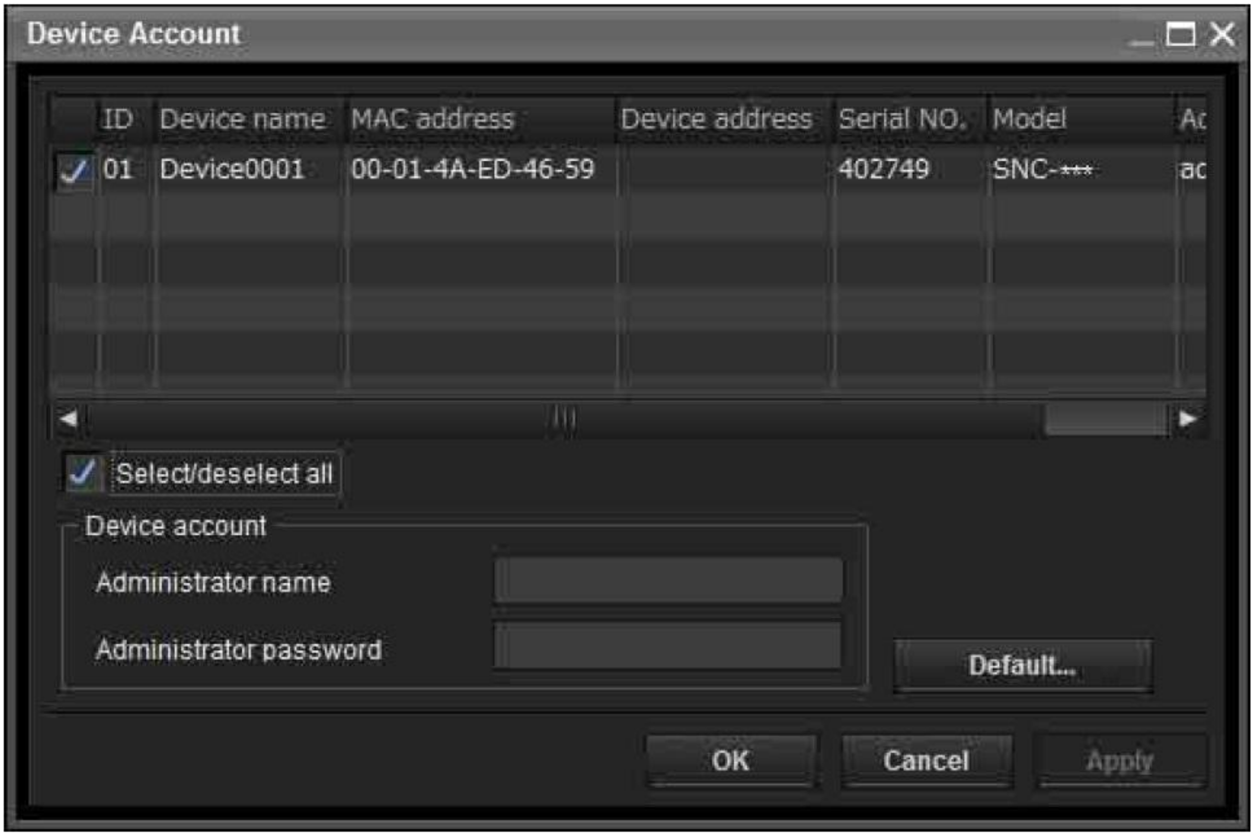


Figure C-2 Camera Access Account

Set the IP address, DNS server address and Http port number in the window shown in Figure C-3. If a DHCP server is running on the network, user can choose "obtain IP address automatically" and/or "obtain DNS server address automatically".

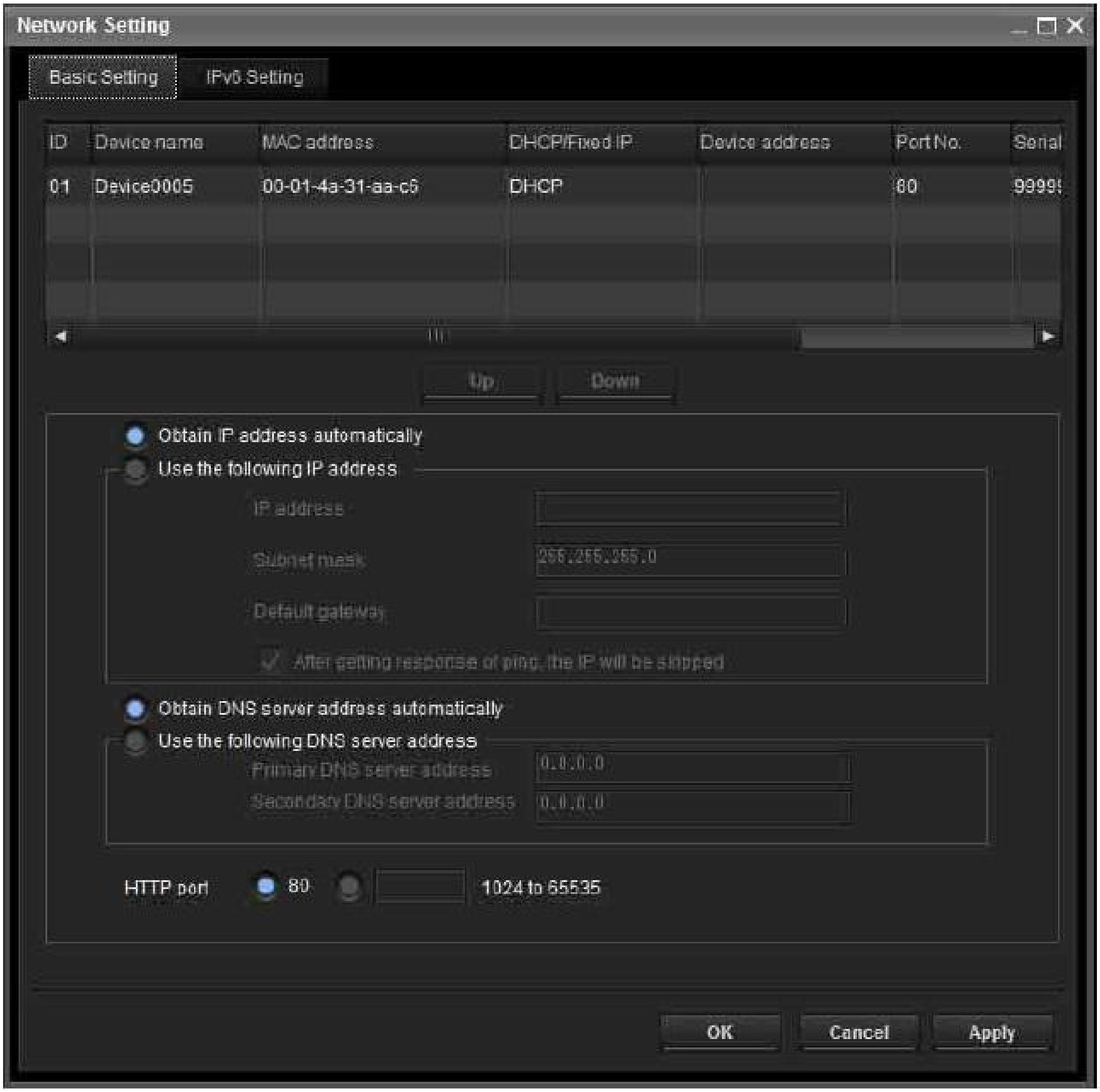


Figure C-3 Network Setting for Camera Access

When setting is finished, confirm the entered information and double-click the device name to access the camera directly. Also by entering the assigned IP address in the URL address bar of the compatible web browser, the viewer screen of the network camera is displayed on the web browser as shown in Figure C-4.

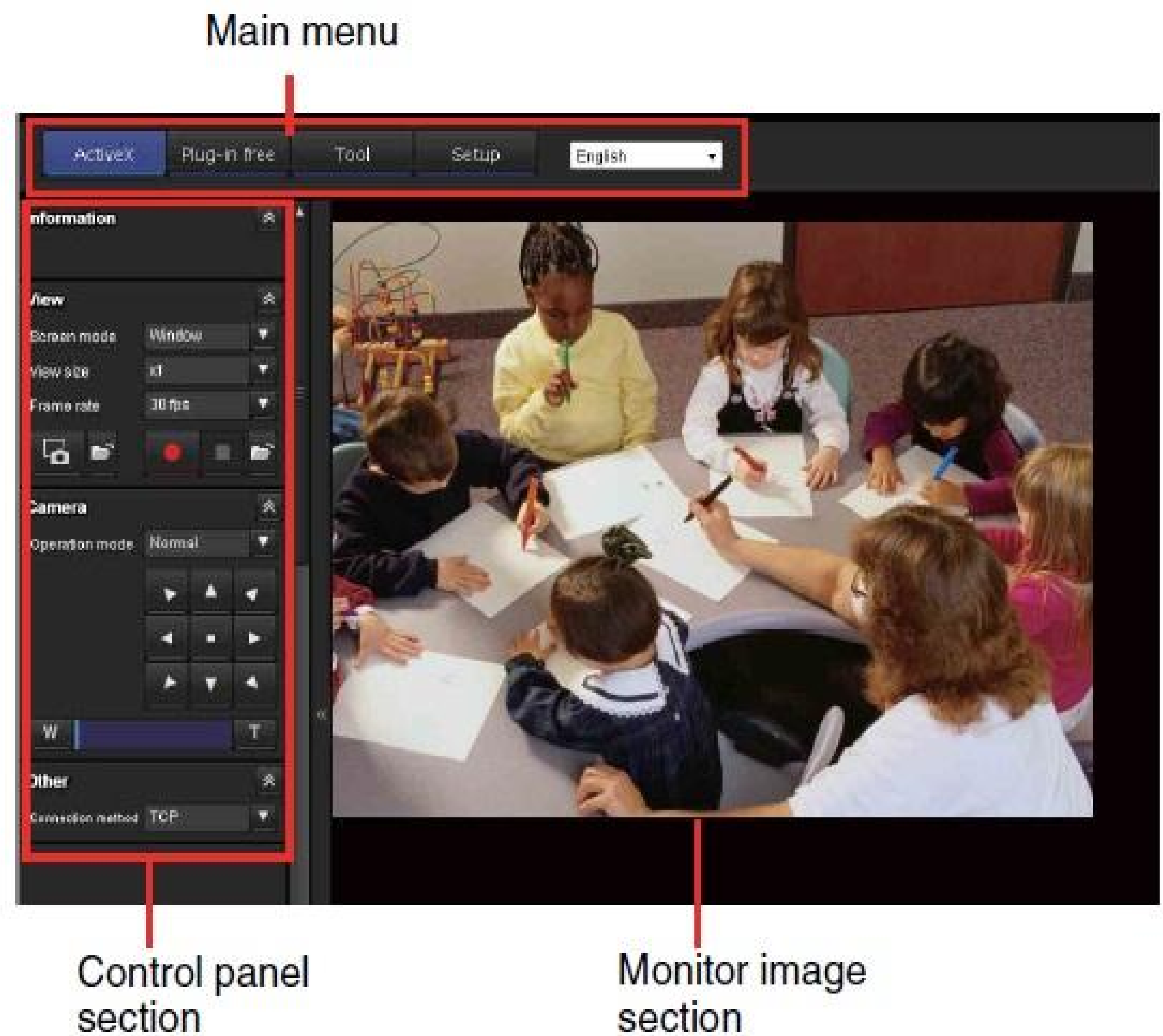


Figure C-4 Camera Viewer Screen in the Web Browser

After first time camera setup via the SNC toolbox application, user also can use the web browser itself to modify the initial settings. The Setup menu of the web browser display provides complete list of setup options that enables user for easy and quick setup while monitoring.

Appendix D: Implementation Details of the BIM Plug-in

Revit plug-in uses the Revit API and adds a functionality, usually a command, which is not directly provided by the software and will be loaded as an "External Tool" into the Revit environment as illustrated in Figure D-1.

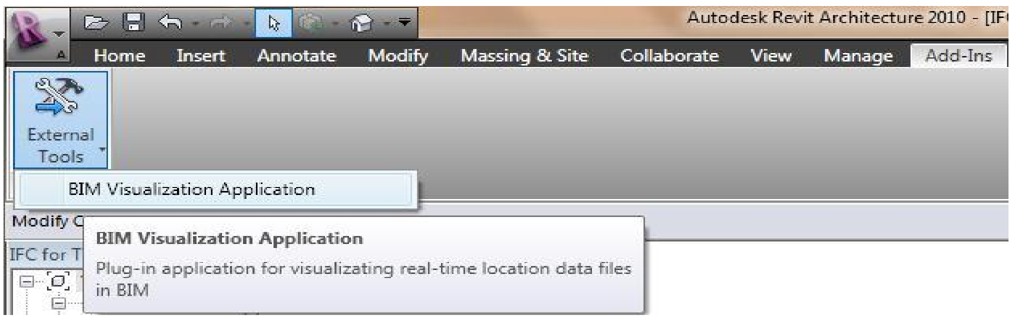


Figure D-1 BIM Visualization Application as an "External Tool" for Revit Architecture (Jian, 2011)

BIM Visualization application is developed as a C# Class Library project within the Visual Studio 2010. This project contains the following classes:

(1) **Command.cs:** this class contains the Command function which is similar to the main function in some programming languages, as it is the starting point of the execution of the code. This class must be defined in Revit.ini.

(2) **Creator.cs:** this class generates the dots and traces from the text data related to the coordinates.

(3) **DataManager.cs:** this class manages the storage of different data types recorded in the input log file.

(4) **FileReading.cs:** this class imports the location data from the selected log file.

(5) **MainForm.cs:** this is the class which includes the code written or automatically generated for designing the main form, GUI, of the plug-in.

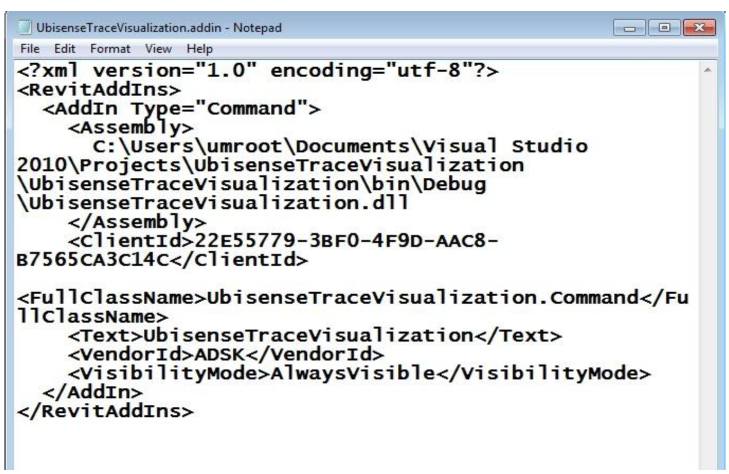
The following steps are taken in development of this plug-in application:

(1) The required DLL files are added to the created project from the Revit installation sub-folder in order to expose the required Revit APIs to the project.

(2) The coding starts with adding the “using” statements for the required Revit namespaces.

(3) After implementing the desired command within the “execute” method and completing the library class code, the project first must be saved and built in the Visual Studio.

(4) The built application must be first registered in Revit Architecture by creating a manifest which introduces the plug-in to Revit. The manifest file is located in a specific location which is checked by Revit and includes some information about loading and running the plug-in. Figure D-2 illustrates the manifest of the BIM Visualization plug-in. After creating the manifest, by starting the Revit software the plug-in is also loaded. On the Add-Ins ribbon tab, click the External Tools drop-down list then click BIM Visualization Application.




```
<?xml version="1.0" encoding="utf-8"?>
<RevitAddIns>
  <AddIn Type="Command">
    <Assembly>
      C:\Users\umroot\Documents\Visual Studio
2010\Projects\UbisenseTraceVisualization
\UbisenseTraceVisualization\bin\Debug
\UbisenseTraceVisualization.dll
    </Assembly>
    <ClientId>22E55779-3BF0-4F9D-AAC8-
B7565CA3C14C</ClientId>

<FullClassName>UbisenseTraceVisualization.Command</Fu
llClassName>
    <Text>UbisenseTraceVisualization</Text>
    <VendorId>ADSK</VendorId>
    <VisibilityMode>AlwaysVisible</VisibilityMode>
  </AddIn>
</RevitAddIns>
```

Figure D-2 Revit Plug-in Manifest File

Appendix E: Implementation Details of the UWB Module

UWB logger includes some client Ubisense Schemas that connect to the Ubisense platform in order to query about the detected tags' names and types. The obtained name and type information are shown to the user for selection. Logging is enabled in two options: (1) via manual start/stop buttons; and (2) via a logging interval which the user inputs. Starting the logger through each of the above-mentioned options for the selected tag names will generate the same result for an identical logging period. The tags' location information is logged in a text file based on their recorded timestamps. The text file can be used by other applications for analysis and trajectory visualization (e.g. BIM Visualization Plug-in, XQuery processing and so forth). The UWB module is developed as a C# Windows form application within Visual Studio 2010. The Form1.cs class includes both GUI design code and event handling methods of the Logger application.

(a) Logging location data captured by RTLS platform

First the required Ubisense DLLs are referenced from the Logger project. In the "Solution Explorer" window of the Visual Studio 2010, by right clicking on the "References" of the Logger application and choosing "add reference", we added the following Ubisense DLLS to our project:

- UbisenseLocationEngine.dll which contains modules to configure and use the actual real-time location engine hardware.
- UbisenseLocationServices.dll which contains the modules which manage the site tasks such as spatial monitoring and cell extents.
- UbisensePlatform.dll which contains the definition of core Ubisense objects.

- UbisenseVisualization.dll which contains the modules which give the visualization functionality.

Certain Ubisense Namespaces also must be linked to the Logger application via "using" statements, as shown in Figure E-1, to enable: (a) getting tags' types and names, (b) visualizing the RTLS cell, (c) logging the tags' location updates. This will allow instantiating the client schemas which connect to the Ubisense (or user created) services available on the Ubisense system and query the required information.

```
using Ubisense;
using Ubisense.ULocation;
using Ubisense.USpatial;
using Ubisense.UBase;
using Ubisense.UVis;
using Naming = Ubisense.UName.Naming;
using Vis = Ubisense.UVis.Representation;
using Monitor = Ubisense.USpatial.Monitor;
using Building = Ubisense.UBuilding;
using CellData = Ubisense.ULocation.CellData;
using Uspatial = Ubisense.USpatial;
using Config = Ubisense.UCell.Config;
```

Figure E-1 Using Ubisense DLLs and Schemas

The used Ubisense Namespaces introduce some client schemas to the project scope. However, schema objects which we intend to use must be declared in the form1.cs class similar to the example shown in Figure E-2.

```
namespace UbisenseApp1
  {
    public partial class Form1 : Form
    {
      public Form1()
      {
        InitializeComponent();
      }
      // Declarations of the objects we need
      private Naming.Schema namingSchema; }
  }
```

Figure E-2 Declaring Ubisense Client Schemas in the Application Namespace

Then, the schemas must be instantiated and connected to the Ubisense system to allow query of information. This will be explained through the methods and update handlers which are introduced in following.

getTypes() and getNames() Methods

The Ubisense.UName.Naming schema holds the ObjectName relation which allows querying the types and names of the Ubisense objects. For connecting the declared client schema to the Ubisense platform, we will create a simple "OnLoad" event for the designed form, which is executed when the application starts, and will add the code in Figure E-3 into it (to do this, return to the Form View where you see the GUI. In the right column click the lightning bolt icon to get a list of events. Double click the Load event and it will automatically drop and add the event code to yours).

```
private void Form1_Load(object sender, EventArgs e)
  {
    // Instantiate the objects
    namingSchema = new Ubisense.UName.Naming.Schema(false);

    // Connect the cached naming schema as client to read transactions
    namingSchema.ConnectAsClient();
  }
```

Figure E-3 Instantiating and Connecting the Client Schemas to the Ubisense Server

We need to iterate over the ObjectName relation of our namingSchema object to load the "Object Types" and "Object Names" checked list boxes with any Ubisense objects found when the form is loaded. We handle these two functions by two separate methods called: getTypes() and getNames() which are invoked from the LoadForm() method.

We need to get a ReadTransaction from the naming schema. Any row of the type ObjectName.RowType we find in this table will contain an object and a name for that object, which we will add to the corresponding checked list boxes. It is important to wrap ReadTransaction in a using statement, otherwise the ReadTransaction is not disposed, and the lock is not released until it is garbage collected (this blocks subsequent events indefinitely). The getTypes() method is implemented as Figure E-4 shows.

```
private void getTypes(){
        using (Naming.ReadTransaction names =
           naming_schema.ReadTransaction())
        {
           foreach (Naming.ObjectName.RowType row in
              Naming.ObjectName.name_(names))
           {
              String type = row.object_.DynamicType.ToString();
              if (!typescheckedListBox.Items.Contains((System.Object)type))
                 typescheckedListBox.Items.Add(type);
              if (!typescheckedListBox.Items.Contains((System.Object)type))
                 typescheckedListBox.Items.Add(type); } }
 }
```

Figure E-4 Body of getTypes() Method

The namingSchema object should be disposed of by the garbage collector when it falls out of scope. Go back to the properties of the form in the form design window and select the lightning bolt icon again. Double click the FormClosed event. In the code of the event disconnect and dispose the schema as Figure E-5 shows.

```
private void Form1_FormClosed(object sender, FormClosedEventArgs e)
  {
    namingSchema.Disconnect();
    namingSchema.Dispose();
  }
```

Figure E-5 Client Schema Disposal

Log the Tags' Location Updates

A location update handler is added to the application in order to monitor the tags movements and keep track of each set of coordinates through querying the cellData schema. The update handler is declared in the following way in the LoadForm method:

```
Ubisense.ULocation.CellData.Location.AddUpdateHandler(cellData, CaptureOnUpdate);
```

Each time a tag movement is detected the "CaptureOnUpdate" update handler is invoked and the (x, y, z) coordinates of the selected tags are captured and transferred to the logger method along with their recorded timestamps.

The logger method writes its input data to a StreamWriter object which is used for appending the received text to the log file. By pressing the "Stop" button or elapse of the user-entered log interval, the StreamWriter closes the text file and is disposed.

(b) Visualizing the RTLS Cell

A similar approach is taken for declaring, instantiating, and connecting the client schemas as shown in Figure E-6, which enable location visualization on a map created on the GUI. Ubisense.UVis.Representation and Ubisense.UBuilding schemas are used for this purpose.

```
// to be used by getArea method,in order to query building locations
private static Building.Contents.Schema building_schema = new
Ubisense.UBuilding.Contents.Schema(false);
// The cell data schema allows us to query locations
private static Ubisense.ULocation.CellData.Schema cellData = new
Ubisense.ULocation.CellData.Schema(false);
```

Figure E-6 Creating Building_schema and cellData Schemas

In the getArea() method a MapModel is specified and set to be used by the visualization object. It also creates a combo box control for area changes which are handled by an AreaChangeHandler. We need to get a ReadTransaction from the building-schema which iterates over the AreaProperties relation in order to build and load the area into the MapModel. Ubisense enables both 2D and 3D views of the area on the created map.

(c) XQuery Processing

XQuery processing in this thesis is implemented for handling a query in the following form: "Retrieve the video frames which are recorded during the time the tag ID "x" has been detected." For processing such a query, the UWB data are obtained from the input log file which can be generated using the developed UWB logger. XQuery processing in this application starts with opening an UWB logging file which user has selected to parse its contents for extracting the tag names. These names are listed in the "Tag ID" panel as radio buttons which restrict the user for a single selection at a time. The first and last line logged about the selected tag ID are found in the file as the starting and ending timestamps of the tag detection. The start timestamp is searched within the archive of video files and the files with names presenting equal or bigger timestamp than the search term, the detection interval, are returned to the user. For a video file which user selects the starting and ending frame number is calculated with respect to the extracted RTLS timestamps. This frame number calculation and frame retrieval are implemented in a Matlab code which is linked to our .Net application.

Appendix F: Implementation Details of the Video Module

F.1 Camera Application Implementation Procedure

(a) Camera Connection and Real-time Monitoring

Using the Sony camera SDK, the following procedure must be taken to implement camera connection and video monitoring functions:

(1) To communicate with the camera over the network, a handle must be opened via *SNC Core Library*. Therefore, first the *SNC Core Library* must be initialized by sncOpenCameraHandle. Also, the SNC Stream Library must be initialized by sncInitialize in order to play stream captured by the connected camera.

(2) Then the parameters of the two libraries must be set. Network information of camera such as IP address and port number are set by sncSetNetwork. Video and audio settings such as getting video and audio stream are set by sncEnableVideoStream, sncEnableVideoPlay, sncEnableAudioStream and sncEnableAudioPlay.

(3) Live streaming is started by calling sncStart.

(4) For teardown, sncStop is called and the libraries are finalized by sncFinalize and sncCloseCameraHandle.

In order to use the above-mentioned libraries, first a reference to these libraries in the SDK must be added to the Visual Studio solution. The snccomdef_dotnet, snccore_dotnet, sncstrm_dotnet DLL files are added to the references and the following lines are added to the program:

```
using Sony.SNC.CommonDef;
using Sony.SNC.Core;
using Sony.SNC.StreamLib;
```

(b) Video Recording

To integrate the video recording feature to the application, the following procedure is used:

(1) For the opened camera handle, recording video and audio streams are implemented via the same initialized SNC Stream Library. Two more video and audio settings are required:

```
StreamLib.EnableVideoDecode(sncHandle, false);
StreamLib.EnableAudioDecode(sncHandle, false);
```

(2) Set callback from SNC Stream Library to get the compressed data. This function is the key action of this procedure which buffers the captured video/audio to be written in a file.

(3) A timer is used for recording by interval of 500 milliseconds.

(4) When the user stops the recording, the recording timer is stopped and then a binary writer writes all the buffered video and audio to the user defined video file.

(5) In this application the camera handle is not closed after stopping the recording, therefore there is no need to teardown and finalize the libraries.

(c) Recorded File Playback

Playback function can be integrated to the application through the following procedure:

(1) Set parameters for the two libraries; the recorded file information such as video and audio codec can be set by:

```
Core.SetMemory(sncHandles[1], ref codecInfo, frameRate);
```

Video and audio setting such as display video and stop displaying video can be set by:

```
StreamLib.EnableVideoStream(sncHandles[1], enableVideo);
StreamLib.EnableVideoPlay(sncHandles[1], enableVideo);
StreamLib.EnableAudioStream(sncHandles[1], enableAudio);
StreamLib.EnableAudioPlay(sncHandles[1], enableAudio);
```

(2) Create a new thread for playback of video from recorded files.

(3) Write compressed video and audio using following the statements in the thread procedure.

```
StreamLib.WriteVideoStream(sncHandles[1], videoPtr, writeSize, out writtenSize);
StreamLib.WriteAudioStream(sncHandles[1], audioPtr, writeSize, out writtenSize);
```

(4) Teardown the playback by aborting the thread.

The whole procedure of SNC Stream Library is summarized into Figure F-1.

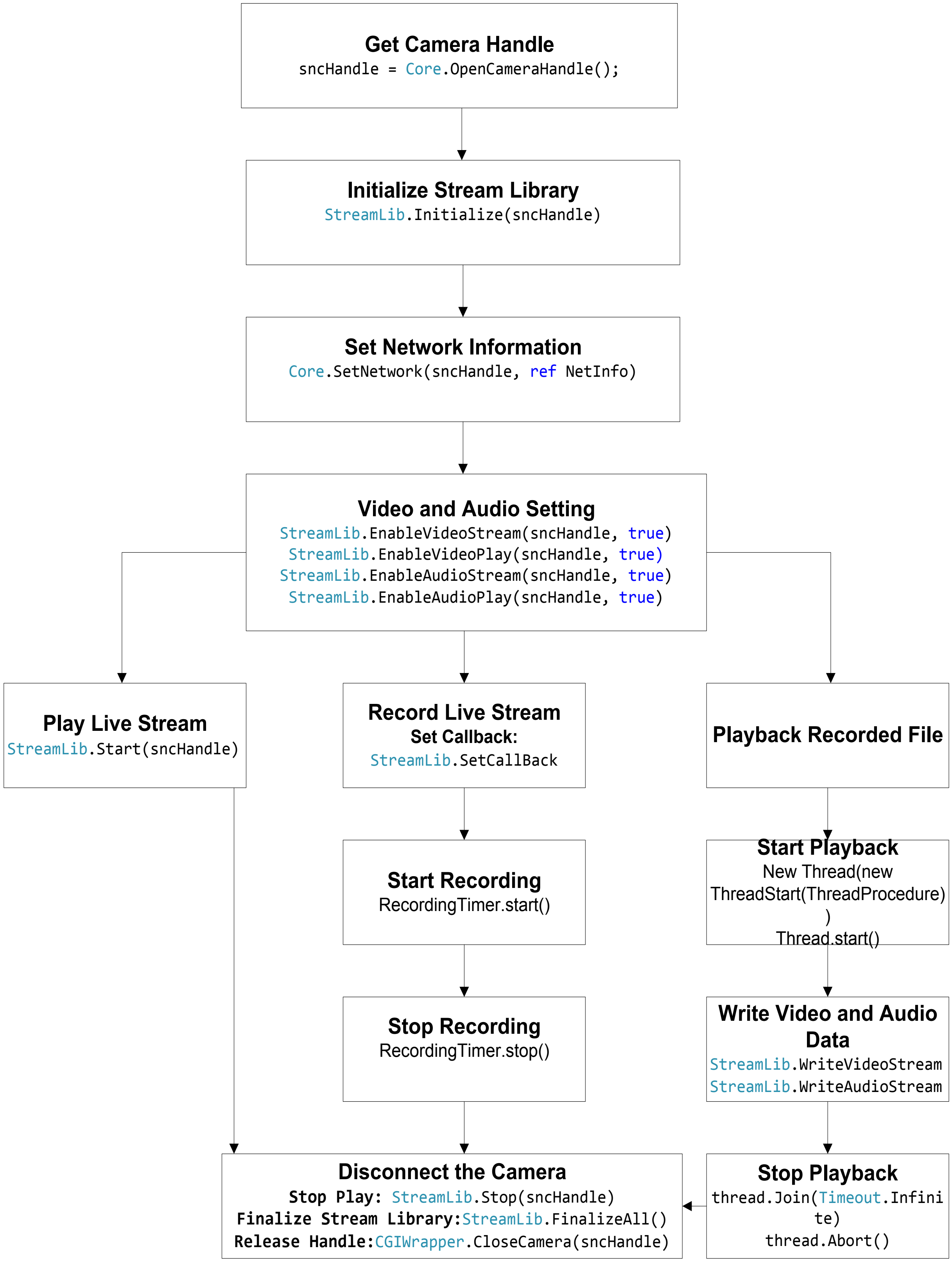


Figure F-1 Flowchart of Implementing Features Provided by Sony Stream Library APIs

(d) Pan/Tilt/Zoom Control

There are two ways of PTZ controlling by using two different SDK libraries:

(1) Control by *SNC CGIWrapper* Library

(2) Control by *SNC JoyStick* Library

We used CGIWrapper Library in the development by adding a reference to this library as: snccgiw_dotnet.dll and the following statement:

```
using Sony.SNC.CGIWrapper;
```

There are 2 different APIs in *SNC CGIWrapper* Library, one is like sncwSetXXX, and the other is like sncwSetXXXEx. We used following commands from the first API, sncwSetXXX.

For relative move:

```
CGIWrapper.SetMovePanTiltRelative(sncHandles[0], targetPan, targetTilt, speed);
```

For Absolute move:

```
CGIWrapper.SetMovePanTiltAbsolute(sncHandles[0], targetPan, targetTilt, speed);
```

For continuous move:

```
CGIWrapper.SetMovePanTiltZoomEx(sncHandles[0], panspeed, tiltspeed, zoomspeed);
```

And to stop the continous move

```
CGIWrapper.SetMovePanTiltZoom(sncHandles[0], panspeed, tiltspeed, zoomspeed);
```

With panspeed, tiltspeed, zoomspeed = 0 values.

(e) Automatic PTZ Control for Auto-Tracking a Target

Although the ultimate objective of the application is real-time auto-tracking of the identified intruder, the current version only implements auto-tracking trajectory of one UWB tag from the logged data.

In programming we used the function "*atan2*" to compute the pan angle. In a variety of computer languages, the function "*atan2*" is the arctangent function with two arguments. The two arguments *atan2* function, involves the signs of the inputs in order to return the appropriate quadrant of the computed angle, which is not possible for the single-argument arctangent function. For any real number (e.g., floating point) arguments x and y not both equal to zero, atan2(y, x) is the angle in radians between the positive x-axis of a plane and the point given by the coordinates (x, y) on it. The angle computed in radians then is converted to degrees to be sent to the PTZ camera API. The angle is positive for counter-clockwise angles (upper half-plane, $y > 0$), and negative for clockwise angles (lower half-plane, $y < 0$). Figure F-2 illustrates values of *atan2* in different quadrants of a unit circle.

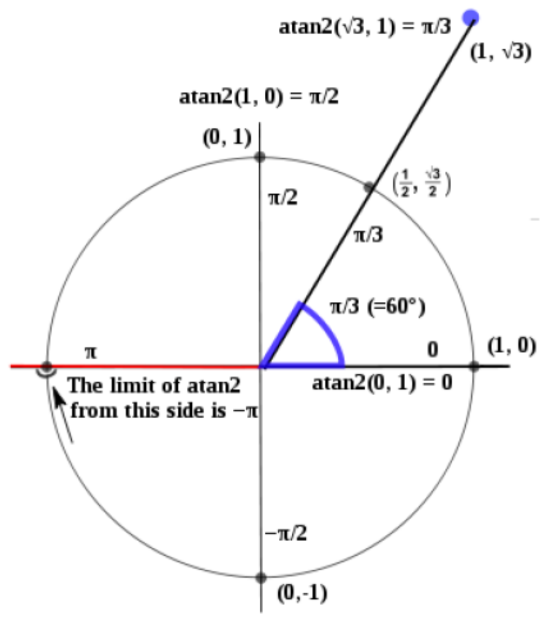


Figure F-2 Values of atan2 Function in Different Quadrants (Wikpedia, 2013b)

The implemented feature uses the following command in order to direct the camera to each location logged for the selected UWB tag.

```
CGIWrapper.SetMovePanTiltAbsolute(sncHandles[0], targetPan, targetTilt, speed);
```

The problem which this line of code causes for the tracking application is the camera display is not updated during the rotation until the camera reaches the point and stops. This prevents monitoring and recording of the travelled path by the UWB tag. However, to relieve this problem we used following code to force the application to update the screen after each successful camera rotation in the loop of calculating angles from the input Cartesian coordinates.

```
Application.DoEvents();
```

F.2 CV program Implementation

Human Locating

The following code is applied to the images having the resolution of 375 × 250 and our case study in a FoV with the width of 4.87 m and length of 8.9 m.

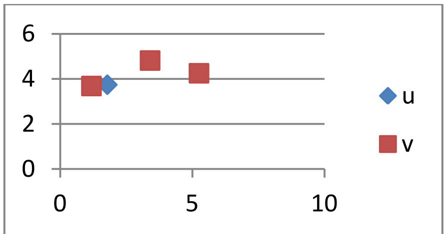
```
xcWorld = (xc/375)*8.9;
ycWorld = (yc/250)*4.87;
worldRectangleCenters (:,1) = xcWorld;
worldRectangleCenters (:,2) = ycWorld;
```

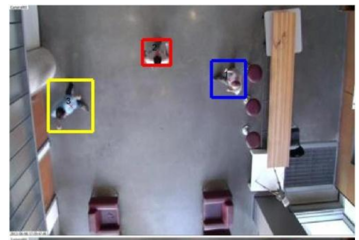
Appendix G: Scatter Plots of the Case study Positioning Results

Table G-1 Validating Fusion Results Using MS Excel Scatter Plot

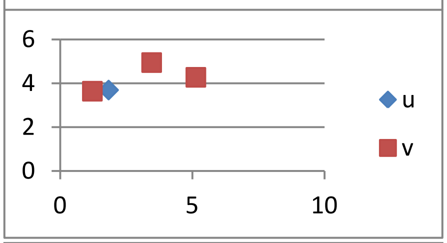
Frame No.

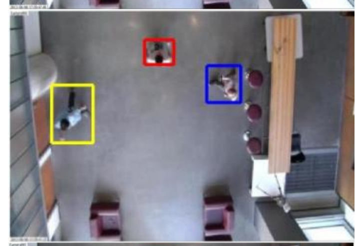
1895



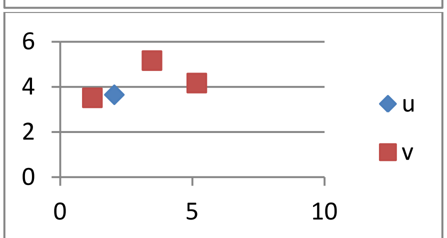


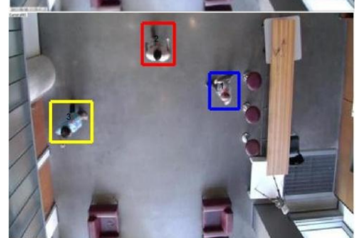
1900



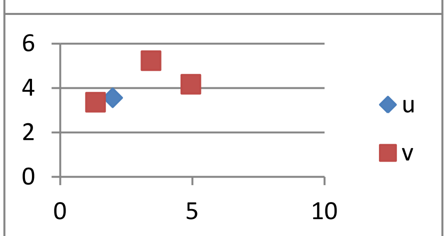


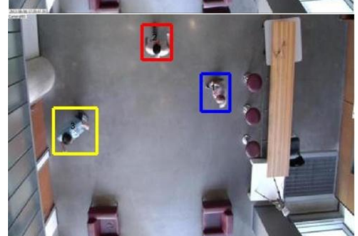
1905



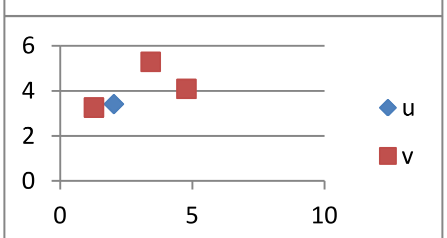


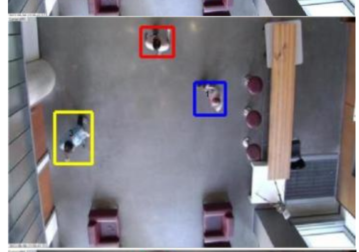
1910



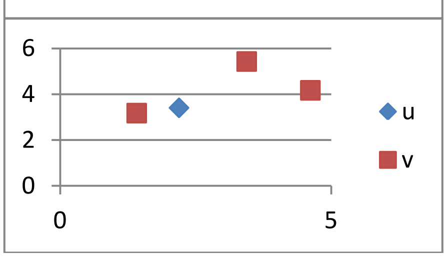


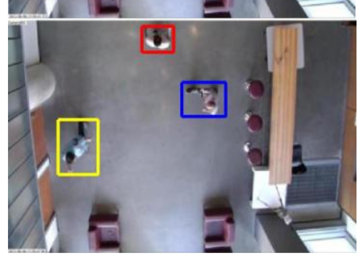
1915



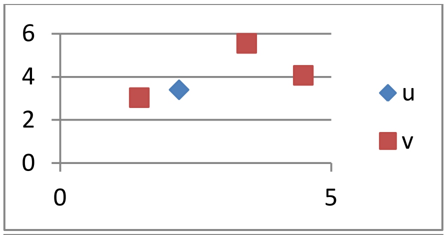


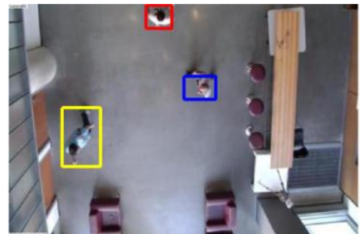
1920



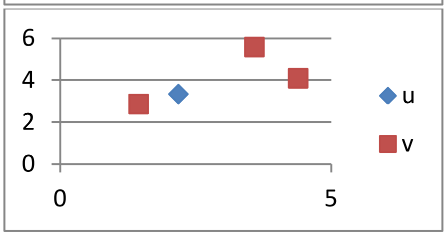


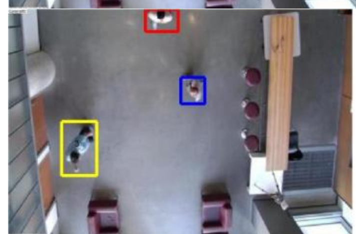
1925



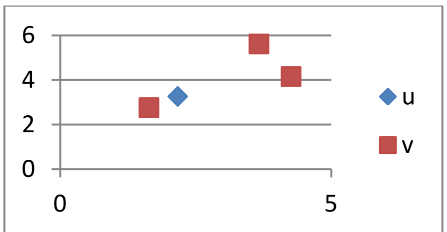


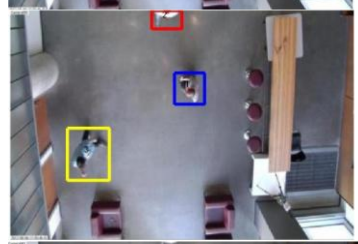
1930



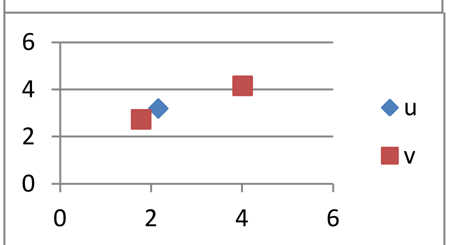


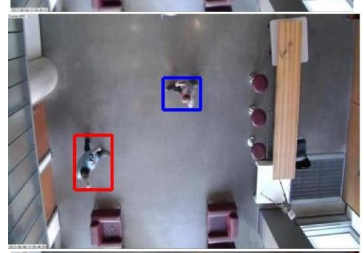
1935



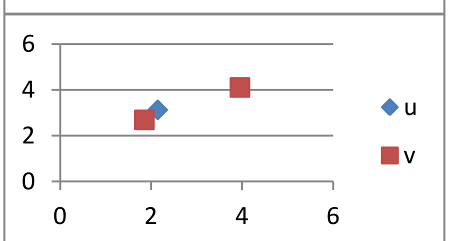


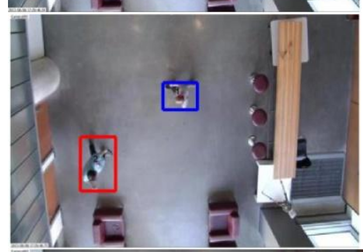
1940



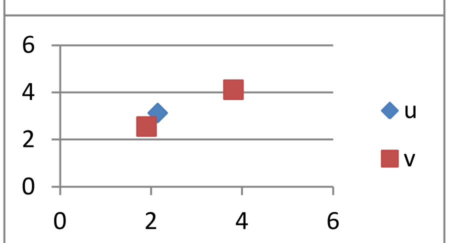


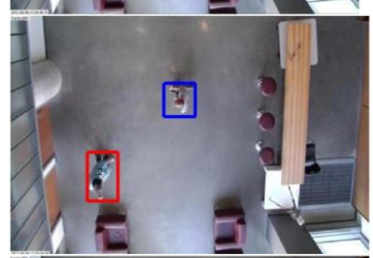
1945



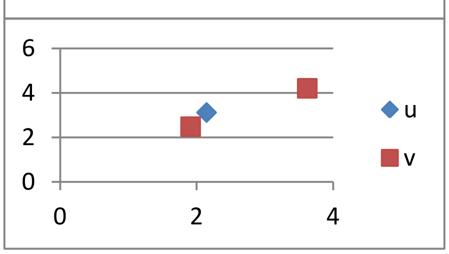


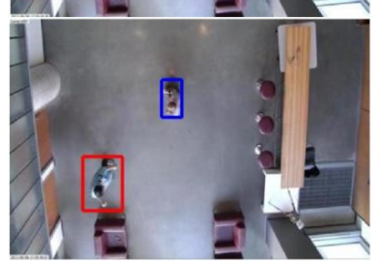
1950



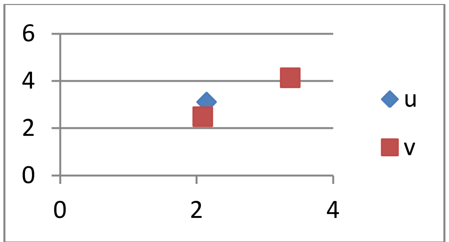


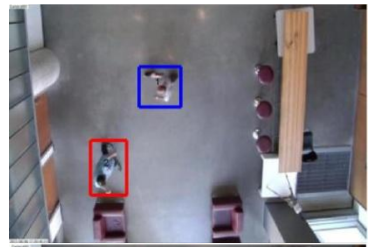
1955



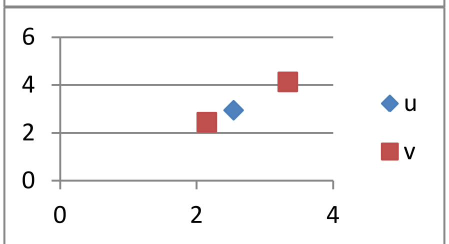


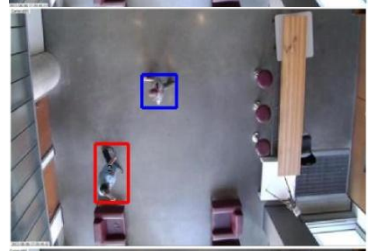
1960



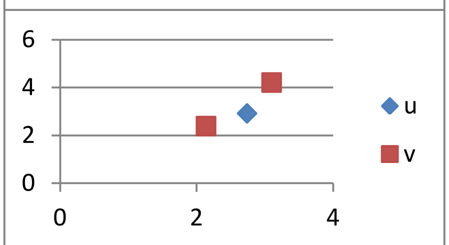


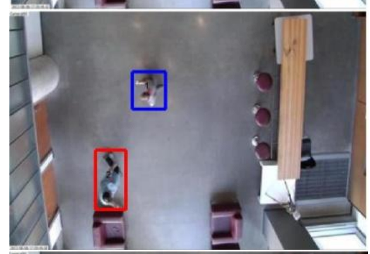
1965



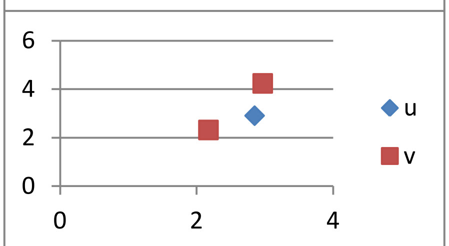


1970

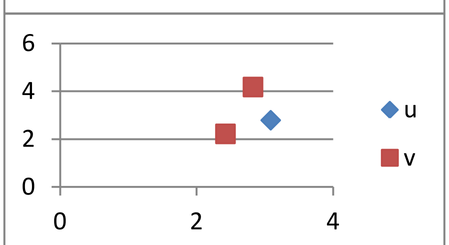


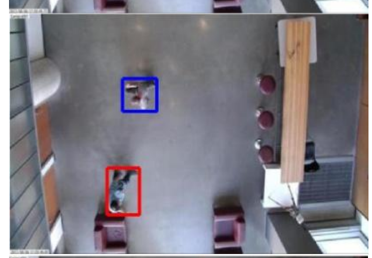


1975

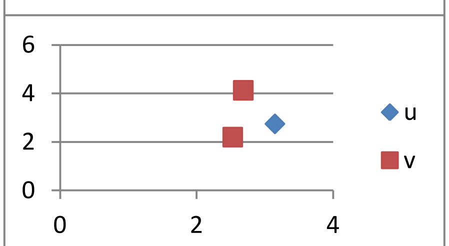


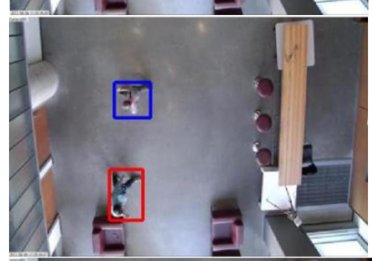
1980



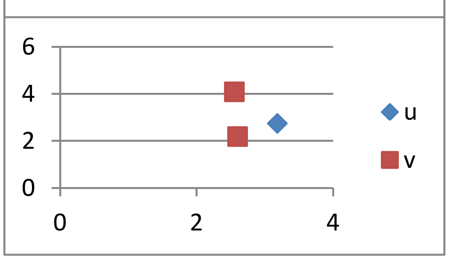


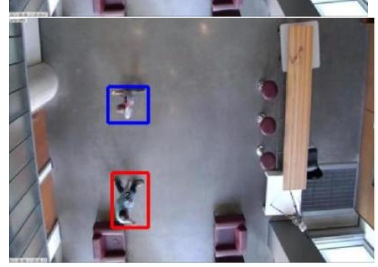
1985



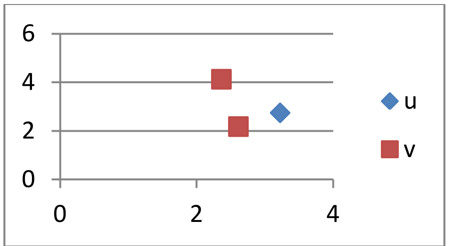


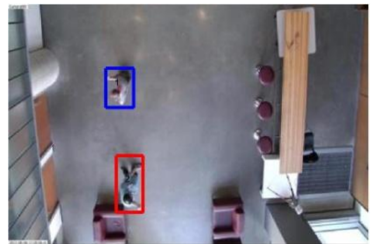
1990



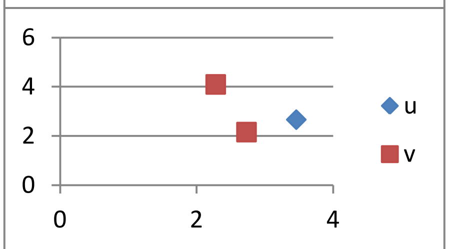


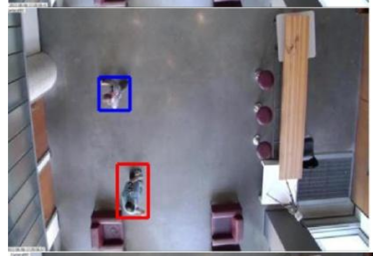
1995



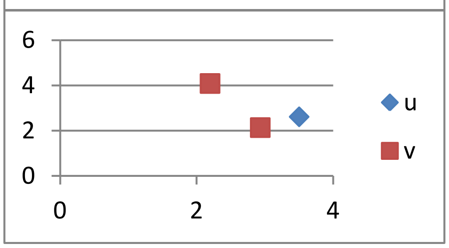


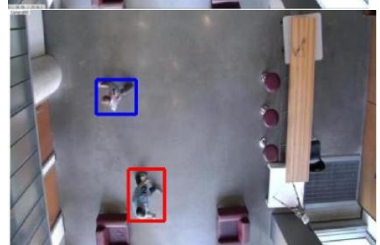
2000



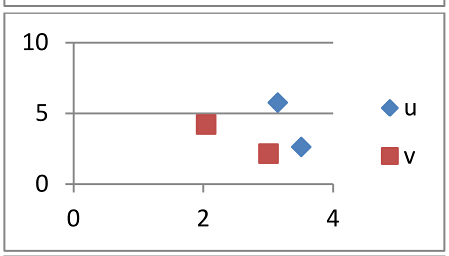


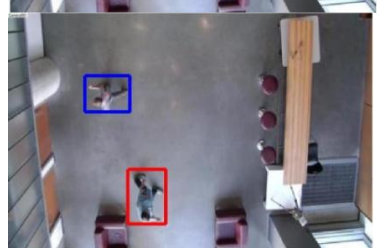
2005



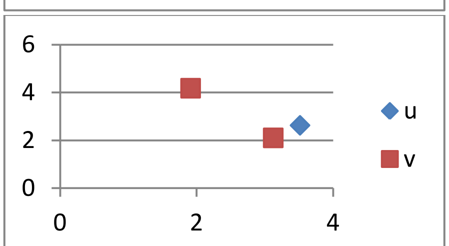


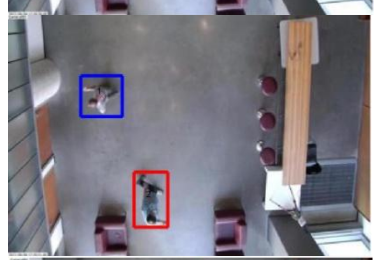
2010



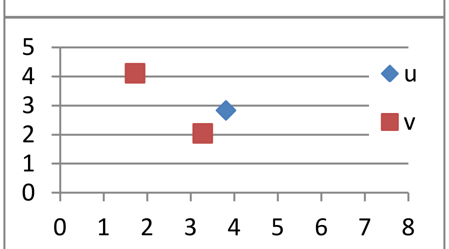


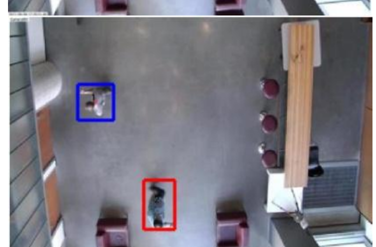
2015



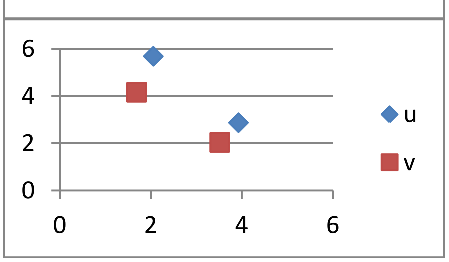


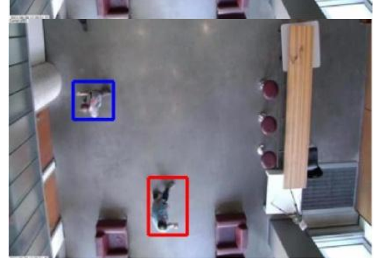
2020



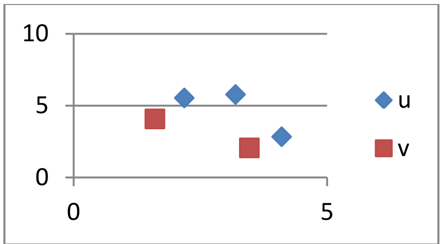


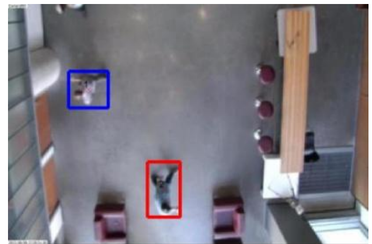
2025



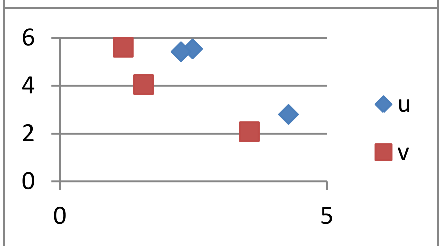


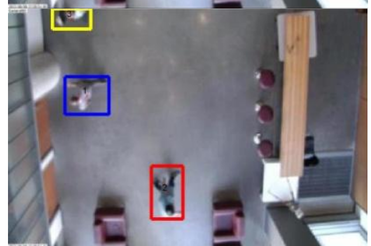
2030



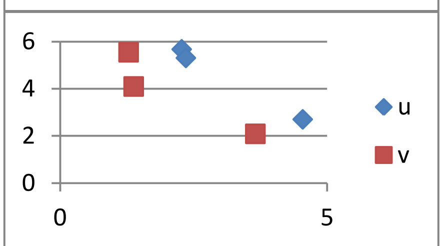


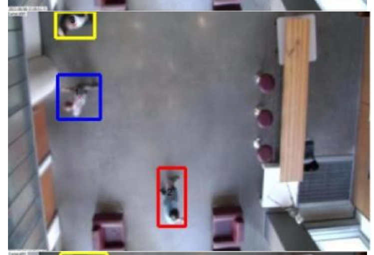
2035



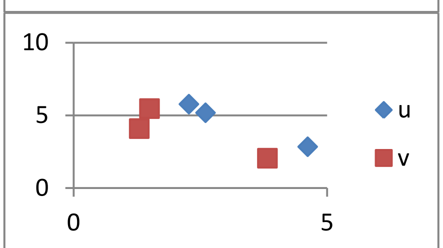


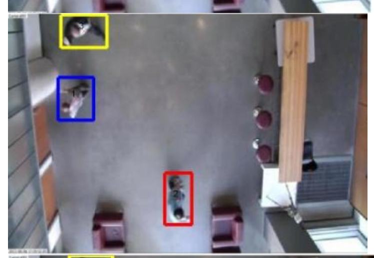
2040



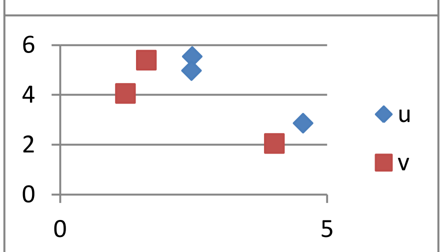


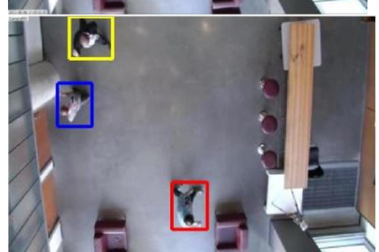
2045



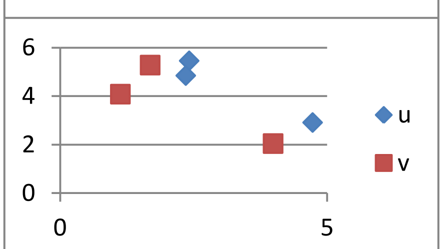


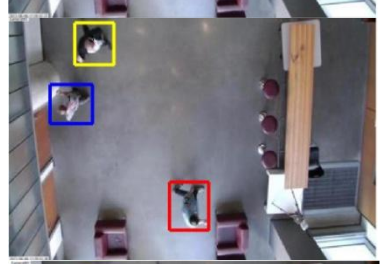
2050



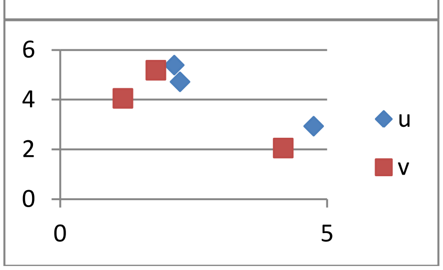


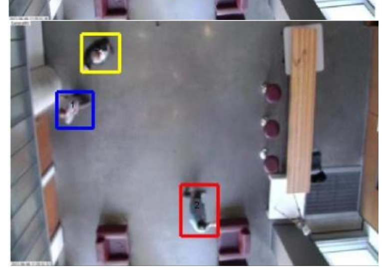
2055



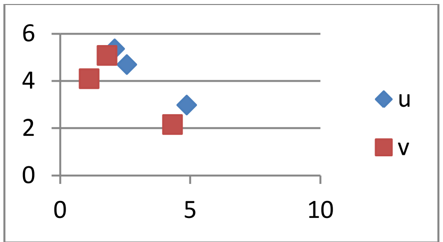


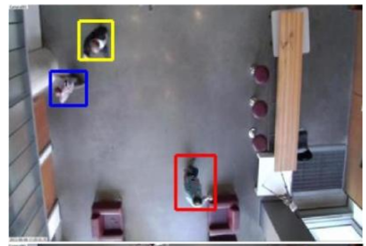
2060



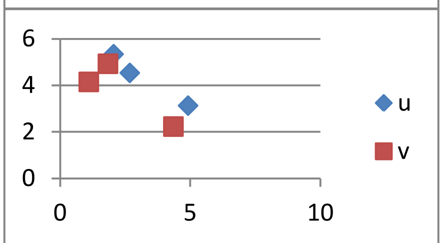


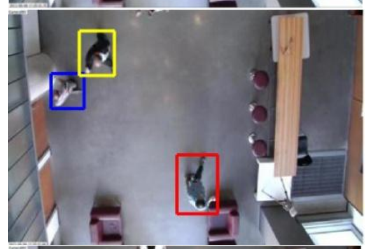
2065



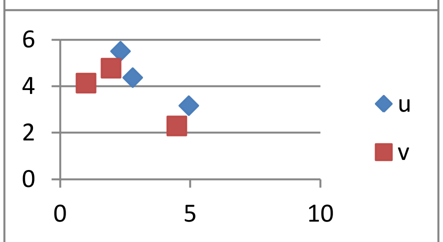


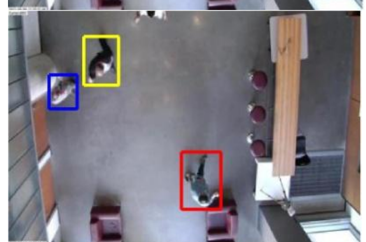
2070



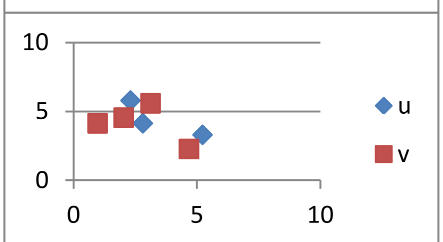


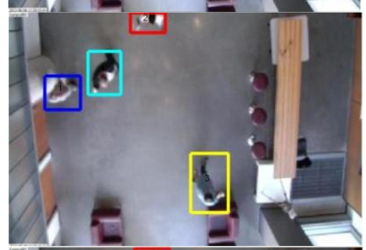
2075



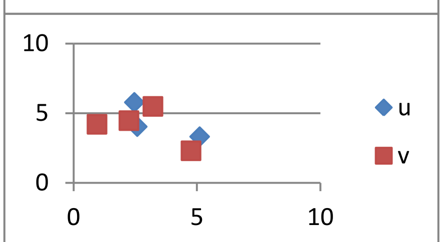


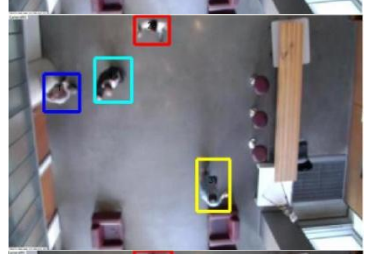
2080



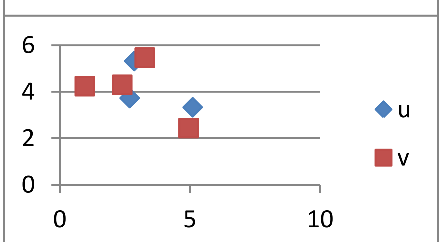


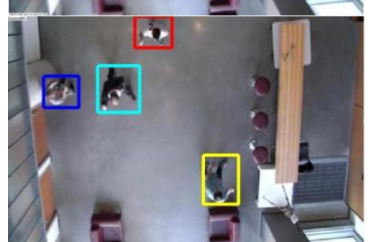
2085



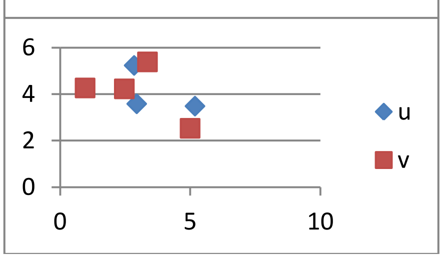


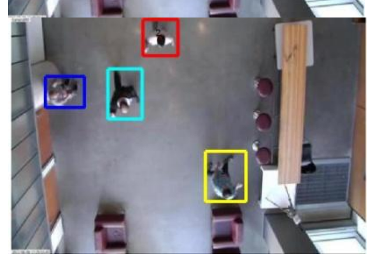
2090



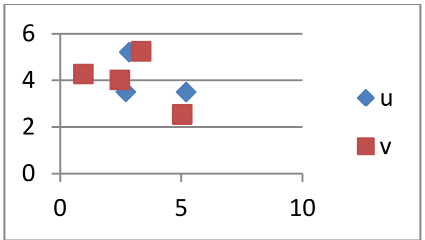


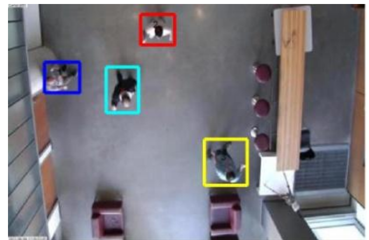
2095



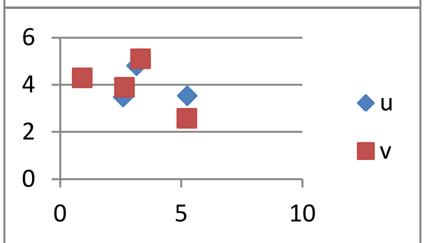


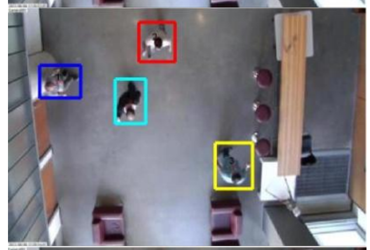
2100



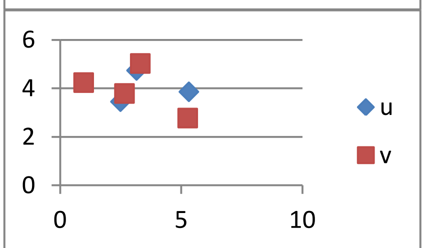


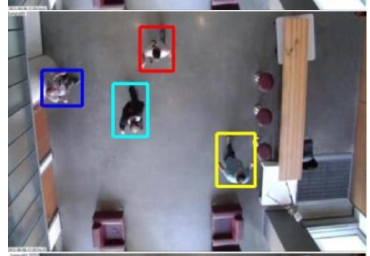
2105



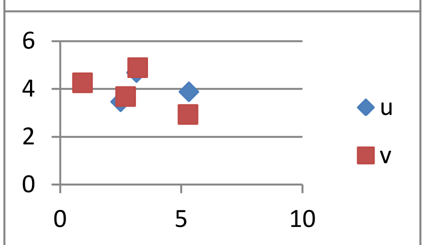


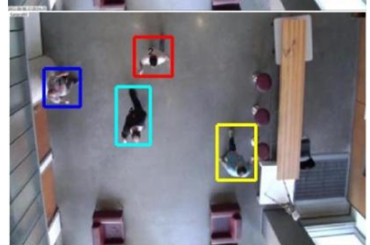
2110



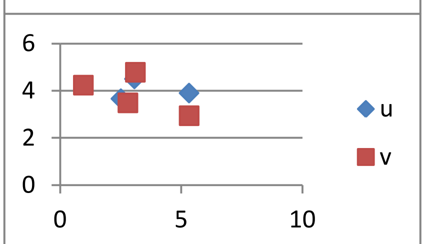


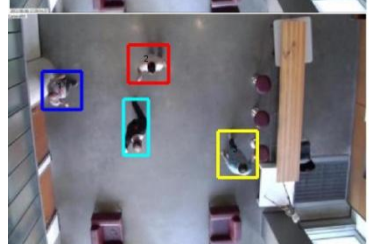
2115



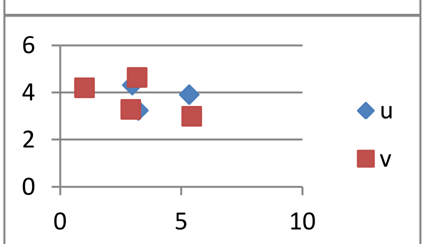


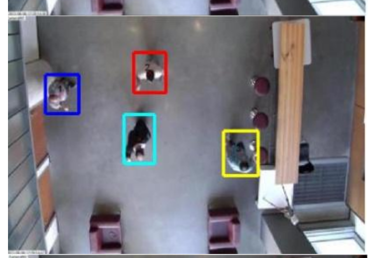
2120



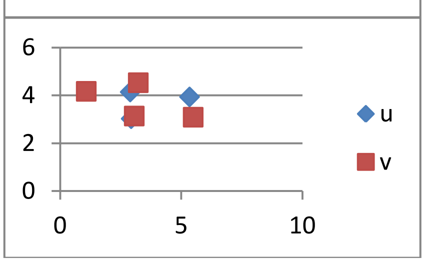


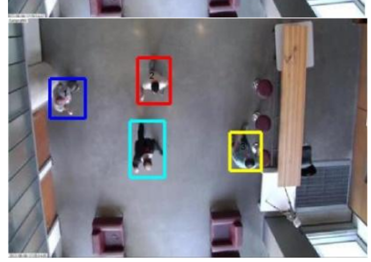
2125



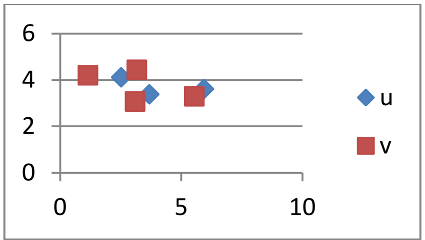


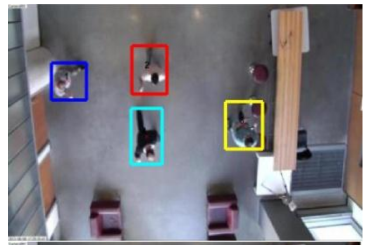
2130



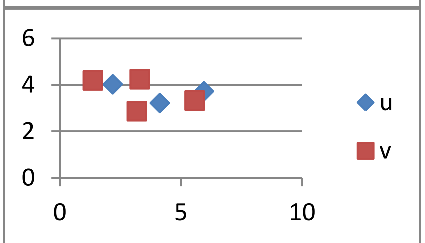


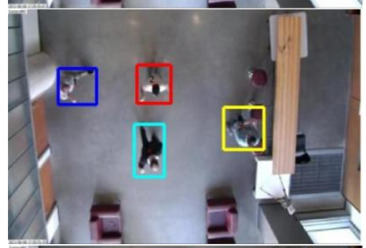
2135



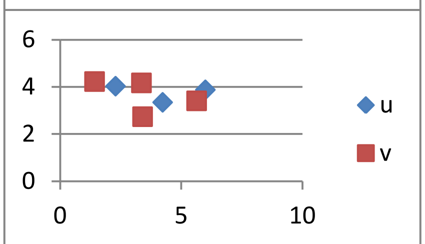


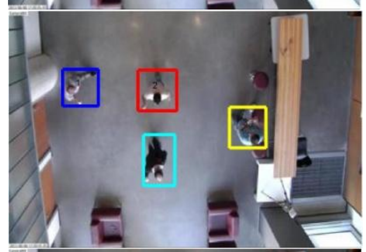
2140



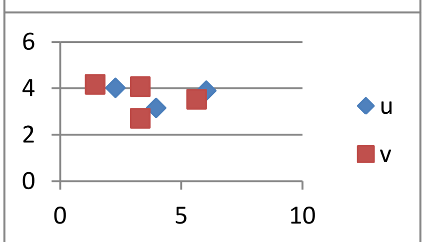


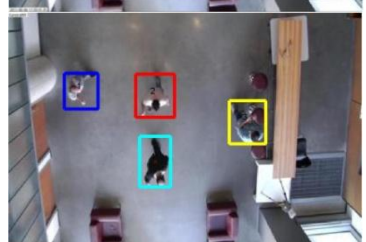
2145



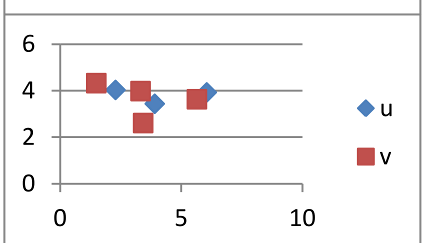


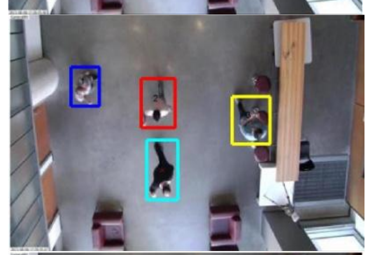
2150



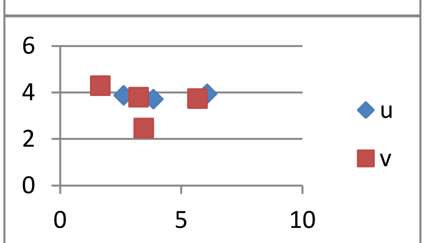


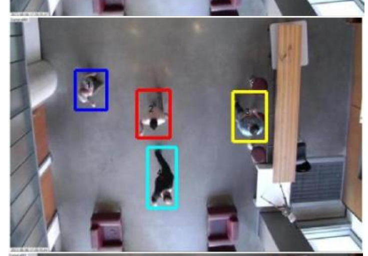
2155



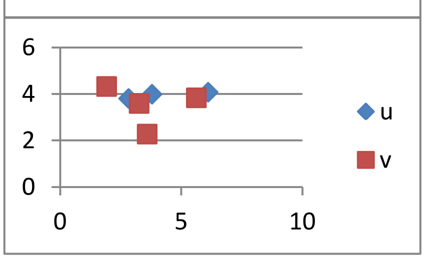


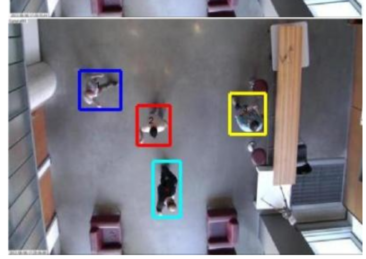
2160



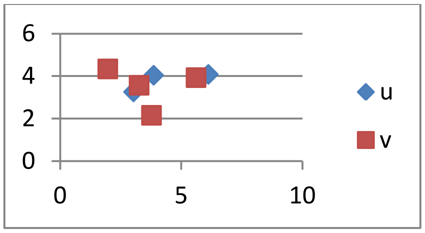


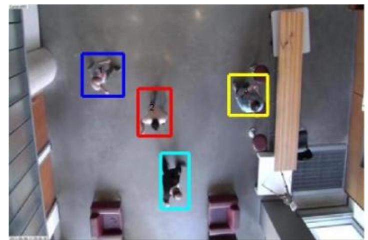
2165



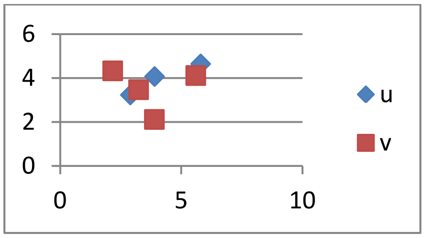


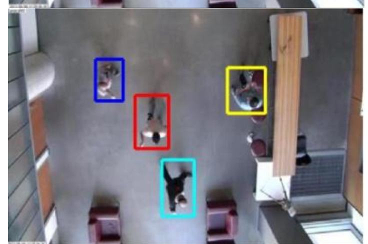
2170



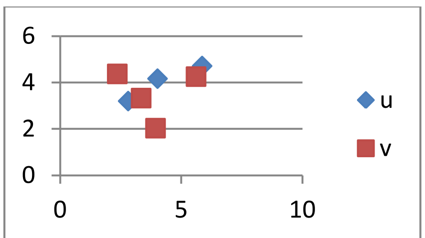


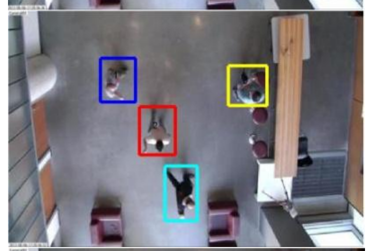
2175



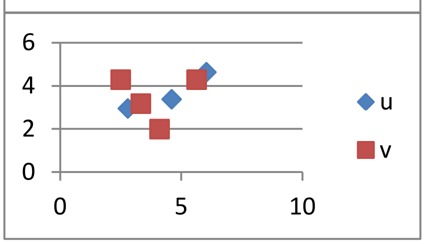


2180

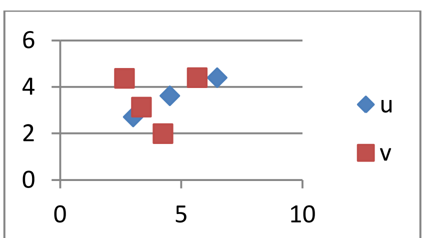


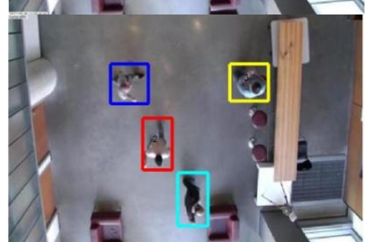


2185

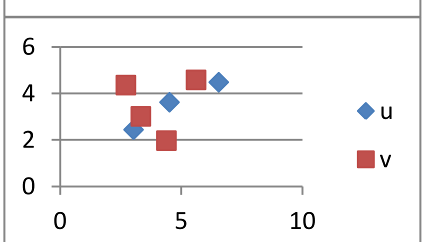


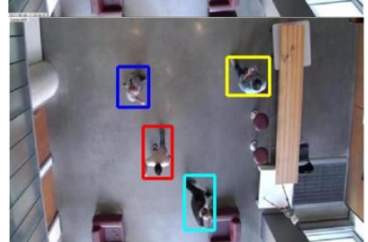
2190



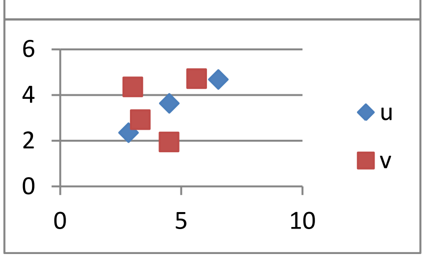


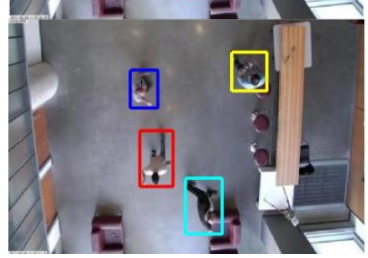
2195



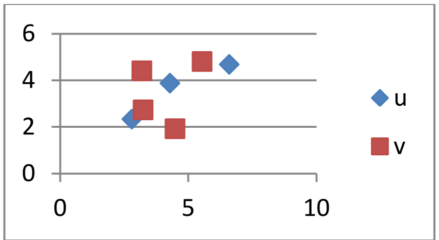


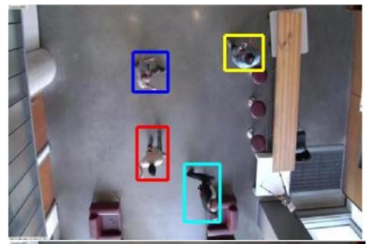
2200



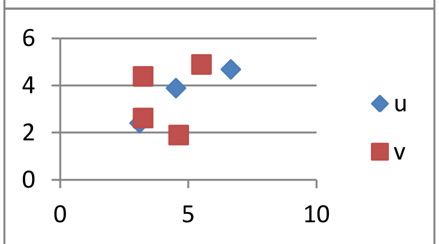


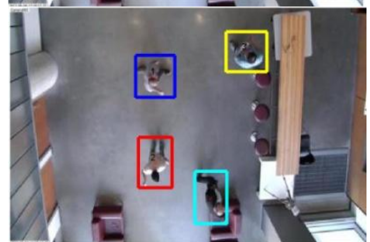
2205



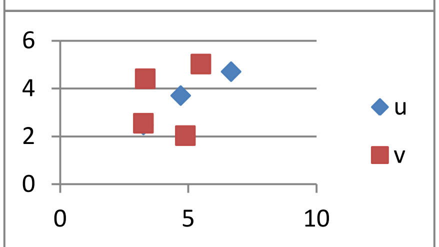


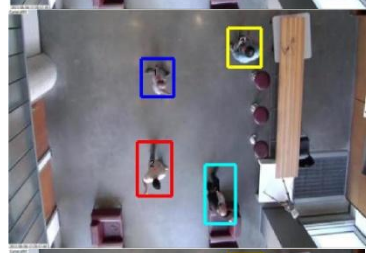
2210



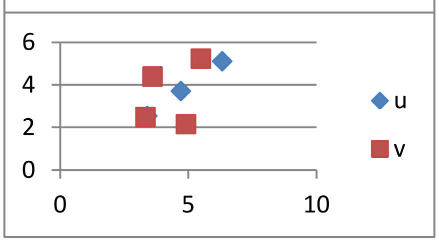


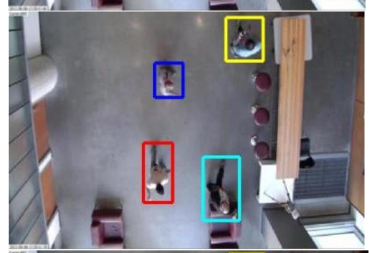
2215



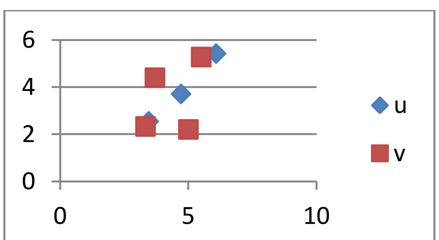


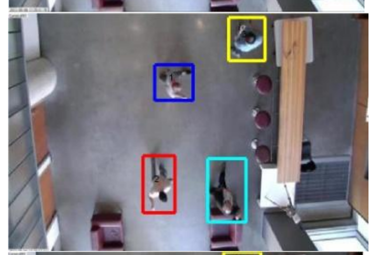
2220



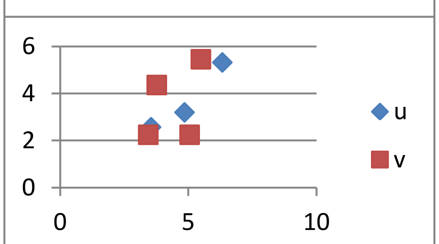


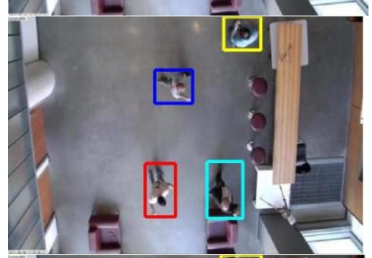
2225



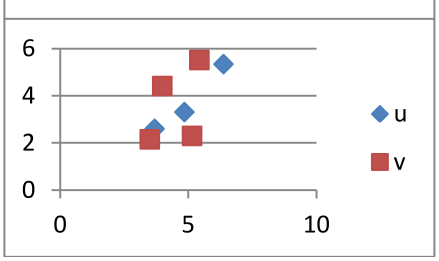


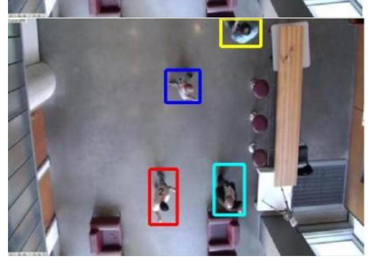
2230



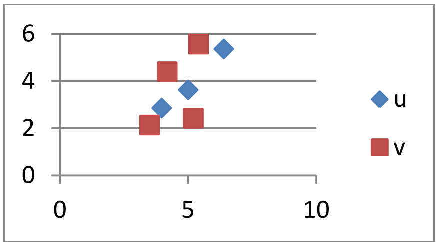


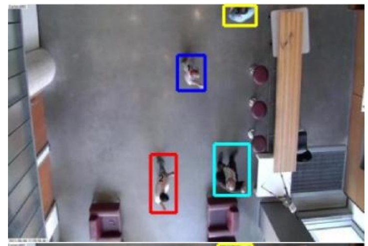
2235



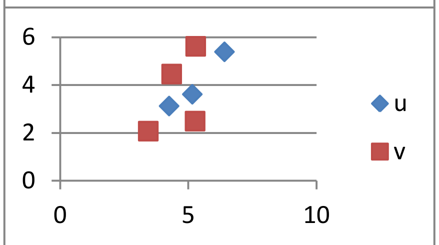


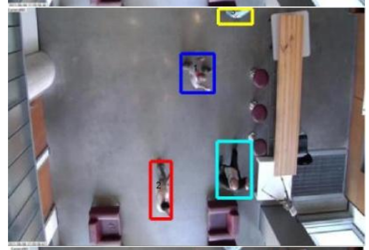
2240



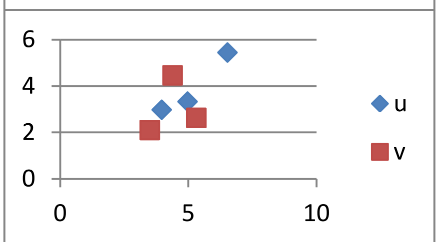


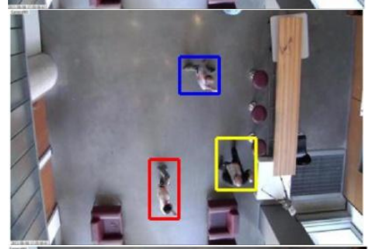
2245



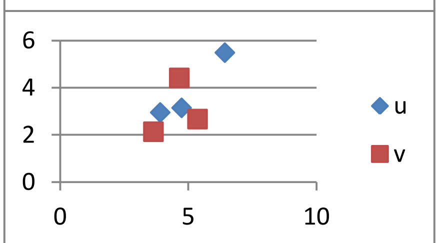


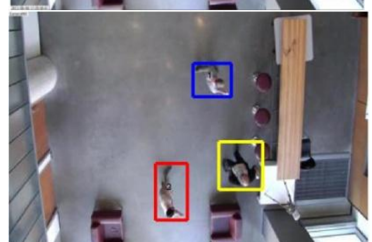
2250



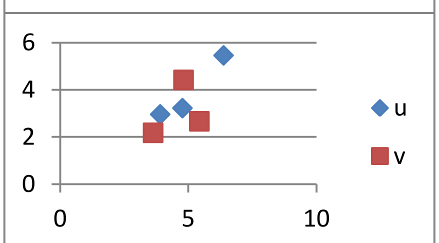


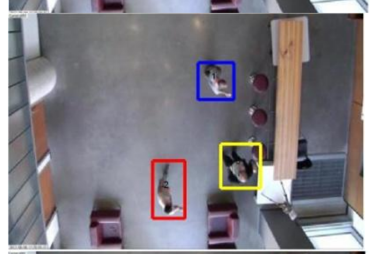
2255



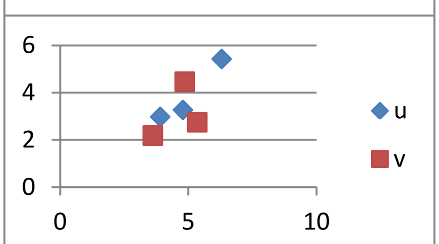


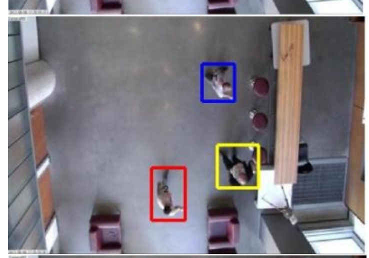
2260



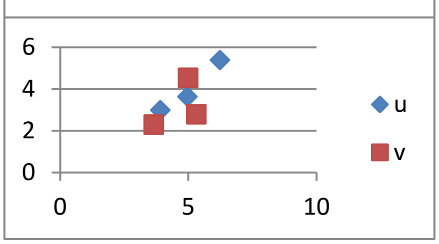


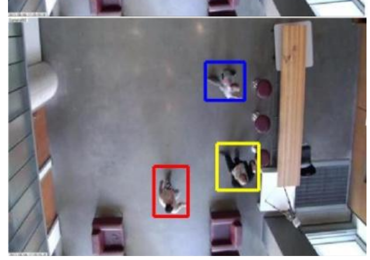
2265



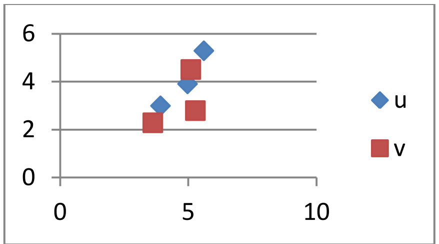


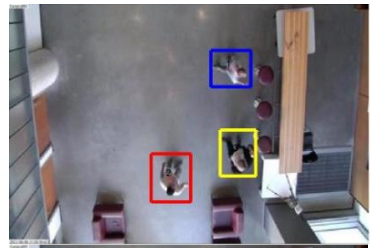
2270



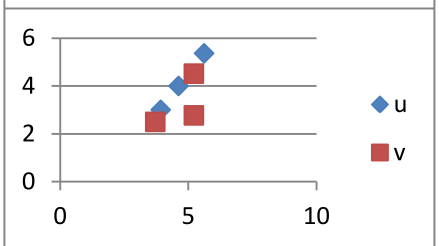


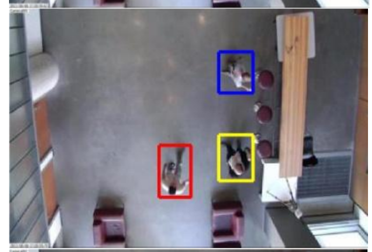
2275



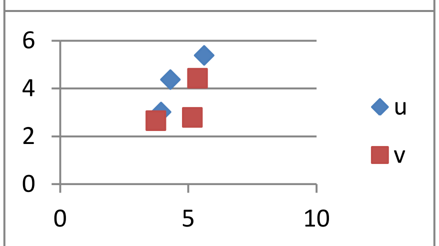


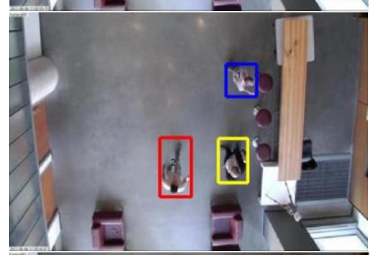
2280



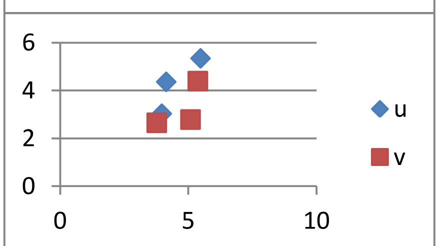


2285

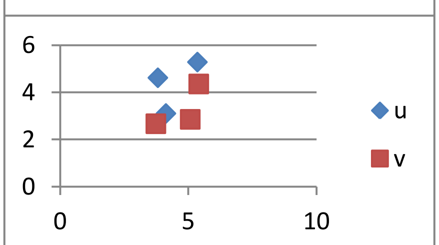


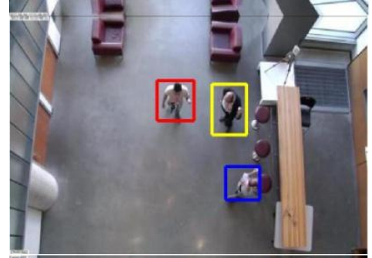


2290

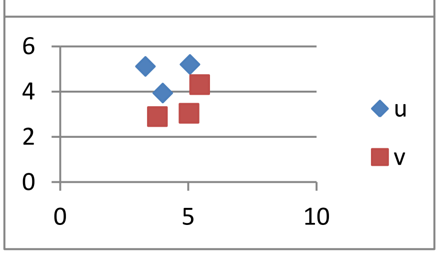


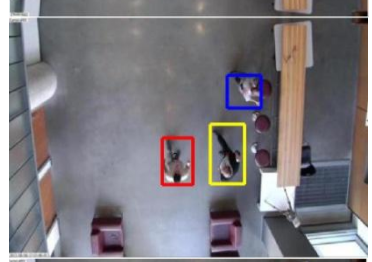
2295



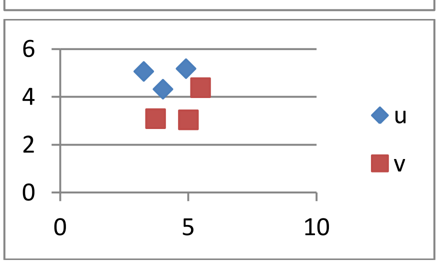


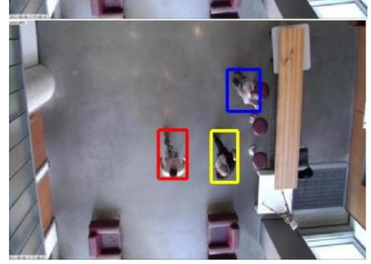
2300



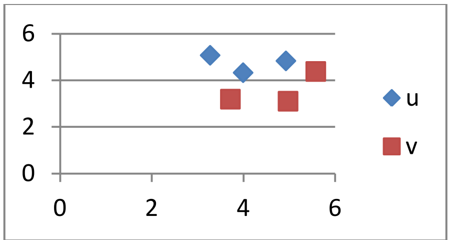


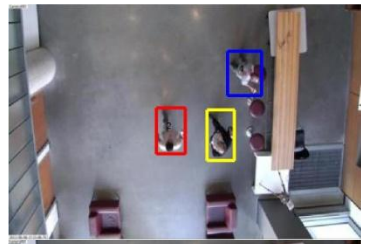
2305



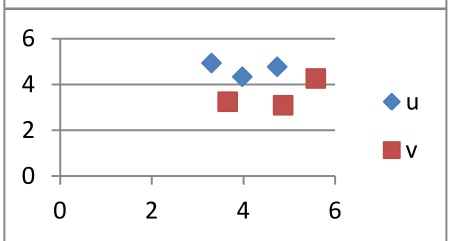


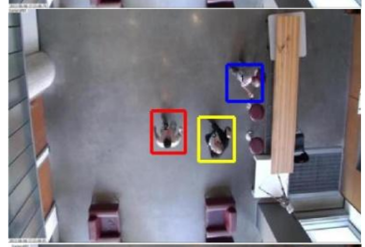
2310



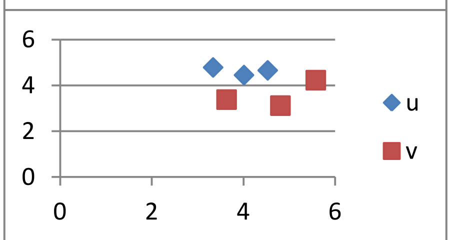


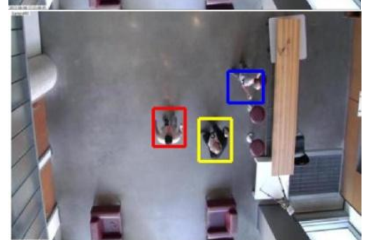
2315



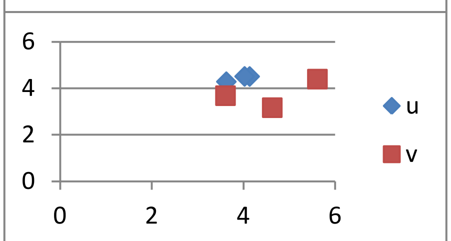


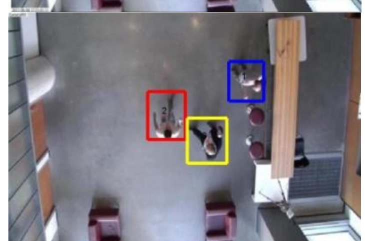
2320



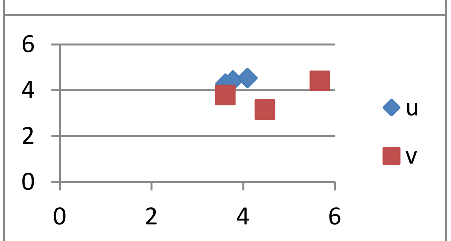


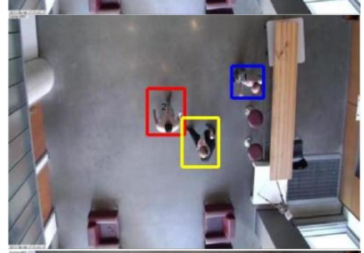
2325



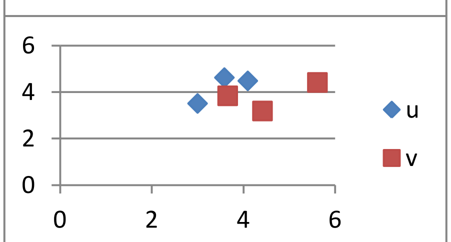


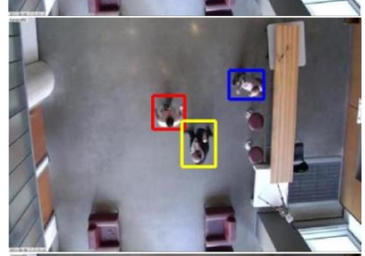
2330





2335





2340