THERMAL AND VISIBLE BAND IMAGE FUSION FOR ABANDONED OBJECT DETECTION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

AHMET YİĞİT

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

FEBRUARY 2010

Approval of the Graduate School of Informatics

____

Prof. Dr. Nazife Baykal
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

____

Assist. Prof. Dr. Tuğba Taşkaya Temizel
Head of Department

This is to certify that I have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

____

Assist. Prof. Dr. Alptekin Temizel
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Erkan Mumcuoglu (METU, II) _____________________

Assist. Prof. Dr. Alptekin Temizel (METU, II) _____________________

Assist. Prof. Dr. Erhan Eren (METU, II) _____________________

Assist. Prof. Dr. Altan Kocyigit (METU, II) _____________________

Assist. Prof. Dr. İlkay Ulusoy (METU, EE) _____________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Ahmet Yiğit

Signature : __________________

ABSTRACT

THERMAL AND VISIBLE BAND IMAGE FUSION FOR ABANDONED OBJECT DETECTION

YİĞİT, Ahmet
M.S., Department of Information Systems
Supervisor: Assist. Prof. Dr. Alptekin Temizel

February 2010, 79 pages

Packages that are left unattended in public spaces are a security concern and timely detection of these packages is important for prevention of potential threats. Operators should be always alert to detect abandoned items in crowded environments. However, it is very difficult for operators to stay concentrated for extended periods. Therefore, it is important to aid operators with automatic detection of abandoned items. Most of the methods in the literature define abandoned items as items newly added to the scene and stayed stationary for a predefined time. Hence other stationary objects, such as people sitting on a bench are also detected as suspicious objects resulting in a high number of false alarms. These false alarms could be prevented by discriminating suspicious items as living/nonliving objects. In this thesis, visible band and thermal band cameras are used together to analyze the interactions between humans and other objects. Thermal images help classification of objects using their heat signatures. This way, people and the objects they carry or left behind can be detected separately. Especially, it is aimed to detect abandoned items and discriminate living or nonliving objects

Keywords: Living/Nonliving Discrimination, Thermal Camera, Visible Band Camera, Thermal Image, Abandoned Object

ÖZ

TERK EDİLEN OBJELERİN BULUNMASI İÇİN TERMAL VE GÖRÜNÜR BANT VİDEO VERİ TÜMLEŞTİRMESİ

YİĞİT, Ahmet
Yüksek Lisans, Bilişim Sistemleri
Tez Yöneticisi: Yrd. Doç. Dr. Alptekin Temizel

Şubat 2010, 79 sayfa

Ortak yaşam alanlarında bırakılan paketler ciddi bir güvenlik problemi olup, zamanında tespit edilmeleri potansiyel tehlikelerden korunmak açısından oldukça önemlidir. Bu alanların güvenliğinden sorumlu olan kamera operatörlerinin sürekli dikkatli olmaları gerekmektedir. Ancak, bu tür ortak alanlari uzun süre dikkatli bir şekilde izlemek operatörler icin oldukça zordur. Bu yüzden, terk edilen paketlerin otomatik bir şekilde tespit edilmesi önemlidir. Literatürde terk edilen nesneleri bulmak icin kullanılan bir çok method önplana yeni giren ve belirli bir süre sabit duran nesneleri bulma amaçlı geliştirilmiştir. Ama, diger sabit duran nesneler, örneğin bir insanın belli bir süre bir bankta oturması, yanlış bir şekilde şüpheli nesne olarak algılanmasına sebebiyet verir. Şüpheli nesnelerin canlı/cansız olarak sınıflandırılması ile, bu tür yanlış alarmlardan korunulabilir. Bu tezde görünür bant kameralarının yanısıra aynı görüş alanına bakan bir termal kamera da kullanan bir sistem önerilmektedir. Bu sistemde görünür bant kamerası ve termal kameradan alınan görüntüler birlikte analiz edilerek canlılar ve cansız nesneler arasındaki ilişkilerin incelenmesi amaçlanmaktadır. Termal kameralardan elde edilecek sıcaklık bilgisi nesnelerin sınıflandırılmasına yardımcı olacaktır. Özellikle, terkedilen nesneler için canlı/cansız ayrımının yapılması amaçlanmaktadır

Anahtar Kelimeler: Canlı/Cansız ayrımı, Isı kameraları, Görünür bant kameraları, Sıcaklık bilgisi, Terk edilen Nesne

ACKNOWLEDGMENTS

I am deeply grateful to my supervisor Assist. Prof.Dr. Alptekin Temizel, who has helped me throughout my research, and encouraged me in my academic life. He always had time to discuss things and showed me the way when I felt lost in my research. It was a great opportunity to work with him.

I would like to thank my friends, Fatih Ömrüuzun, Çigdem Beyan and Ersin Karaman for their help in capturing the test videos.

Finally, I would also like to address my thanks to HAVELSAN for its encouragement during my MS study.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

LIST OF ABBREVIATIONS

1D	: One Dimensional
2D	: Two Dimensional
3D	: Three Dimensional
ACL	: Average Contour Length
AIR	: Automatic Image Registration
BG	: Background
CM	: Combination Model
CSM	: Contour Saliency Map
EO	: Electro Optic
FG	: Foreground
FOV	: Field of View
GA	: Genetic Algorithm
GFM	: General Fusion Model
HGA	: Hierarchical Genetic Algorithm
HOG	: Histogram Oriented Gradient
IR	: Infrared
LIO	: Local Intensity Operation
MAT	: Mean Value Thresholding
OpenCV	: Open Computer Vision Library
OS	: Operating System
ROI	: Region of Interest
SKDA	: Sequential Kernel Density Approximation
SVM	: Support Vector Machine
tCSM	: Thinned CS

CHAPTER 1

INTRODUCTION

In most crowded environments such as shopping malls and airports, surveillance system operators often watch a high number of cameras simultaneously to control the security of environment. Unfortunately, these systems are left unattended at certain times which result in security lapses during which criminals might leave suspicious packets in this environment which may result in catastrophic events. Therefore, detecting suspicious packets on time is crucial to protect the security of the public areas. In recent years, some studies have been made to detect abandon items automatically with computer assisted systems. In such systems, attaining low false alarm rates while not missing the real alarms is important. The high number of false alarms will result in operators ignoring these alarms. Therefore, the system should generate low false alarm rates.

Most of times, it is not sufficient to detect only abandon object in environment. People standing steadily in environment may also be detected as an abandon object. Also, terrorists generally wait for packages in certain time not to attract attention after leaving packages. So, living/nonliving discrimination can be made to lower the number of false alarms and detect not only abandon objects but also to spot the person who left the package behind.

To discriminate objects as living or nonliving objects, thermal imaging technology can be used. Thermal cameras give us the opportunity to discriminate objects whether they are living or not by using objects' temperature. Thermal camera uses Infrared (IR) energy as an imaging method which detects radiation in infrared range of electromagnetic spectrum and generates images according to radiation emitted by objects. The radiation emitted increases with temperature of the object so living objects like human and other warm-blooded animals appear much brighter against a cooler background. Thermal cameras are often confused with near infrared cameras (NIR) that work by imaging the non-visible infrared spectrum at 0.7-1 µm range. These cameras measure the infrared light reflected by objects similar to visible band cameras which measure the reflected light in the visible spectrum. These cameras require illumination in these respective spectrums to be able to capture images. Thermal cameras, on the contrary, measure the thermal radiation emitted by the objects typically at 8-14 µm range and hence doesn't require any kind of illumination. Thermal cameras are often called far-infrared (FIR) cameras.

There are two types of sensors used in thermal cameras:

Cooled Infrared Detector: This type of sensor operates at cooled environment. Cooled camera systems are more expensive, but generally have a longer range than uncooled systems under many conditions [1].

Uncooled Infrared Detector: Uncooled cameras use detectors operating at room temperature. Uncooled sensors have inherently less sensitivity than cooled sensors with comparable optics and bigger pixels [1].

There are a number of methods in the literature which are aimed to detect abandoned objects. However, these methods do not provide us any information about whether they are living or not.

In this thesis, we propose a solution to detect abandon objects with their aliveness information. Visible band and thermal band cameras will be used together to analyze the interactions between humans and other objects. Block diagram of the system is given in Figure 1.

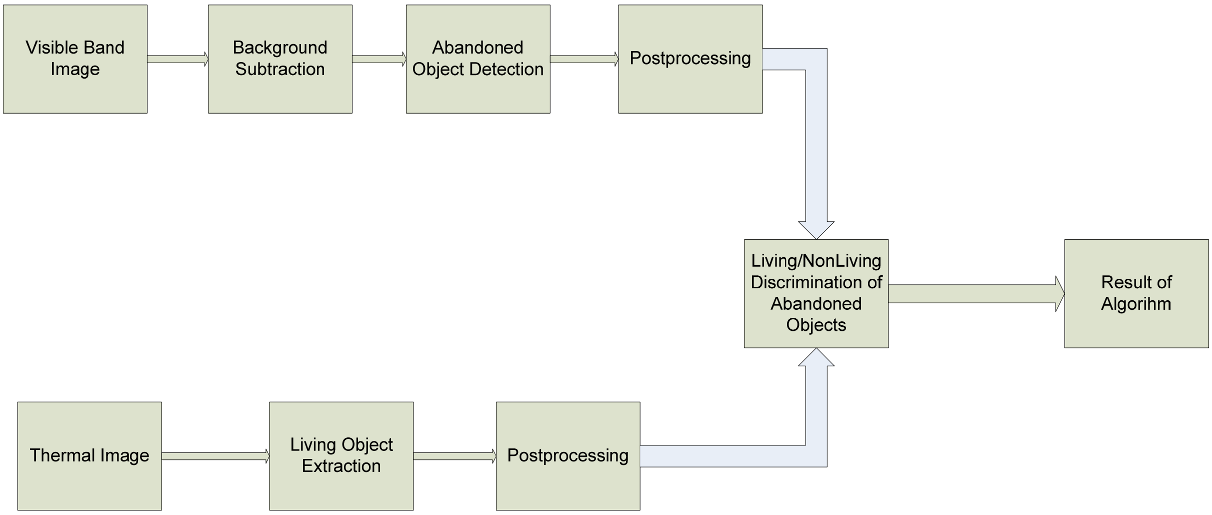


Figure 1: Block diagram of the proposed system

The system is composed of a visible band and a thermal camera which are located on non-moving surface and adjusted to capture almost the same Field of View (FOV). Our application is not working real-time so we start running our algorithm after registering thermal and visible band images. The algorithm consists of three main parts: Background Subtraction, Abandoned Object Detection, and Living/Nonliving Discrimination of Abandoned Object. All main parts run sequentially. First, background subtraction is used to find out the static background and discriminate the foreground. Then, Abandoned Object Detector detects abandoned objects by using dual background approach. Finally, living objects are extracted from thermal images and detected abandoned objects are discriminated as living or nonliving objects.

This thesis is organized in six chapters. In chapter 2, firstly the background subtraction methods in the literature are given. After that the background subtraction method that we use is explained. In chapter 3, the methods in the literature regarding abandoned object detection and thermal-visible band image fusion are summarized. In chapter 4, the algorithms to detect abandon objects with dual background approach, extract living objects from thermal images, explain post processing of results and discriminate abandon objects as living or nonliving with feature level data fusion. Following these, in chapter 5 the experimental results are given. Performance of the methods in the literature for background subtraction is compared to the methods which are proposed in this thesis and the results in various cases are discussed. In the last chapter conclusions and future work are summarized.

CHAPTER 2

BACKGROUND SUBTRACTION

2.1 Overview

Background subtraction method is a method which is used to detect stationary parts of video. Abandoned objects are even though not a part of the background initially; they eventually become a part of the background. Therefore, using effective background detection is one of the most important parts of detection of abandoned objects. There are several methods for background subtraction in the literature. In this chapter we give information about these techniques. After that the background subtraction method used in the thesis is explained in more detail.

2.2 Background Subtraction Methods in the Literature

Background subtraction is a widely used approach for detecting stationary parts of video. Methods in literature use image sequence of video for estimating background and detecting moving objects. However, all background subtraction methods face common problems and challenges while detecting background. Illumination change is one of the most important problems. To get better result for background subtraction, illumination should be constant in video. Also, moving objects, weather conditions such as rain and snow, shadows are other important challenges for background subtraction algorithms. A good background subtraction algorithm must be robust against such changes in the environment.

In [2], the author evaluates different background subtraction methods by means of their speed, memory requirements and accuracy. These methods are:

- Running Gaussian average
- Temporal median filter
- Mixture of Gaussians

- Kernel density estimation (KDE)
- Sequential Kernel Density approximation
- Cooccurence of image variations
- Eigenbackgrounds

Running Gaussian average: In this method, last *n* pixels' value are used to fit onto the Gaussian distribution over histogram. The method is proposed in [3] and at each *t* frame, a running average is calculated as:

$$\begin{aligned} \mu_t &= \alpha I_t + (1-\alpha)\mu_{t-1} \\ \sigma_t^2 &= \alpha(I_t - \mu_t)^2 + (1-\alpha)\sigma_{t-1}^2 \end{aligned} \qquad (2.1)$$

where

μ_{t-1} : the previous average value

α : update parameter used to change the stability of background model.

μ_t and σ_t are the two parameters of Gaussian probability density function (PDF) and mean and standard deviation for current frame *t*. After the calculation of Gaussian PDF's parameters, inequality (2.2) should be satisfied to set related pixel as a background.

$$|I_t - \mu_t| > k\sigma_t \qquad (2.2)$$

where

k : a constant used to change sensitivity

I_t : the current value of the pixel

This method can be chosen for fast runtime and low memory usage.

Temporal median filter: In this method, the algorithm is based on median of last *N* frames. Whether the pixel value of corresponding image is background or not is decided according to this median value. In his paper [2], Piccardi describes this method as "*The main disadvantage of a median-based approach is that its computation requires a buffer with the recent pixel values. Moreover, the median filter does not accommodate for a rigorous statistical description and does not provide a deviation measure for adapting the subtraction threshold.*"

Mixture of Gaussians: This method copes with multi modal background distribution. This model is one of the most powerful methods for background subtraction in case that background changes very fast. In proposed method [4], the probability of a certain pixel value, *x*, at time *t* is described by means of a mixture of *K* Gaussian distributions:

$$P(x_t) = \sum_{i=1}^{K} \omega_{i,t} * \alpha I_t + \eta\left(x_t, \mu_{i,t}, \sum\nolimits_{i,t}\right)$$

$$\eta(x_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{\frac{1}{2}(x_t - \mu_t)^T \sum^{-1}(x_t - \mu_t)} \tag{2.3}$$

where

$\omega_{i,t}$: an estimate of weight of the i^{th} Gaussian at time t

$\sum_{i,t}$: the covariance matrix of the i^{th} Gaussian at time t

η : Gaussian probability density function.

In [2], it is given as "*first, all the distributions are ranked based on the ratio between their peak amplitude, ω_i, and standard deviation,σ_i. The assumption is that the higher and more compact the distribution, the more is likely to belong to the background. Then, the first B distributions in ranking order satisfying:*

$$\sum_{i=1}^{B} \omega_i > T \tag{2.4}$$

with T an assigned threshold, are accepted as background."

This method models both background and foreground without holding large number of video frame. This model should be chosen when multimodal is need for background subtraction. Because its accuracy is really higher than the other methods in that case.

Kernel Density Estimation: This method models background with PDF given by the histogram of most recent pixels' value and Gaussian kernel is used to smooth each frame of video. Elgammal *et al.* in [5] proposed non-parametric method based on Kernel Density Estimation (KDE) which provides smoothed and continuous histogram for background subtraction. The probability of observing a certain pixel value, *x*, at time *t* is described as:

$$P(x_t) = \frac{1}{n} \sum_{i=1}^{n} \eta\left(x_t - x_i, \sum\nolimits_t\right) \tag{2.5}$$

where

η : Gaussian probability density function

$\sum_t$: Covariance matrix at time *t*.

After the calculation of *P(x_t)*, *x* can be classified as foreground if following inequality satisfied.

$$P(x_t) < T \tag{2.6}$$

where T is threshold value for selecting background.
This method requires high memory and works slow. However, the accuracy of this method is high [5].

Sequential Kernel Density Approximation: This method is based on mean-shift vector but mean-shift vector is an iterative method which has a very high computational cost and so much memory usage for background subtraction. In [2], author proposed a solution to reduce this computational cost and memory usage which is called Sequential Kernel Density Approximation (SKDA). In [6], Han *et al.* proposed a method which detects the initial set of Gaussian modes of the background PDF in the offline model initialization process. Later, modes are propagated by adapting them with new samples. Equation 2.7 shows us how this process works.

$$PDF(x) = \alpha(new\_mode) + (l - \alpha)\left(\sum existing\_modes\right) \tag{2.7}$$

where α is update parameter for SKDA
In [6], SKDA is faster than KDE and needs lower memory usage.

Cooccurrence of image variations: In [7] Seki *et al.* propose a method that neighboring blocks of pixels belonging to background change in a similar way over time. This is true if you are working on blocks of same background object in stable environment. However, background is not always stationary especially in outdoor scenes. The performance of algorithm for stationary backgrounds are not good because of some challenges such as illumination changes and motions in background objects such as swinging leaves, rain or snow.

The proposed method in [7] uses block based approach that improves background subtraction in dynamically narrowing the ranges of background image variations for each video frame. There are 2 phases is this method. In first phase called Learning phase, certain number of samples are collected and image variation is computed by means of difference between samples and mean of these samples for each block. Then, image covariance matrix with respect to mean of samples and eigenvector transformation are computed. Normally, dimension of covariance matrix N^2 but eigenvector transformation reduces the dimension of covariance matrix to K where $K < N^2$. This reduction speeds up the system. In second phase, all blocks are classified as background or foreground with using eigen image variation for each block.

Cooccurance of image variations method is a complex and effective method. Although this method is slower and consumes more memory than all the other methods explained, the accuracy of this method is really high.

Eigenbackgrounds: This method in [8] is a method based on eigenvalue decomposition applied on the whole image instead of blocks of image. Eigenvalue decomposition is applied to sequence of images to compute eigen backgrounds. This method consists of 2 main phases. In first phase called Learning phase, n frames are acquired and the mean image is calculated. After that, covariance matrix is computed with respect to the mean image and best eigenvectors are stored in eigenvectors matrix. In second phase called Classification phase, after new image is available, it is projected onto the eigenvectors sub-space and reconstructed as a projected image. The difference between new image and projected image is calculated. If this difference is greater than threshold value, then system decides that it is foreground.

According the experimental results in [2]**,** this method has more accuracy than the other methods explained above. Its memory usage depends on the number of recent images which are used to calculate average image. The speed of the method changes with the number of best eigenvectors stored in the matrix.

2.3 Improved Adaptive Gaussian Mixture Model for Background Subtraction

Detection of abandoned objects is the main aim of this thesis. To extract abandon item correctly, we need a powerful background subtraction algorithm. Therefore, we decided to use Improved Adaptive Gaussian Mixture Model [9] which has been reported to be producing reliable background information while being computationally not very complex.

This method is a pixel-based background subtraction. The pixel should belong to Background (BG) or some foreground object (FG). So we can make a Bayesian decision *R*.

$$R = \frac{p\left(BG \mid \vec{x}^{(t)}\right)}{p\left(FG \mid \vec{x}^{(t)}\right)} = \frac{p\left(\vec{x}^{(t)} \mid BG\right)p\left(BG\right)}{p\left(\vec{x}^{(t)} \mid FG\right)p\left(FG\right)} \tag{2.8}$$

where $x^{(t)}$ is the value of pixel at time t.

In method proposed in [9], no prior information about foreground objects is assumed such as when and how often foreground objects will be in the samples. Pixels have equal probability of being foreground and background. Therefore, we can set $p(BG) = p(FG)$ and assume uniform distribution for foreground object appearance $P(x^{(t)}|FG) = c_{FG}$. So, a pixel belongs to background if the following inequality is satisfied.

$$p\left(\vec{x}^{(t)} \mid BG\right) > c_{thr}\left(= Rc_{FG}\right) \tag{2.9}$$

where c_{thr} is a threshold value and $p(x^{(t)}|BG)$ is a background model.

We can conclude from inequality 2.9 that a pixel belongs to the background if it is greater than the threshold that discriminates background from foreground. Background model estimates background over training set *X*. After starting to estimate background, model should be updated with new samples and old samples should be deleted to adapt to the changes in background. Let training set at time t be $X_T = \{x^{(t)}, ..., x^{(t-T)}\}$ where T is the time period. Also, X_T should be updated with new samples and background model reestimates background according to the new training data. However, there can be a foreground object in previous background so background model should estimate $p(x^{(t)}| X_T, BG + FG)$. We can express this model as Mixture of Gaussians with M components as follows:

$$\hat{p}\left(\vec{x} \mid X_T, BG + FG\right) = \sum_{m=1}^{M} \hat{\pi}_m N\left(\vec{x}; \hat{\vec{\mu}}_m, \hat{\sigma}_m^2 I\right) \tag{2.10}$$

where $\mu_1, \mu_2, ..., \mu_M$ are estimates for mean and $\sigma_1, \sigma_2, ..., \sigma_M$ are estimates for variance that describe the Gaussian components. $\pi_1, \pi_2, ..., \pi_M$ are the weight values that are nonnegative and up to one. Therefore, in this model we have three variables that we need to update once new samples are available. So, given new data sample $x^{(t)}$ at time *t*, the recursive update equations:

$$\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha\left(o_m^{(t)} - \hat{\pi}_m\right) \tag{2.11}$$

$$\hat{\vec{\mu}}_m \leftarrow \hat{\vec{\mu}}_m + o_m^{(t)} \left(\alpha / \hat{\pi}_m\right) \vec{\delta}_m \tag{2.12}$$

$$\hat{\sigma}_m^2 \leftarrow \hat{\sigma}_m^2 + o_m^{(t)} \left(\alpha / \hat{\pi}_m\right) \left(\vec{\delta}_m^T \vec{\delta}_m - \hat{\sigma}_m^2\right) \tag{2.13}$$

where $\delta_m = x^{(t)} - \mu_m$, $o^{(t)}$ is a ownership, and α is exponentially decaying envelope and it is approximately $\alpha = 1/T$, T is a time period. $o^{(t)}$ for each component m is set to one if it is "close" component to largest π_m and the others are set to 0. New sample is "close" to component if the Mahalanobis distance from component is less than four standard deviation and square distance from m_{th} component is as follows:

$$D_m^2\left(\vec{x}^{(t)}\right) = \vec{\delta}_m^T \vec{\delta}_m / \hat{\sigma}_m^2 \tag{2.14}$$

Where $D_m(x^{(t)})$ is the mahalanobis distance from m_{th} component.

If there is no close component found, we need to restart system with $\pi_{M+1} = \alpha$, $\mu_{M+1} = x^{(t)}$, $\sigma_{M+1} = \sigma_0$ where σ_0 is appropriate initial variance. Also, if number of components is reached to maximum limit, we need to discard the smallest π_M.

This algorithm is an online clustering algorithm and the foreground objects can be detected as some additional clusters with small weights π_M. So we can select the first B largest clusters to approximate the background model as follows:

$$B = \arg\min_b \left(\sum_{m=1}^{b} \hat{\pi}_m > \left(1 - c_f\right) \right) \tag{2.15}$$

$$p\left(\vec{x} \mid X_T, BG\right) \sim \sum_{m=1}^{B} \hat{\pi}_m N\left(\vec{x}; \hat{\vec{\mu}}_m, \sigma_m^2 I\right) \tag{2.16}$$

where c_f is a measure of maximum portion of foreground objects' data. If the object remains stationary long enough, its weight becomes larger than c_f and it is detected as background.

CHAPTER 3

LITERATURE REVIEW

3.1 Overview

In this chapter, we review the literature on thermal and visible imagery fusion. There are many domains and research topics that image fusion can be applied. For example, in [10], image fusion is used for background subtraction to obtain better background subtraction result. In [14], image fusion is used to detect moving objects. However, to our knowledge, there isn't any method in the literature that is directly related with image fusion for abandoned object detection. Hence, we explain methods in the literature about image fusion that are helpful and inspired us to develop image fusion methodology in this thesis.

3.2 Background-subtraction using contour-based fusion of thermal and visible imagery

One of the most important features of the surveillance applications is persistence. However, it is generally hard to provide this feature by using single sensor. [10] proposes a solution for this problem by using both thermal and visible band cameras. Especially in background subtraction applications, visible band video is poor to detect background and extract foreground at bad illuminations or raining, snowing conditions. Thermal camera is used to overcome these problems. Thermal camera provides a video that is extracted from the relative energy emitted by objects. Therefore, thermal camera is not affected by bad illumination and weather conditions such as rain and snow.

In [10], authors propose enhanced background subtraction algorithm using thermal and visible band video. This method is a region-based processing algorithm that finds and fuses contours that is most salient for both thermal and visible images. Figure 2 shows flow chart of this algorithm.

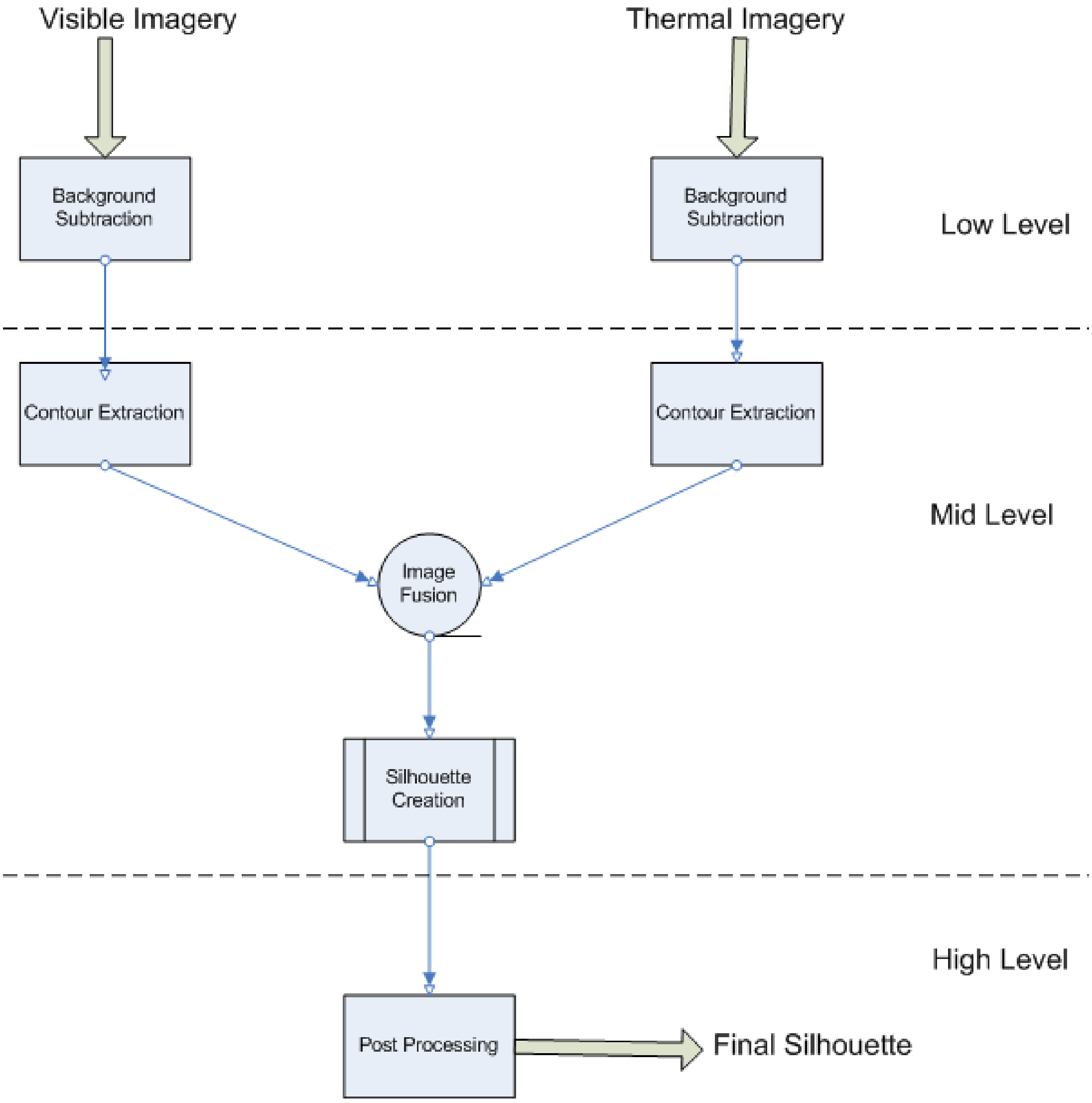


Figure 2: Flowchart of Background Subtraction using contour based fusion of thermal and visible imagery [10]

There are 3 main stages in this method. In the first stage, region of interest (ROI) is identified for both thermal and visible domain. ROI in thermal imagery is used to localize the background subtraction in visible domain. In the second stage, contours are extracted from both thermal and visible image. Then, fused contours are closed and completed by using flood-fill operations to create contours. In the last stage, some regions that do not satisfy minimum size threshold are discarded by post-processing operation.

Stage 1, Low-Level Processing: The initial step of this stage is to register thermal and visible images. After registration, we need to find ROI for both thermal and visible domain. Background subtraction algorithm runs on thermal domain since environmental changes, such as bad illumination, have no effect in thermal domain. Therefore, background subtraction algorithm will run on thermal domain to get ROIs and extract corresponding ROIs from visible domain.

To model background in thermal domain, the method uses one Gaussian at each pixel with using N samples. After modeling background, foreground, D^T, can be obtained by using Mahalanobis distance in thermal domain as follows:

$$D^{\mathrm{T}}(x,y)=\begin{cases}1 & \dfrac{\left(I(x,y)-\mu(x,y)^2\right)}{\sigma(x,y)^2}>Z^2\\ 0 & \text{otherwise}\end{cases} \tag{3.1}$$

where Z is threshold value that discriminate foreground from background, $I(x,y)$ is the pixel value of intensity image, and $\mu(x,y)$ and $\sigma(x,y)$ is the mean and standard variance value of samples respectively.

After detecting foreground objects, 5x5 dilation operation is applied on the background subtracted image, D^T, to extract ROIs.

The next step is to extract ROIs, D^V, in visible domain. In stable case, we can set ROI in visible domain that is the same ROI in thermal domain. However, there can be unwanted things such as shadows on visible band. To overcome these challenges, we need to define 2 background models for intensity and colored image for visible domain. After modeling 2 backgrounds D_{Int}, D_{Col} and extracting ROIs for both of them, 5x5 dilation operation is applied on D_{Int} and D_{Col}. Finally, we apply pixel-wise union operation on D_{Int} and D_{Col}, and obtain ROIs in visible domain D^V.

Stage 2, Mid-Level Processing: In previous stage, we determined ROIs for both thermal and visible domain. Now, we need to extract these ROIs. To do that, we first need to create Contour Saliency Map (CSM) [11] for each ROI. Pixel value of CSM is the confidence of that pixel belonging to the boundary of foreground objects. CSM is calculated as follows:

$$\mathrm{CSM}=\min\left(\frac{\left\|\langle I_x,I_y\rangle\right\|}{\mathrm{Max}_{\mathrm{I}}},\frac{\left\|\langle (I_x-BG_x),(I_y-BG_y)\rangle\right\|}{\mathrm{Max}_{\mathrm{I\text{-}BG}}}\right) \tag{3.2}$$

where I_x and I_y are input gradients, BG_x and BG_y are background gradients, Max_I is maximum value of the Input gradients and $Max_{I\text{-}BG}$ are maximum value of input-background gradient-differences in the ROI. CSM is calculated for both of thermal and visible domain. Gradient calculation is made by using 7x7 Gaussian derivative mask. CSM values are between 0 and 1. Larger CSM means stronger confidence and pixel is probably boundary of foreground object.

After finding CSM, we need to make thinning and thresholding operations on CSM to extract contours correctly. Thinned CSM (tCSM) is the multiplication of the CSM with thinning mask which is non-maximum suppression of the input gradients. For thresholding, the method [12] can be used for thermal and visible domain. This method first clusters tCSM pixels and discard the lowest clusters.

After extracting contours in thermal and visible domain, [10] uses simple pixel-wise union to fuse contours coming from thermal and visible domain.

$$\mathrm{tCSM_b} = \mathrm{tCSM_b^T} \cup \mathrm{tCSM_b^V} \tag{3.3}$$

where $tCSM_b$ is binary thinned CSM for fused result, $tCSM_b^T$ is binary tCSM for thermal domain, and $tCSM_b^V$ is binary tCSM for visible domain.

Although it is simple and effective to fuse contours, corresponding contours may not be matched for thermal and visible domain and this will cause the contour fragments. To get rid of these contour fragments, $\mathrm{tCSM_b}$ needs to keep these fragments that correspond to maximum gradient point in both domains. We can do that by generating a combined gradient map from foreground gradients of each domain. Gradient information in $\mathrm{tCSM_b}$ is selected from either thermal or visible domain depending on it being present in $\mathrm{tCSM_b}^\mathrm{T}$ and $\mathrm{tCSM_b}^\mathrm{V}$. If gradient information exists for both domains, we can choose any of them. After selecting contour fragments, second thinning operation should be applied to get better alignment for contours.

After fusion of contours, we need to complete and close contours since contours could still be broken. However, we need to select valid contour and eliminate others before complete and close contour operation. To eliminate contour fragment that can harm completion and close operation, this method uses basin-merging algorithm that uses Student’s t-test with a confidence threshold of 99% on watershed (WT) partitions. Basin-merging algorithm checks whether pixels of two adjacent basins in ROI is similar or different. If adjacent basins are similar, these two basins are merged. After this merging operation on WT, author says

“Based on the merged WT, the binary $tCSM_b$ is partitioned into distinct segments that divide pairs of adjacent basins. A $tCSM_b$ segment is considered valid only if its length is at least 50% of the length of the WT border separating the two basins”. Once valid contours are found, contour gaps should be completed using watershed lines on pathways of $tCSM_b$. Therefore, each loose endpoint of contour segments should lie on the watershed lines until other endpoint is found. To find best path to connect loose endpoints where gaps are, search algorithm [13] is used. Algorithm [13] first calculates Euclidean distance from current pixel to the other thresholded pixel and selects minimum one. Valid search path should be on the watershed. After finding the optimum path, gaps are completed. After complete operation, contours should be in closed loop. To close the contours, we need to choose the nearest pair of points that lie on closed loop connected by this contour. The path that connects these two points will close the contour.

Stage 3, High-Level Processing: In this phase, we weight each resulting silhouette with contrast value that shows how different that region is from background. After that, 3-tap temporal median filter is applied to 3 adjacent frames to improve results.

Fusion method used in [10] is very similar to our fusion method. In our fusion method, we use feature based union operator. Aliveness information of object is coming from thermal domain, and abandoned information of object is coming from the visible domain. Then, we fuse these features by using our feature-based union operator.

3.3 Fusion of color and infrared video for moving human detection

This paper [14] deals with human body extraction from thermal and visible band video with Automatic Image Registration (AIR). In most of this kind of algorithms, the initial step is the extraction of human body by using electro-optic (EO) sensors. However, there are so many challenging issues that we can face when using EO sensors such as shadows, human clothing that is almost with the background. To solve this problem, this method proposes a solution that thermal imagery can be used to avoid these disadvantages. Unlike EO sensors, thermal sensors create image from radiation energy emitted by objects. Therefore, thermal sensors extract human silhouette better than the EO sensors. In this paper, author propose a method which will give us better result for human extraction by making fusion of thermal and color image.

In this paper, author proposes a hierarchal correspondence search method that is based on Genetic Algorithm (GA) [14]. There are 5 main parts in this method: Image Transformation model, Preliminary human silhouette extraction and correspondence initialization, Automatic Image Registration, Sensor Fusion, and Registration of EO/IR sequence with multiple objects. Figure 3 shows us block diagram of proposed method.

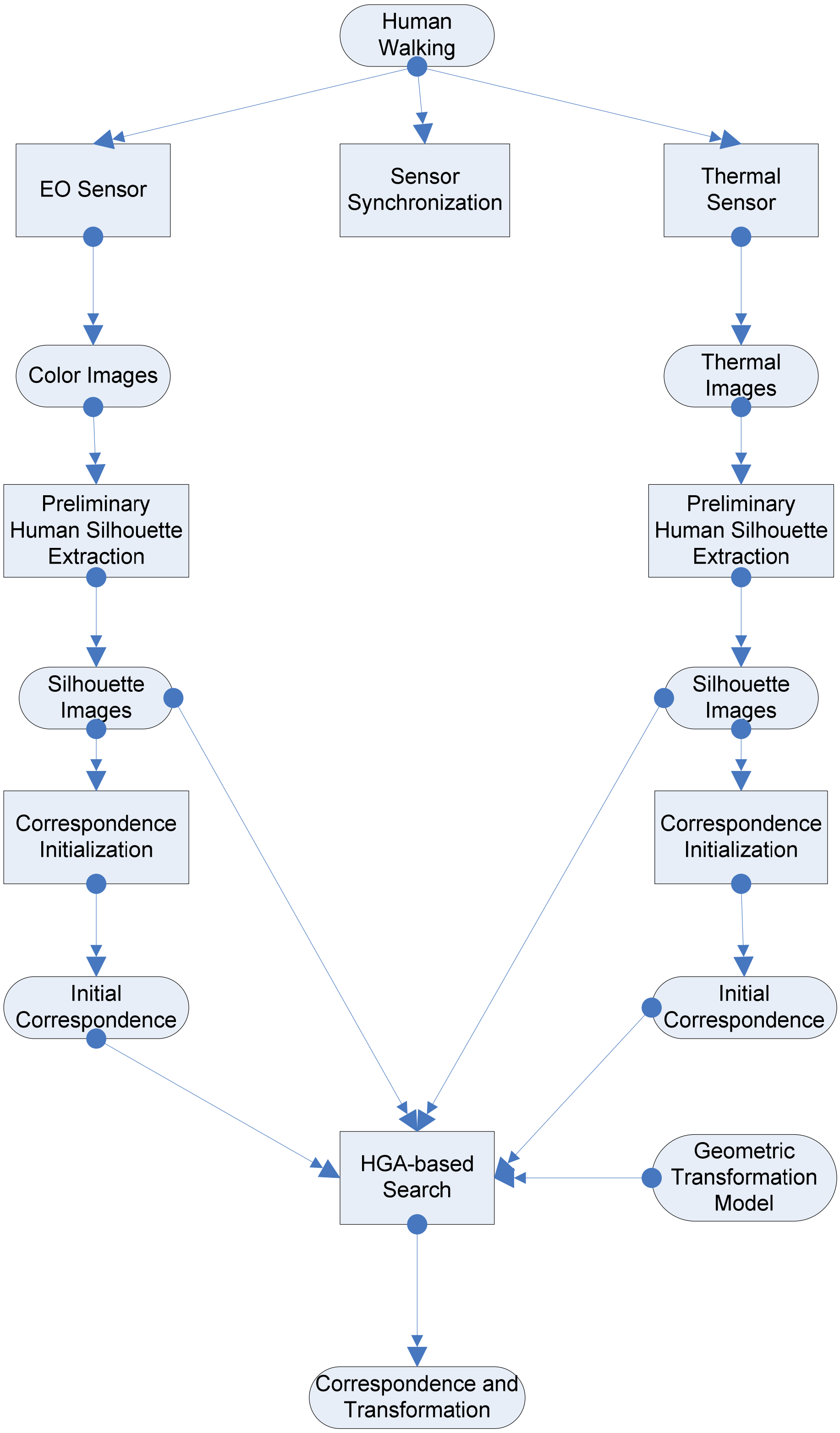

Human Walking
EO Sensor
Sensor Synchronization
Thermal Sensor
Color Images
Thermal Images
Preliminary Human Silhouette Extraction
Preliminary Human Silhouette Extraction
Silhouette Images
Silhouette Images
Correspondence Initialization
Correspondence Initialization
Initial Correspondence
Initial Correspondence
HGA-based Search
Geometric Transformation Model
Correspondence and Transformation

Figure 3: Hierarchical genetic algorithm-based multi-modal image registration approach [14].

Image Transformation model: The EO and thermal camera are placed as close as possible to capture desired image which human is moving. However, EO and thermal camera do not capture the same FOV exactly. Therefore, 2D transformation can be done from visible domain to thermal domain.

To model into 2D, rigid transformation can be used. In rigid transformation, the distance between 2 points in colored image should be preserved while mapping these 2 points into the thermal domain.

Mapping can be done as follows:

$$\begin{pmatrix} X' \\ Y' \end{pmatrix} = s \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix} + \begin{pmatrix} \Delta X \\ \Delta Y \end{pmatrix} \tag{3.4}$$

where θ is a rotation angle, s is a scaling factor $(\Delta X,\ \Delta Y)^T$ is the translation matrix, *(X,Y)* is 2D point colored image plane, and *(X',Y')* is 2D point thermal image plane.

Preliminary human silhouette extraction and correspondence initialization: Author assumes that there is only one human at any time in the scene and only human can walk in the scene. Under this assumption, background subtraction method can be used to extract silhouettes of human. Mean and variance of all colored frames that contains only background are calculated for background subtraction. Then, it is assumed that background has Gaussian distribution at each pixel (X, Y).

Background subtraction is applied for both thermal and visible imagery.

After silhouette of human are extracted from both of thermal and visible domain by using background subtraction model, the centeroid of the silhouette human can be computed as initial correspondence between each pair of color and thermal images.

Automatic Image Registration: There are two steps in automatic image registration between visible and thermal domain: model parameter selection and parameters estimation on Hierarchical Genetic Algorithm (HGA).

There are so many phenomenological differences between thermal and visible domain to select model parameters. However, human silhouette gives us many hints about finding correspondence points. Human body silhouette regions provide a more reliable correspondence between visible and thermal domain. Therefore, author of [14] propose a method that finds matches between transformed color silhouette and thermal silhouette with estimating parameters set ***p*** to maximize:

$$\text{Similarity}(\text{p}; I_i; C_i) = \prod_{i=1}^{N} \frac{\text{Num}\left(T_{C_i;p} \cap I_i\right)}{\text{Num}\left(T_{C_i;p} \cup I_i\right)} \tag{3.5}$$

where *I* is binary image of the silhouette from thermal domain, *C* is binary image of the silhouette from visible domain, $T_{C;p}$ is transformed binary image of *C* by rigid transformation with parameters ***p***, *N* is the number of color and thermal image pairs, and *Num(A)* is the number of silhouette pixels in binary image.

We know from equation 3.4, parameters of rigid transformations is elements of 2D linear transformation. Therefore, to determine range of parameters, we need to determine 2 corresponding pair of points from thermal and visible domain. Centroid of the silhouette is determined as an initial corresponding point from previous part. The second corresponding point is determined under the assumption of planar surface that human walks along the same direction and human body surface in each frame lies on the same plane over whole sequence. Therefore, we can choose a centroid of human silhouette from different thermal image frame as second corresponding point. These two corresponding points should be as far as possible from each other to get better registration result. After choosing 2 corresponding points from thermal and visible domain, we can estimate parameters.

To maximize equation 3.5, Genetic Algorithm (GA) can be used. GA is iterative method that calls the hypothesis and combines its best part until find best fitness for problem. However, it is not enough to use single GA for this problem. In this method, author proposes HGA based search method to estimate model parameters within series of windows whose size is reduced in a time.

Sensor Fusion: In this phase, fusion techniques in [14] are used to improve the accuracy of human silhouette extraction. Max rule, Min rule, Sum rule and Product rule procedure can be used as fusion method in the extraction of human silhouette according to author as a following way:

$$\begin{aligned}&\text{Product rule}: (X,\ Y) \in S,\\ &\mathit{if}\ P(S \mid c(X,Y))P(S \mid t(X,Y)) > \tau_{product}\end{aligned} \tag{3.6}$$

$$\begin{aligned}&\text{Sum rule}: (X,\ Y) \in S,\\ &\textit{if}\ P(S \mid c(X,Y)) + P(S \mid t(X,Y)) > \tau_{sum}\end{aligned} \tag{3.7}$$

$$\begin{aligned}&\text{Max rule}: (X,\ Y) \in S,\\ &\textit{if}\ \max\{P(S \mid c(X,Y)), P(S \mid t(X,Y))\} > \tau_{\max}\end{aligned} \tag{3.8}$$

$$\begin{aligned}&\text{Min rule}: (X,\ Y) \in S,\\ &\textit{if}\ \min\{P(S \mid c(X,Y)), P(S \mid t(X,Y))\} > \tau_{\min}\end{aligned} \tag{3.8}$$

where (X,Y) is 2D points, S is the human silhouette, $\boldsymbol{c}$ is the color value vector, t is the thermal value vector and $\tau_{product}$, τ_{sum}, τ_{max}, and τ_{min} are threshold values.

Probability can be estimated as a following ways

$$P(S \mid c\,(X,Y)) = 1 - e^{-\|c(X,Y) - \mu_c(X,Y)\|^2} \tag{3.9}$$

$$P(S \mid t\,(X,Y)) = 1 - e^{-\|t(X,Y) - \mu_t(X,Y)\|^2} \tag{3.10}$$

where μ_c is the mean for colored background and μ_t is the mean for thermal background.

According to author's experiment results, the product, sum, and max fusion methods increase the accuracy of detection instead of using color and thermal classifiers only.

Registration of EO/IR sequence with multiple objects: This method is tried with a scenario that contains only one human is walking at any time and cameras are fixed during certain time. The transformation model presented in equation 3.4 can be used for image registration of thermal and visible images with using the same camera setup. Number of object does not affect the registration process. Therefore, registration and detection can be made for image pairs that include multiple moving objects with same camera setup.

Author in [14] evaluates the different fusion methods. This evaluation helps us to understand which fusion method works better, how they work, and where they can be used. Also, we examine the automatic image registration method used in [14] to apply to our method. However, we decide that this method is not appropriate for our methods after capturing videos. Instead of this registration method, we use manual registration by selecting reference points for homography.

3.4 Person Surveillance Using Visual and Infrared Imagery

In [15], authors propose an algorithmic framework for detecting people using thermal and visible image sequences. Authors register images accurately with using two thermal and 2 visible band cameras. This method uses Support Vector Machine (SVM) to detect people with using Histogram Oriented Gradient (HOG) features from color and thermal domains. Also, disparity based detectors are created with learning relationship between person size and depth. This framework is an extended version of multispectral framework proposed in [16].

There are four main parts in this framework: Image Registration with Trifocal Tensor, Annotation, Image Features, Learning and Classification.

Image Registration with Trifocal Tensor: In this phase, three cameras (thermal camera, color camera, and camera that is used to combine thermal and visible band images) are used for registration process. Disparity estimates from thermal and visible band images are used to register corresponding points in third image with trifocal tensor that is matrices of correspondence between three images.

Normally, trifocal tensor is calculated with only seven point-point-point correspondence. However, author use more than seven correspondence to smooth errors and obtain better result.

Annotation: In this phase, positive and negative samples surrounding with bounding box are extracted from registered images for classification process. Author takes 2:5 aspect ratio for bounding box of samples for consistency. Also, samples can be multiplied with changing scale factor for image.

Image Features: This phase is the feature extraction part of method. We need to extract HOG feature similar to feature proposed in [17] for both thermal and visible image samples. These features are about relevance of edges in terms of gradient and spatial position. First, each sample resize to same size and X x Y x θ is computed where X is the width, Y is the height, and θ is the gradient of orientation histogram. However, disparity image is not suitable for HOG feature extraction. Because many of positive people samples in disparity image has no enough edge information. Therefore, we need to find any other information that is used to differentiate people from other things. In the parity image, size of person is distributed around a mean value. So, linear correlation can be modeled between size of bounding box that covers person and median of region defined by this bounding box in disparity image.

This linear line can be expressed as $Ax + By + C = 0$ where A, B, C are the line parameters, x is the bounding box height and y is the median of disparity image. So distance to this line can be computed as:

$$\Delta L = \frac{|Ax + By + C|}{\sqrt{A^2 + B^2}} \tag{3.11}$$

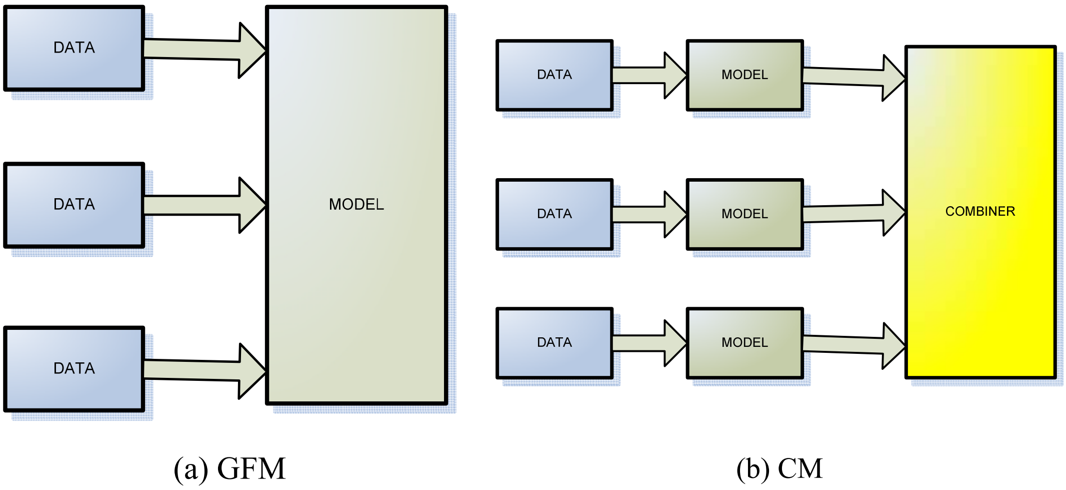
Learning and Classification: During the feature extraction, author used different feature for thermal, visible pair and disparity image. Therefore, we need two classifiers, one of them is for HOG features from thermal and visible images and the other one is for disparity image. For final classification, two classifier results are combined probabilistically. Classifier for HOG features is trained by using SVM using radial basis function kernels [18].

In [15], author extracts feature for classification and fusion process. Similarly, we extract features from thermal and visible domain and then fuse these features. Also, this method is used for surveillance applications so we examine method proposed in [15]. Also, SVM used in [15] is very powerful tool for classification and it can be used in the future for our method to discriminate living object as a human or not.

3.5 Comparison of Fusion Methods for Thermo-Visual Surveillance Tracking

In [19], author analyzes the advantages of using thermal imagery in addition to visible imagery for tracking objects in surveillance scenarios using appearance models. This method evaluates different fusion methods to obtain better result. There are three main parts: Data Capture and Alignment, Appearance Model, and Fusion Model. For our thesis, the important part is the fusion model. So, we tried to give information about fusion methods that author in [19] investigates.

There are two main types of fusion models: General Fusion Model (GFM) and Combination Model (CM) shown in Figure 4.



(a) GFM (b) CM

Figure 4: Fusion Models [19]

General Fusion Model: Fusion methods in this model are pixel based fusion. Each pixel is modeled as k-dimensional Gaussian where k is the number of data sources has an importance value.

Combination Model: Fusion methods in this model is the model based fusion. That is, fusion performs in model level. There are different strategies for performing fusion.

Simple and weighted averaging: Weighted averaging is calculating as follows:

$$S_{average}(X) = \sum_{i=1}^{M} \omega_i S_i(X) \tag{3.12}$$

where $S_i(X)$ is the similarity between image X and ith model, w_i is the weight of model, M is the number of model. For simple averaging, $w_i = 1/M$.

Similarity Score Product: Similarity score product is calculating as follows:

$$S_{product}(X) = \prod_{i=1}^{M} S_i(X) \tag{3.13}$$

where $S_i(X)$ is the similarity between image X and ith model, M is the number of model.

Min and Max score fusion: Min fusion method is based on the idea that choose least bad match according to model's confidence level. However, max fusion is aimed to choose good match that has high confidence level.

Dynamic Weighting: This method is based on the idea that updates the model's weight according to how well the model is discriminated within the search space. Update mechanism works as follows:

$$\hat{s}_i = s_i \Big/ \sum s_p \qquad (3.14)$$

$$\omega_i = \alpha\omega_i + (1-\alpha)\hat{s}_i \qquad (3.15)$$

where α is the update parameter, s_i and w_i are the similarity score and weight of ith model respectively.

According to author's experiments, similarity score product fusion method in Combination Model gives better result than others.

Author in [19] examines the different fusion methods for tracking utility. To understand fusion methods, we investigate this paper [19]. It gives us very useful information about image fusion methods.

CHAPTER 4

LIVING/NONLIVING DISCRIMINATION FOR ABANDONED OBJECT DETECTION

4.1 Overview

In this chapter, we will explain the proposed algorithm used to detect abandoned object and discriminate as a living or nonliving object. Firstly, we explain the detection of abandoned object with using dual foregrounds approach. After that, we explain the algorithm that is used to extract living object in thermal domain by using Local Intensity Operation (LIO) and Mean Value Thresholding (MAT). Then, we investigate segmentation of objects in both thermal and visible domain to improve the accuracy of object extraction. Finally, we explain our fusion method that is used to discriminate abandoned object as a living or nonliving object and error correction methods to improve the result. Figure 5.

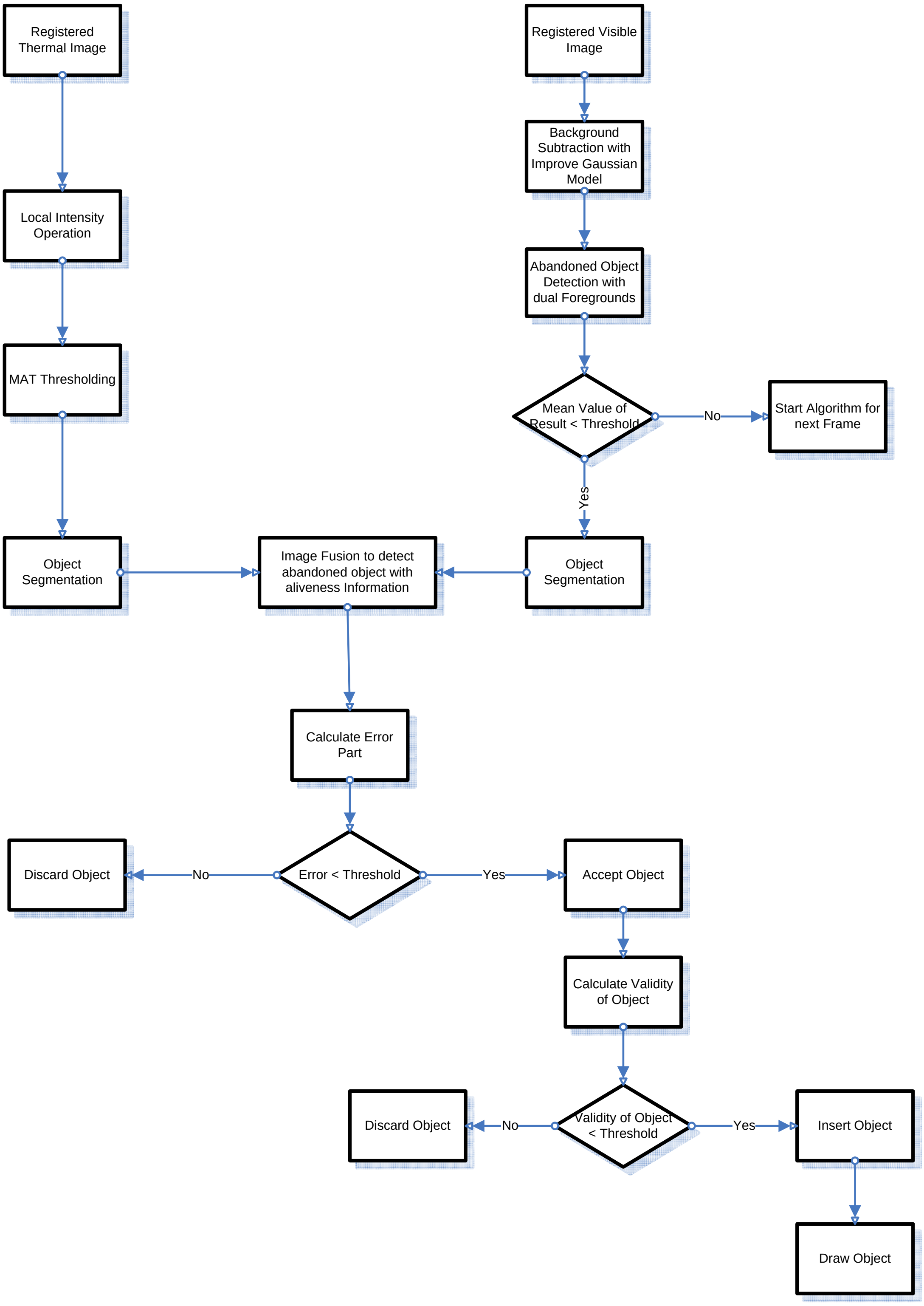

Registered Thermal Image
Local Intensity Operation
MAT Thresholding
Object Segmentation
Registered Visible Image
Background Subtraction with Improve Gaussian Model
Abandoned Object Detection with dual Foregrounds
Mean Value of Result < Threshold
No
Start Algorithm for next Frame
Yes
Object Segmentation
Image Fusion to detect abandoned object with aliveness Information
Calculate Error Part
Error < Threshold
No
Discard Object
Yes
Accept Object
Calculate Validity of Object
Validity of Object < Threshold
No
Discard Object
Yes
Insert Object
Draw Object

Figure 5: Flowchart of the Proposed Method

4.2 Abandoned Object Detection Using Dual Foregrounds Approach

In this thesis, we used the dual foreground approach [20] to detect abandoned item in the colored image sequence. Dual foreground approach is a method that uses two background models: long-term and short-term background. We use Improved Gaussian Background Subtraction Model [9] explained in section 2.3 as a background subtraction method. To obtain two different background models, we need to adjust parameters method. For short-term background model we select $\alpha = 0.02$, threshold on the squared Mahalanobis distance = 16 that means 4 standard deviation, number of Gaussian = 4, $c_f = 0.1$ and initial standard deviation = 11. In long-term model, $\alpha = 0.0002$, threshold on the squared Mahalanobis distance = 16, number of Gaussian = 4, $c_f = 0.1$ and initial standard deviation = 11. We choose threshold for squared Mahalanobis distance as 4 standard deviation since our model should be 99% confidence level to detect abandoned object correctly. Figure 6 shows the long-term and short-term background results. Long-term gives us more segmented foreground objects than short-term since long-term background model updates background more slowly than short-term background model.

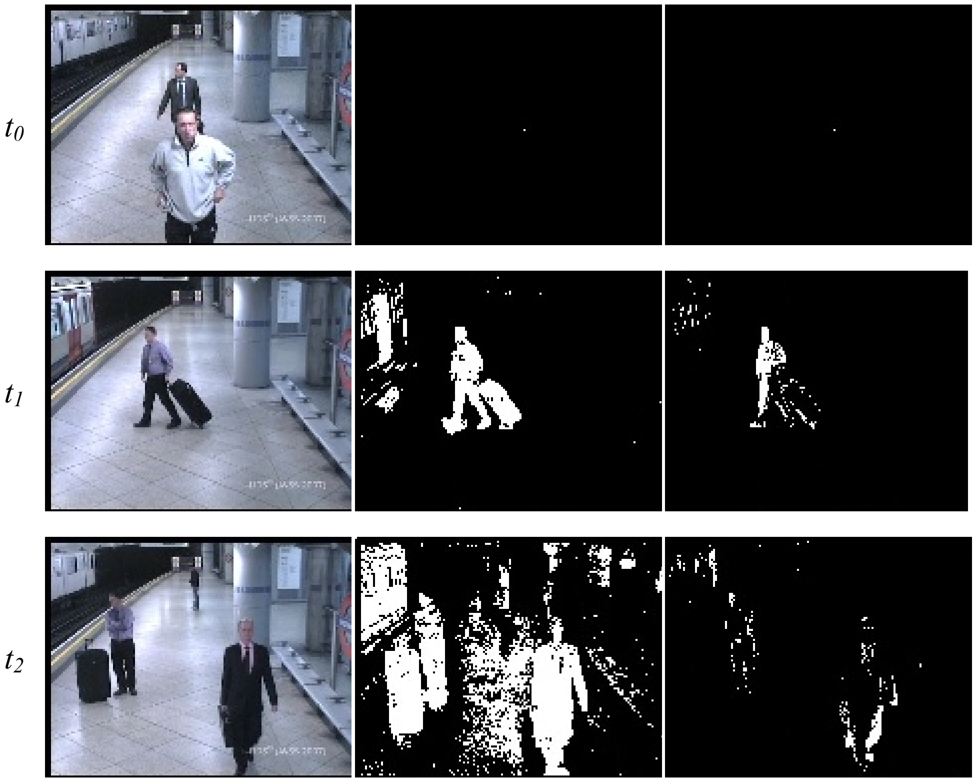
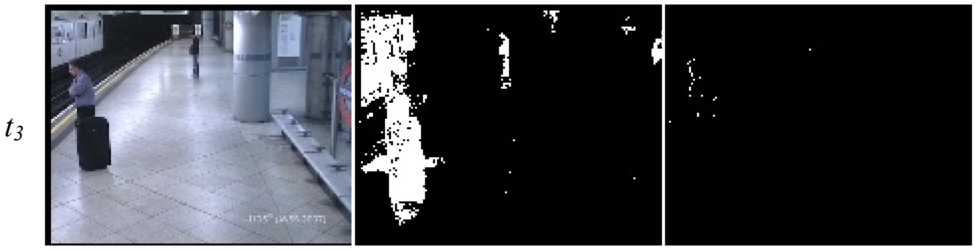


Figure 6 (cont.)



(a) Current Image

(b) Long-term Foreground image

(c) Short-term Foreground image

Figure 6: (a) Visible band current image (b) Result of long-term foreground subtraction model, (c) Result of short-term foreground subtraction model

Abandoned objects are the temporally static objects in the background. Pixels of abandoned object are detected as a background for higher learning rate or detected as foreground in shorter learning rate. [20] uses these facts to detect abandoned object by using dual foreground approach shown in Figure 7.

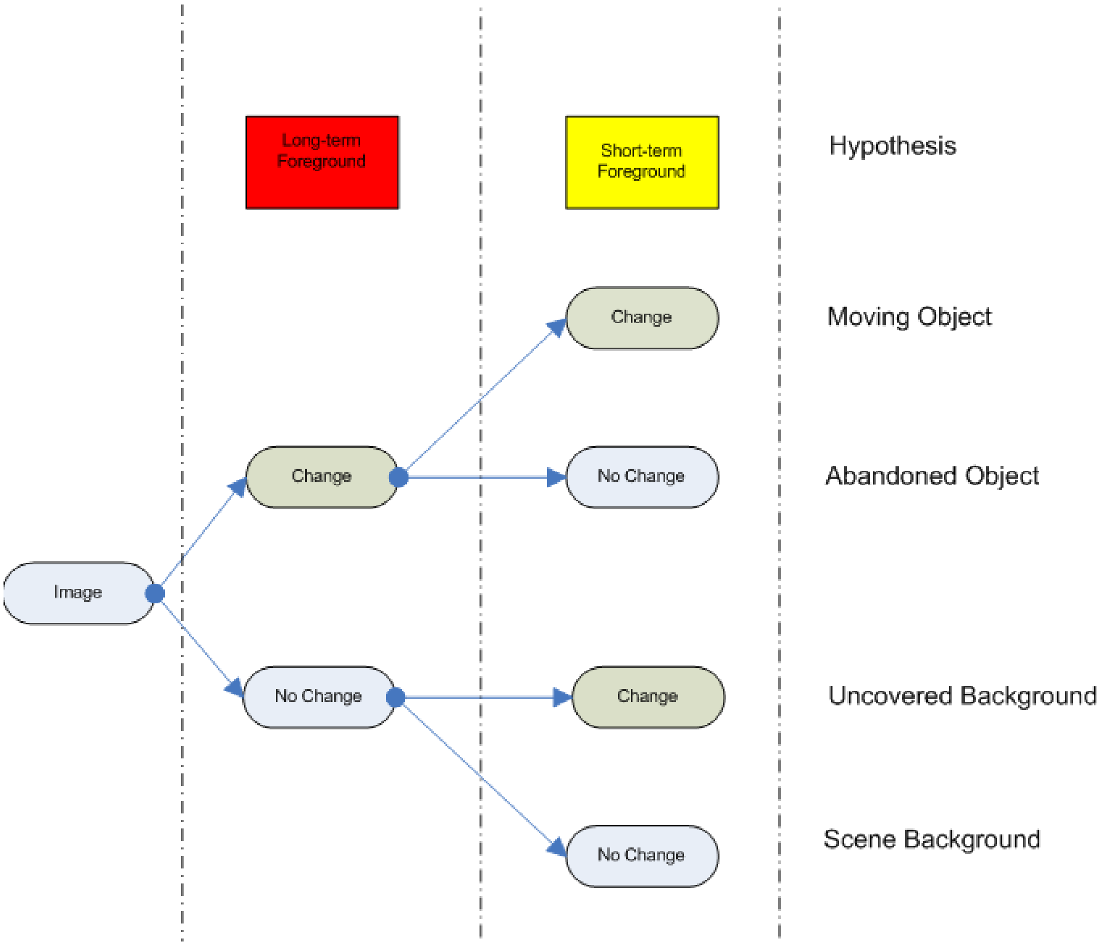


Figure 7: Hypothesis on long-term and short-term foregrounds [20]**.**

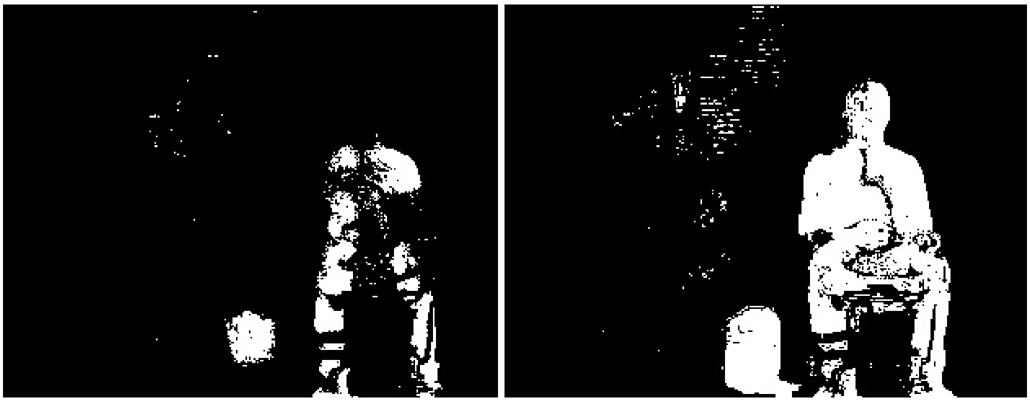
At every frame, we estimate 2 backgrounds B_L and B_S , 2 foregrounds F_L and F_S where *F(x,y)* = 1 for long-term and short-term. F_L includes moving objects and abandoned objects and F_S includes moving objects, noise and so forth. To extract abandoned objects from background, we need to use following rules.

1. If $F_L(x,y) = 1$ and $F_S(x,y) = 1$, then pixel at (*x,y*) is pixel of moving object.
2. If $F_L(x,y) = 1$ and $F_S(x,y) = 0$, then pixel at (*x,y*) is pixel of abandoned object.
3. If $F_L(x,y) = 1$ and $F_S(x,y) = 1$, then pixel at (*x,y*) is pixel of uncovered background object.
4. If $F_L(x,y) = 1$ and $F_S(x,y) = 1$, then pixel at (*x,y*) is pixel of scene background object.

Therefore, we can find abandoned object by finding pixels that satisfy rule 2. After finding pixel belongs to abandoned object, we set that pixel value to 255 for both long-term foreground F_L and short-term foreground F_S. After this operation, we have two binary foreground images. We know that abandoned objects are the temporal static objects and stay stationary at some time. Therefore, we need to create evidence image that is used to decide whether pixel belongs to abandoned object or not. Evidence image can be created as follows:

$$E(x,y) = \begin{cases} E(x,y)+1 & F_L(x,y)=1 \wedge F_s(x,y)=0 \\ E(x,y)-k & F_L(x,y)\neq 1 \vee F_s(x,y)\neq 0 \\ \max_e, & E(x,y) > \max_e, \\ 0, & E(x,y) < 0, \end{cases} \tag{4.1}$$

where *E(x,y)* is the pixel value at *(x,y)*, *k* is the decay constant, and max_e is the maximum value that can be pixel value of evidence image. If pixel belongs to foreground object in long-term background and background in short-term background, then we increase the corresponding pixel value of evidence image. Otherwise, we need to decrease the corresponding pixel value as much as decay constant. If corresponding pixel value reaches to max_e, then we can decide that this pixel belongs to abandoned object. In this thesis, we will use binary image pixel values of that are one if it belongs to foreground, otherwise, it is set to zero. Figure 8 shows binary image for abandoned object.



(a) Abandoned object pixels at t_0 (b) Abandoned object pixels at t_1

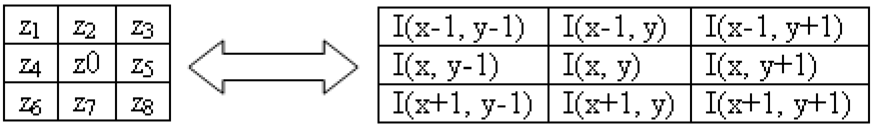
Figure 8: Result of Abandoned Object Detection with Using Dual Foregrounds

4.3 Living Object Extraction

After detection of abandoned object, we need to determine whether object is living or not. However, we cannot extract feature that is used to discriminate object as a living or not from visible domain. Therefore, we need to use thermal imagery registered with visible band images. Because thermal domain construct image from radiated energy emitted by objects. Living objects emit more energy compared to nonliving objects. So, pixels of living object are much brighter than pixels of nonliving objects. The method [21] uses this fact to extract brighter pixels from gray scale image and we can use this simple algorithm to extract living objects from thermal images.

The method [21] uses local intensity operation (LIO) based on local intensity of neighborhood operation. LIO has two modes to discriminate brighter pixels from darker ones. First mode uses a method that brightens the bright pixels and darkens the dark pixels. Second mode uses a reverse of first method. That is, it uses a method that darkens the bright pixels and brightens the dark pixels. In our thesis, we used the first mode of LIO.

Let *I(x,y)* be pixel in thermal image written as z_0 , and neighbors of it *I(x-1,y-1), I(x-1,y), I(x-1,y+1), I(x,y-1), I(x,y+1)* , *I(x+1,y-1), I(x+1,y), I(x+1,y+1)* be written as z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8 respectively shown in Figure 9.



z1	z2	z3
z4	z0	z5
z6	z7	z8

I(x-1, y-1)	I(x-1, y)	I(x-1, y+1)
I(x, y-1)	I(x, y)	I(x, y+1)
I(x+1, y-1)	I(x+1, y)	I(x+1, y+1)

Figure 9: LIO Window [21]

Let *Z* be product of pixels defined in LIO Window.

$$Z = \prod_{k=0}^{8} Z_k \qquad (4.2)$$

We can create new image according to *Z* for each pixel in thermal image by defining intensity brightness operation as follows:

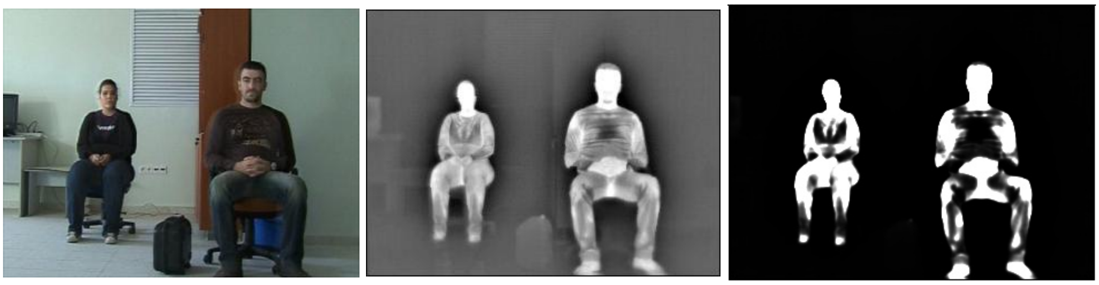
$$g(i,j) = Z \qquad (4.3)$$

where g(i, j) is the pixel value at (i,j) of new image. This operation can be applied from i = 2 to i = number of rows – 1 and from j = 2 to j = number of column -1. That is, we cannot apply this convolution process to first row, first column, last row, and last column. However, we can copy second row to first row, second column to first column, column before last column to last column, and row before last row to last row. After that, we need to normalize these image pixels to gray-scale range. The normalization process can be done by dividing these pixels to maximum pixel value within this image. To get intended result, this process can be done more times.

This operation increases the brightness of bright pixels and the darkness of the dark pixels. However, to get better result, we need to segment this new image by using Mean Absolute Thresholding (MAT). MAT is a simple segmentation mechanism that uses threshold value calculated as follows:

$$T = \text{round}\left[\frac{i_L - i_1}{2}\right] \qquad (4.3)$$

where *T* is the threshold value, i_L is the maximum pixel value, i_l is the minimum pixel value. We know that thermal image is 8-bit gray-scale image. So i_L = 255 and i_l = 0 for thermal image. *T* is always 128 for thermal image. Figure 10 shows the result of this operation.



(a) Colored Image (b) Thermal Image (c) Binary Image generated by LIO

Figure 10: Results of LIO for thermal images

4.4 Segmentation for Objects

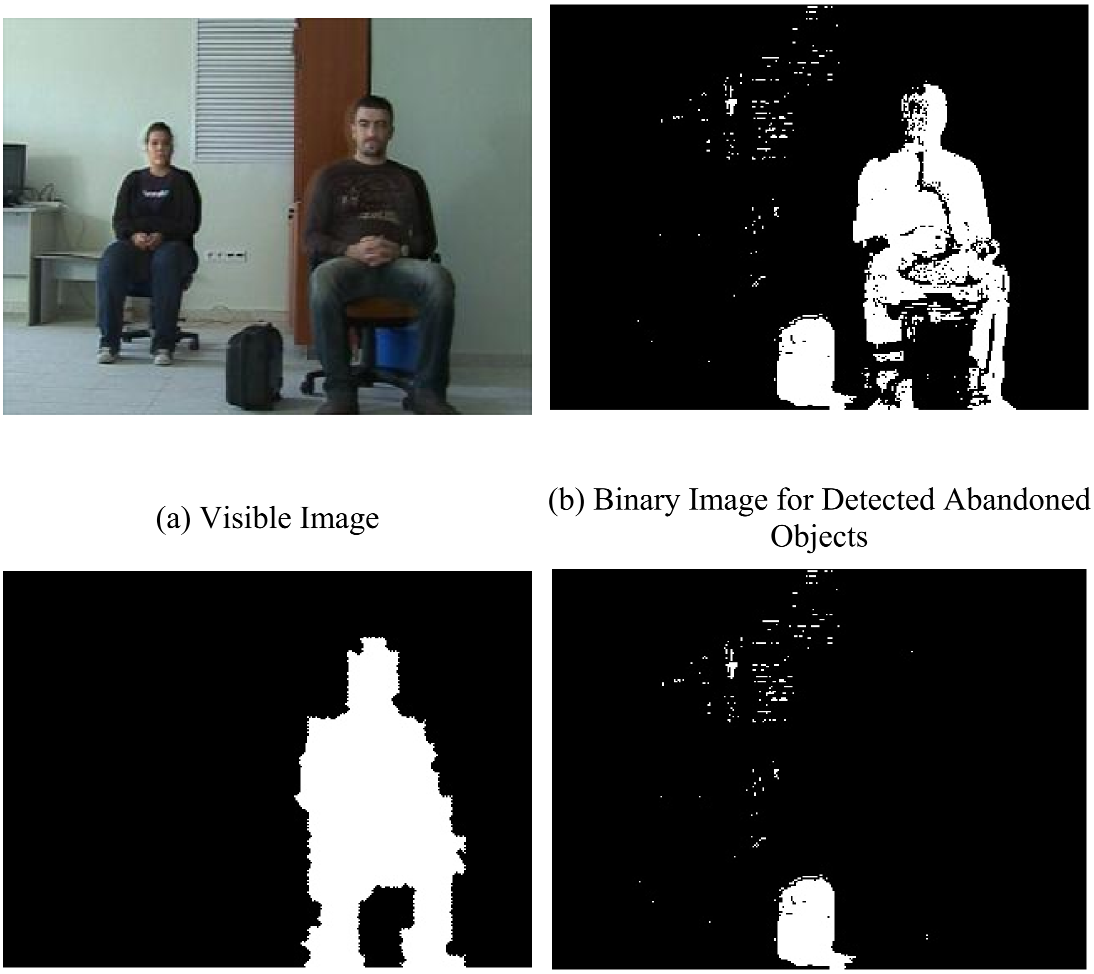
In previous sections, we found abandoned object in visible domain shown in Figure 8 and living object in thermal domain shown in Figure 10. As you can see from these figures, there are holes on binary image. To make these objects one piece, we need to complete and close these holes in binary images by using some morphological operation that Matlab provide us. This method consists of four main phase: Complete, Close, Boundary Extraction and Object Filling, and Post-processing for missing part.

Complete: In this phase, we need to complete binary images. First, we need to fill the holes in binary image by using **'imfill'** function in Matlab toolbox that can be used for filling the holes in binary image. Then, we need to remove noise from this binary image. We assume that size all abandoned object including human and luggage is greater that the 1% percent of whole image. For this thesis, we use 320x240 sized images, so we can take this threshold value that will be used to remove noisy pixels as 320x240x0.01 = 768. Matlab provide us **'bwareaopen'** function that deletes all connected components that have fewer numbers of pixels than this threshold value from binary image.

Close: After completion phase, we need to close these objects. Matlab provide us **'imclose'** function that performs morphological close operation on binary image. However, this operation may result in new holes in binary image. To make sure that everything is closed in this binary image, we need to close these new holes by using **'imfill'** function again.

Boundary Extraction and Object Filling: In this phase, we need to extract boundary of object and create mask for this image by filling this boundary. Matlab provide us **'bwboundaries'** function that find exterior boundaries of object and fill these boundaries.

Post-processing for missing part: After the boundary extraction, most of objects in binary image are filled. However, some of object may disappear due to complete operation. That is, some of abandon objects are not present in binary image because of the complete operation. We need to find and segment all abandoned objects. Therefore, we develop simple approach to fill missing objects. First, we can extract missing object pixels by comparing the abandoned object mask with segmented object mask shown in Figure 11 (b) and (c) respectively. If the pixel in abandoned object mask belongs to foreground and but corresponding pixel in segmented object mask does not belong to foreground, then we mark that pixel as a pixel of missing object as shown in Figure 11 (d).



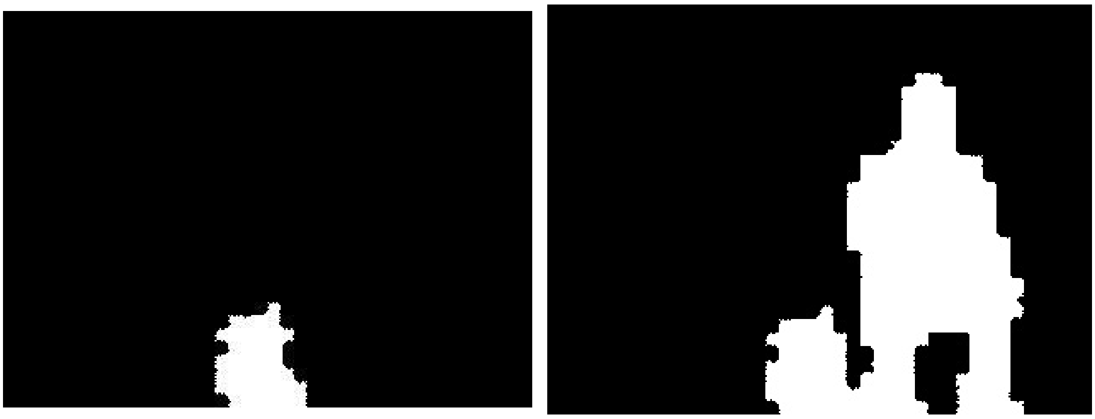
(a) Visible Image

(b) Binary Image for Detected Abandoned Objects

(c) Result of Boundary Extraction Part

(d) Binary Image for Missing part

Figure 11 (cont.)



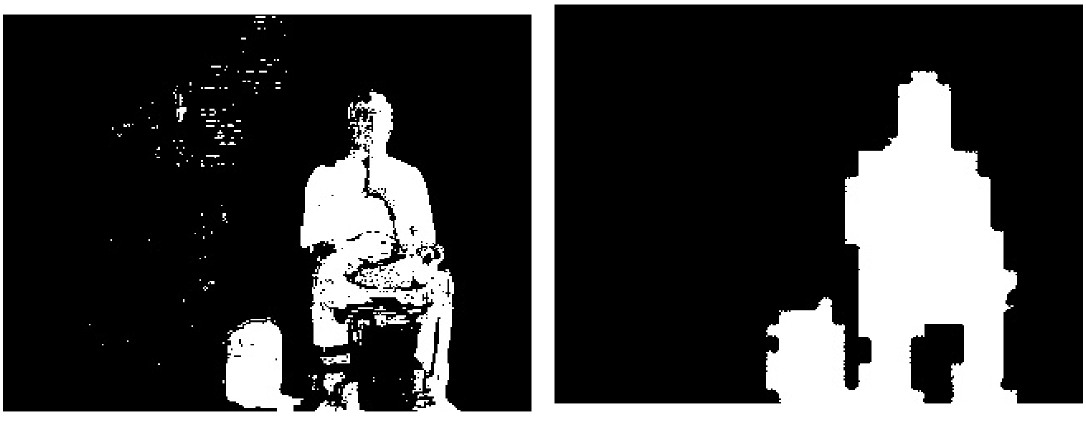
(e) Segmentation Result for Missing Part

(f) Merged Image

Figure 11: Result of Preprocessing part for Missing Object

After extracting the missing object, we segment missing object by applying the complete, close and boundary extraction phase onto this binary image. Then, we merge first segmented image with the last segmented image so that we find and segment all abandoned objects shown in Figure 11 (f).

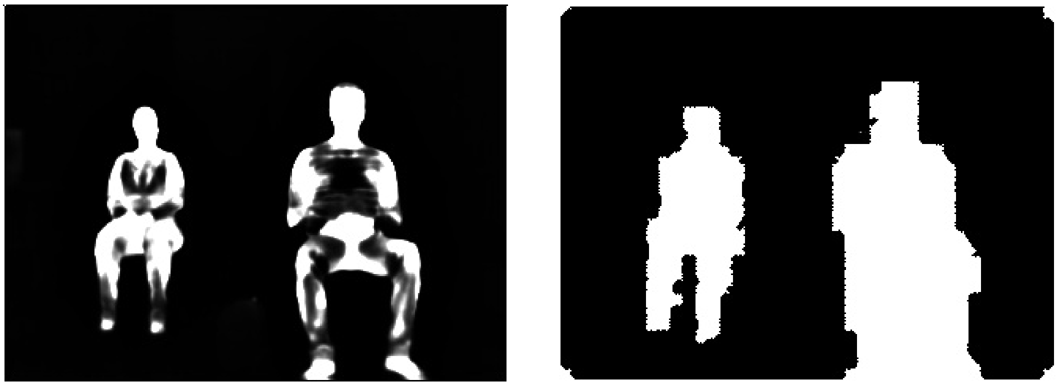
We have abandoned object mask and living object mask after segmenting objects and results shown in Figure 12.



(a) Detected Abandoned Objects in Visible Domain

(b) Detected Abandoned Objects in Visible Domain After Segmentation

Figure 12 (cont.)



(c) Thermal Image After LIO

(d) Thermal Image After LIO After Segmentation

Figure 12: Object Segmentation Result

4.5 Living/Nonliving Discrimination for Abandoned Objects

This part is fusion part that fuses features of object coming from thermal and visible domain. In previous section, we found binary mask images of abandoned object and living objects. We want to discriminate abandoned object as a living object or not. Aliveness of objects can be extracted from the thermal images and we can determine abandoned object as alive or not by fusing abandoned objects mask and living objects mask as shown in Figure 13.

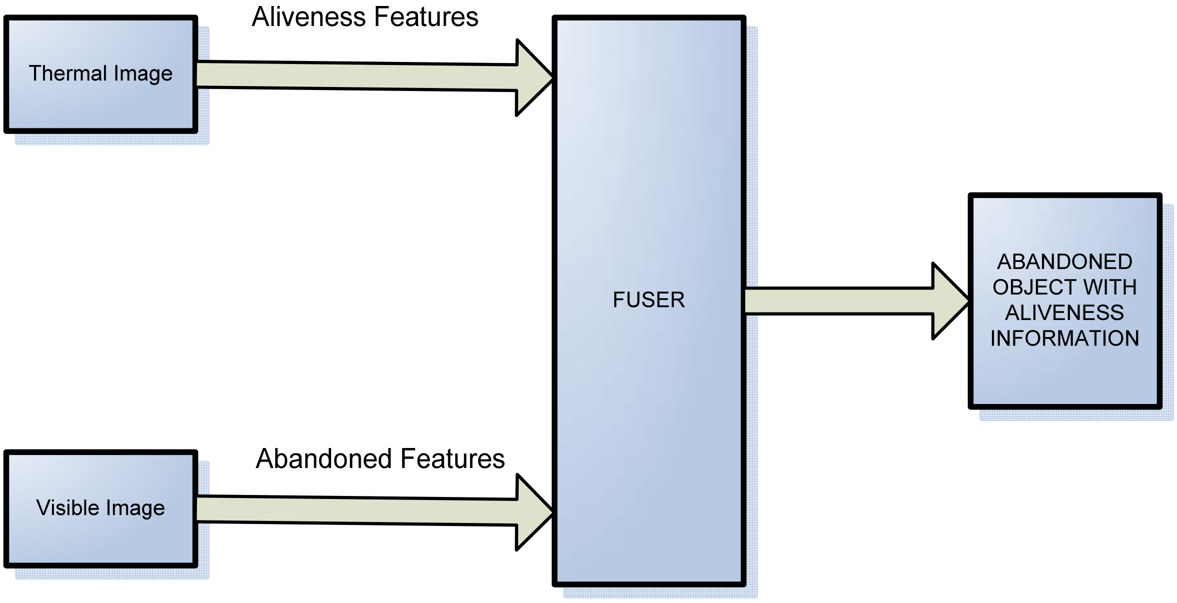
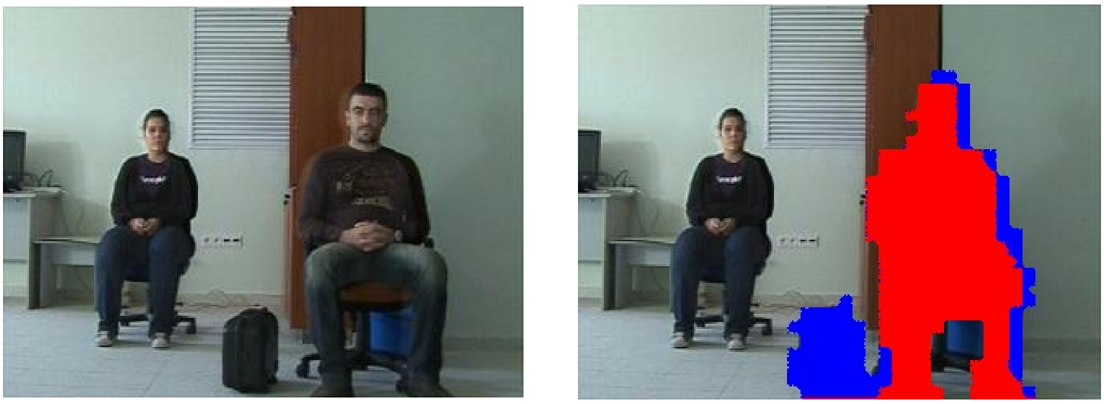


Figure 13: Thermal and Visible Imagery Fusion for Abandoned Object Detection with Aliveness Information

Fuser combines features coming from both thermal and visible domain and classify objects as a nonliving or living abandoned objects with equation 4.4.

$$G(x,y)=\begin{cases} nonliving\ abandoned\ object, E_s(x,y)\neq 0 \wedge F_s(x,y)=0 \\ living\ abandoned\ object, E_s(x,y)\neq 0 \wedge F_s(x,y)\neq 0 \end{cases} \tag{4.4}$$

Where *G(x, y)* is a fusion result for pixel at *(x, y)*, $E_s(x, y)$ is the pixel value at *(x, y)* of image result from section 4.2, and $F_s(x, y)$ is the pixel value at *(x, y)* of image result from section 4.3. Fusion results are shown in Figure 14.

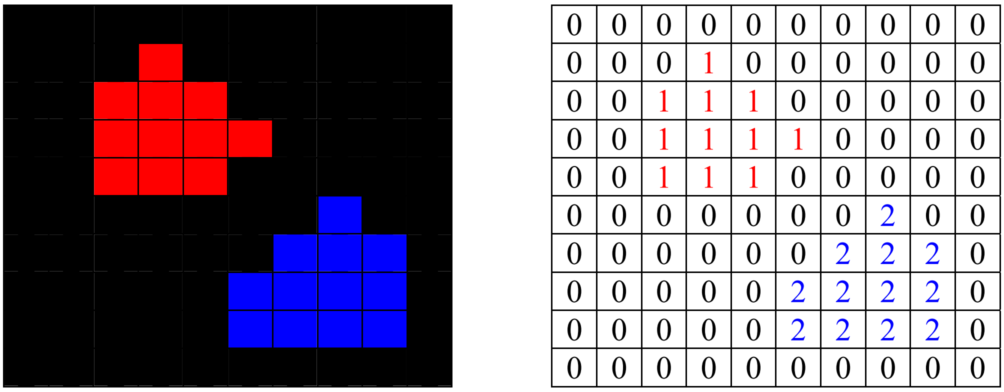


(a) Visible Domain Image

(b) Fusion Result Image

Figure 14: Aliveness Discrimination Result of Abandoned Objects

In Figure 14, there are some discrimination errors around the person. These errors are caused by manual image registration between thermal and visible domain. After image registration, objects in thermal domain do not cover exactly the corresponding objects in visible domain due to inaccuracies in the registration. We propose a simple method to remove these kinds of errors. We know that all pixels of objects are connected to each other with a neighboring relationship. Therefore, we can separate objects by using connected component labeling algorithm with 8-connectivity for abandoned object [22].



0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0
0	0	1	1	1	1	0	0	0	0
0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	2	0	0
0	0	0	0	0	0	2	2	2	0
0	0	0	0	0	2	2	2	2	0
0	0	0	0	0	2	2	2	2	0
0	0	0	0	0	0	0	0	0	0

(a) Colored Image

(b) Connected Component Labeling Matrix

Figure 15: Connected Component Labeling

Figure 15 shows how this simple flood-fill algorithm works. In this thesis, we first choose a pixel that belongs to abandoned object but is not assigned to label value. Then, we check the 8 neighbors of that pixel whether it belongs to abandoned object or not. If any neighbors are part of abandoned object, we set same label to these neighbors. This process continues until there is no pixel to be set same label. Table 1 shows the pseudo code of algorithm.

Table 1: Pseudo code of Flood Filling to remove discrimination errors

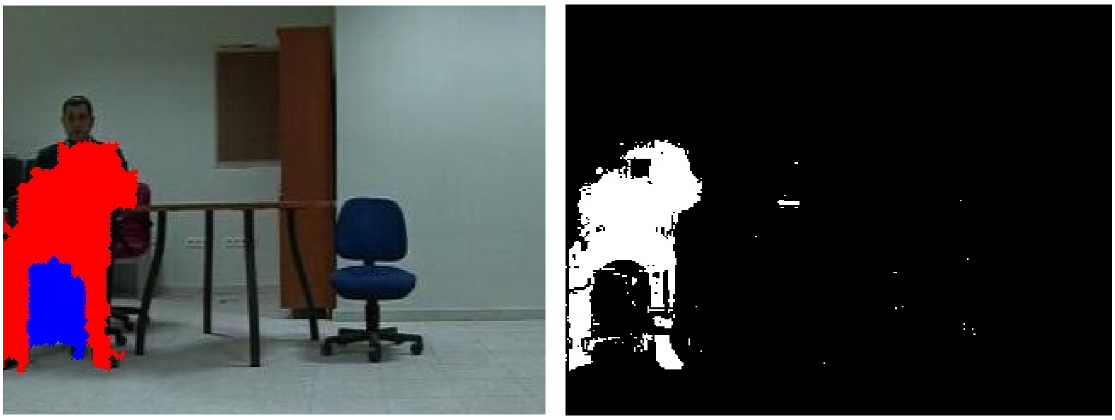
1	Create empty Stack *S*
2	Put coordinate of pixel that belongs to abandoned object but not labeled.
3	Get coordinate *(x,y)* from Stack *S* and remove from Stack *S*
4	If *(x,y)* is inside the image and pixel belongs to abandoned object and not labeled before, then set label value for that pixel and push its neighbors into the Stack *S*.
5	Continue to apply step 3-4 until Stack *S* is empty.

In Figure 14, we obtain 3 objects after connected component labeling: luggage, person, and error around person. After separating the objects by flood filling, we can find a bounding box that surrounds the objects. To eliminate errors around person, we need to calculate density of object in its rectangle as follows:

$$D = N/A_{rect} \tag{4.5}$$

where D is the density of object, N is the number of pixels that object has, and A_{rect} is the area of the rectangle. After finding the density of object, we decide that object is valid if its density is greater than the density threshold and number of pixels that belongs to object is greater than the threshold value for maximum number of pixel. In this thesis, we select the density = 0.4, number of pixel threshold = 1000.

Also, we faced errors due to image noise shown in Figure 16.



(a) Image Fusion Result (b) Corresponding Evidence Image

Figure 16: Fusion Result with Error caused by noise

As shown in Figure 16 (a), there is a detected nonliving object. However, we know that there is no such an object. In Figure 16 (b), there is some noisy pixel in the field where nonliving object is detected. Object Segmentation algorithm detects that field as a hole and fill this hole. Therefore, this field is detected as a nonliving object. To fix this problem, we propose a solution. We know that this area is actually hole in evidence image shown in Figure 16 (b). So, density of pixels that belongs to abandoned object is really low. First, we need to calculate object density in evidence image by using equation 4.6.

$$D_E = N_A/N_T \tag{4.6}$$

Where D_E is density of object in evidence image, N_A is the number of pixels that belongs the abandoned object in evidence image, and N_T is the number of pixels that belongs to whole object detected as a living/nonliving abandoned object as shown in Figure 16 (a). After calculating density of object, algorithm checks whether density is less than the threshold value or not. If it is less than the threshold, then this object is discarded. We choose this threshold value as 0.4.

CHAPTER 5

EXPERIMENTAL RESULTS AND COMPARISONS

5.1 Overview

We have tested the proposed algorithms in indoor environment with different scenarios. The strengths and weakness of the proposed methods in this thesis are evaluated experimentally and illustrated in different test cases. In this chapter, we give detailed information about the testing environment, test data, testing scenarios and results.

5.2 Testing Environment

To test proposed approach, we captured a range of video that reflects different scenarios. Also, we used some video sequence from *AB-Easy, AB-Medium*, and *AB-Hard* included in *i-LIDS* dataset [23] for testing of abandoned object detection.

In this thesis, we use two cameras. One of them is Sony HDR-FX1 HDV Handy Cam (See APPENDIX A) for visible domain and the other one is OPGAL EYE-R640 (See APPENDIX B) for thermal domain. We capture simultaneously in both thermal and visible domain with these cameras. However, after capturing image frames, we saw that FOVs of cameras are different as shown in Figure 17.

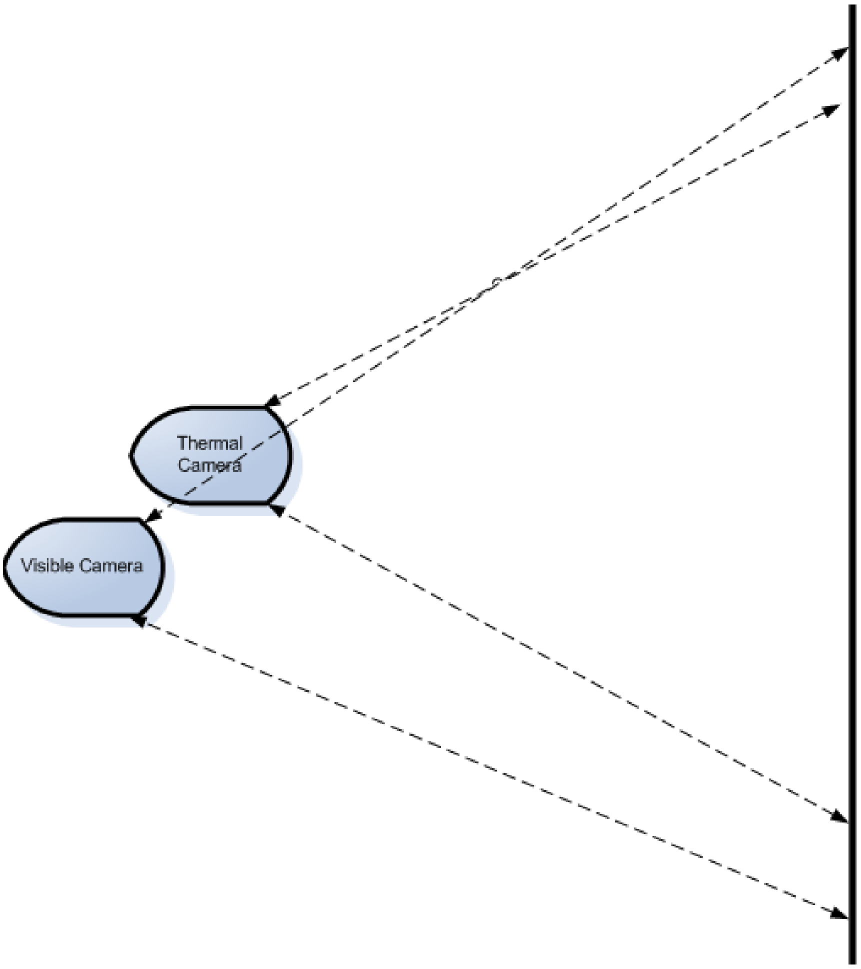
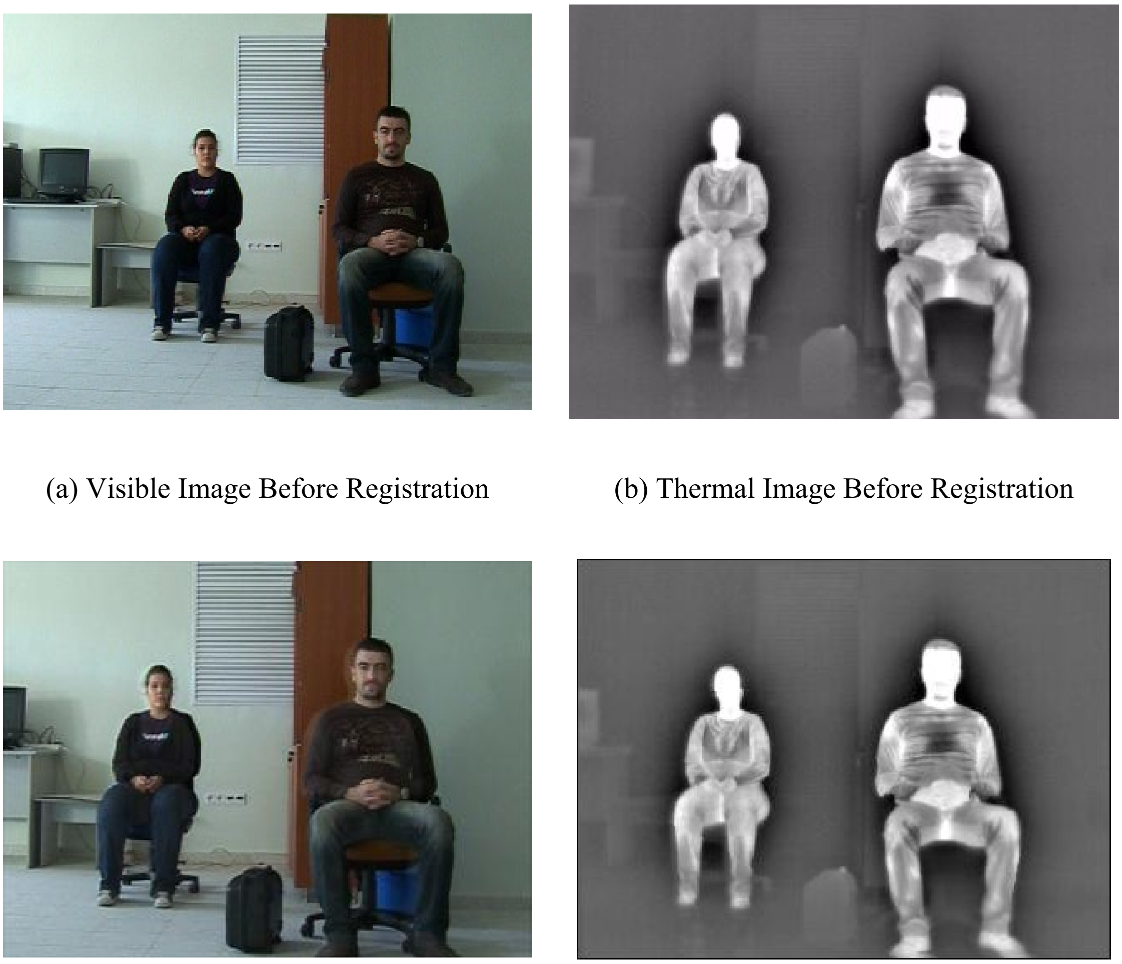


Figure 17: FOV for Thermal and Visible Camera

We need to solve this problem before starting to register images since both images include the same objects and environment. Therefore, we crop the thermal and visible images before the registration process such that same FOV are used for registration. After we set almost same FOV for both thermal and visible images, we need to find reference points for registration. We implemented a small Matlab code that provides us to choose point from images by mouse-click event. Once we obtain the reference point for homography, we can register visible band images to thermal images to use for our proposed approach. Registration result is shown in Figure 18.



(a) Visible Image Before Registration

(b) Thermal Image Before Registration

(c) Visible Image After Registration

(d) Thermal Image After Registration

Figure 18: Image Registration Result

We captured test videos by using camera setup shown in Figure 19.

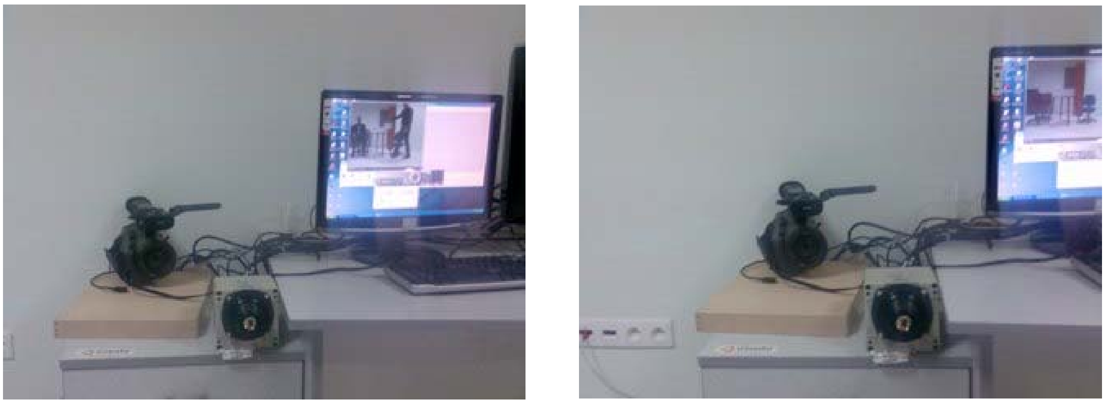


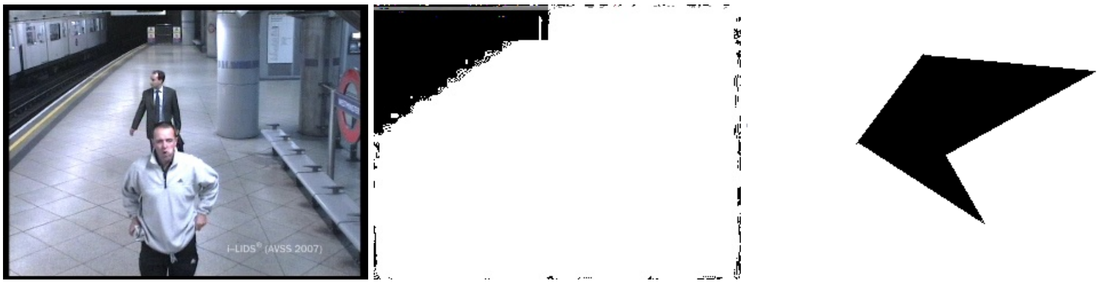
Figure 19: Camera setup for capturing video

The algorithms have been implemented using C++ and Matlab languages and *OpenCV* library [24-25]. OpenCV implemented in C is a computer vision library for originally developed by Intel. OpenCV has more than 500 algorithms, documentation and sample code for computer vision. Also, OpenCV is a platform-independent library that can be used in many Operating System (OS) such as Microsoft Windows, Apple Mac OS X, Linux [25]. Object Segmentation part of the proposed algorithm has been implemented by using Matlab since Matlab provide us very powerful utilities for morphological operations. After Matlab implementation, we ported Matlab code to C++ by generating a Matlab *DLL* file using **mex** utility that Matlab provides. After creating object segmentation DLL file, we have used it in our C++ applications by loading this library.

In our implementations, we needed to adjust some settings regarding performance since object segmentation is really time consuming process. To speed up the algorithms, the size of the images which are captured from the camera is set to 320 x 240. Also we don't make object segmentation and LIO operation on thermal images for every frame. We assume that size of abandoned object is greater than 1% percent of whole image. Therefore, we first calculate the mean of binary image for abandoned object. Then if this mean value is greater than the threshold value that we determine, we run object segmentation and LIO operation for that frame. We choose threshold value 255x0.01 = 2.55 since at least, 1% percent of binary image belongs to abandon object.

5.3 Experimental Results

In this part, we present the test results. Algorithm for abandoned object detection [20] is tested with 6 sample videos that have different scenarios. In some cases, we use the mask that disables the predefined area in image for abandoned object detection. That is, we can choose the detection area by using mask before starting abandon object detection algorithm. This mask can be binary image or any shape that consists of points. Figure 20 shows us the image mask that is used for *AB-Easy, AB-Medium*, and *AB-Hard* included in *i-LIDS* dataset [23]. Only nonblack pixels are in detection area for the mask.



(a) Visible Image (b) Image Mask (c) Shape Mask

Figure 20: Mask Image for Abandoned Object Detection.

If a pixel that belongs to abandoned object is not in detection area, then we ignore the pixel for abandoned object detection. For in *i-LIDS* dataset [23], we know that abandoned object cannot be left in the way that subway train passes and also we eliminate the false alarms that might be raised by the stopping subway trains detected as abandoned objects. Therefore, we set this area as a nondetection area for this data set. Figure 21 shows the abandoned object detection test results for *AB-Easy, AB-Medium*, and *AB-Hard* included in *i-LIDS* dataset [23] at different time.

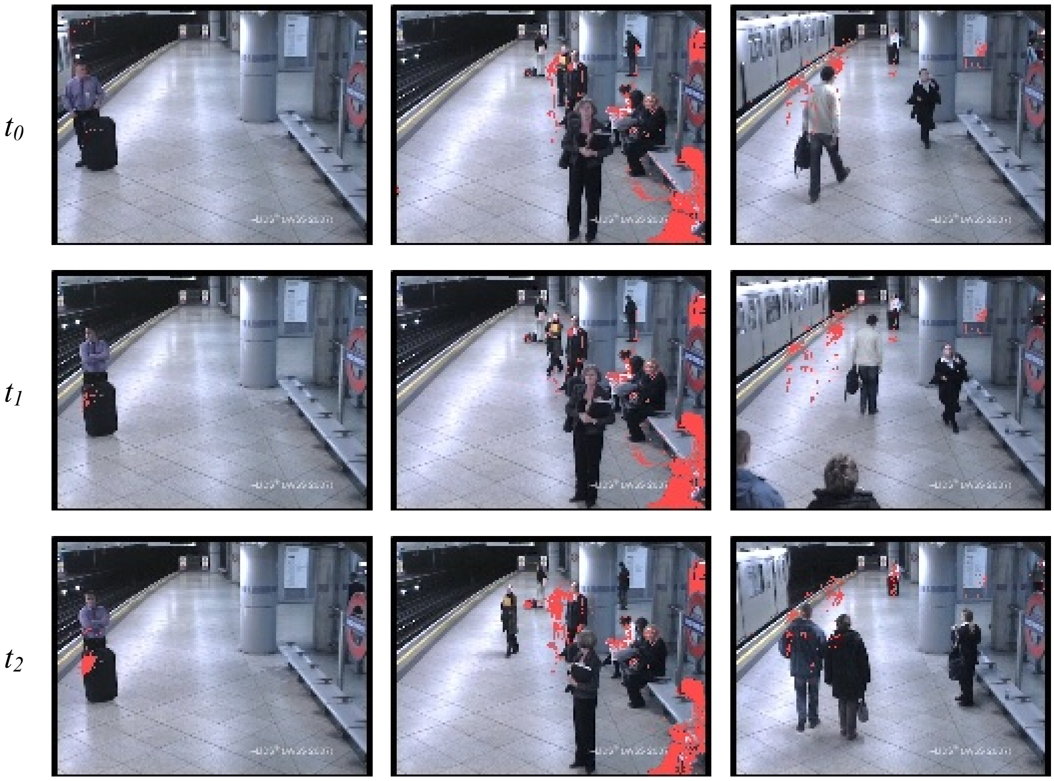


Figure 21 (cont.)

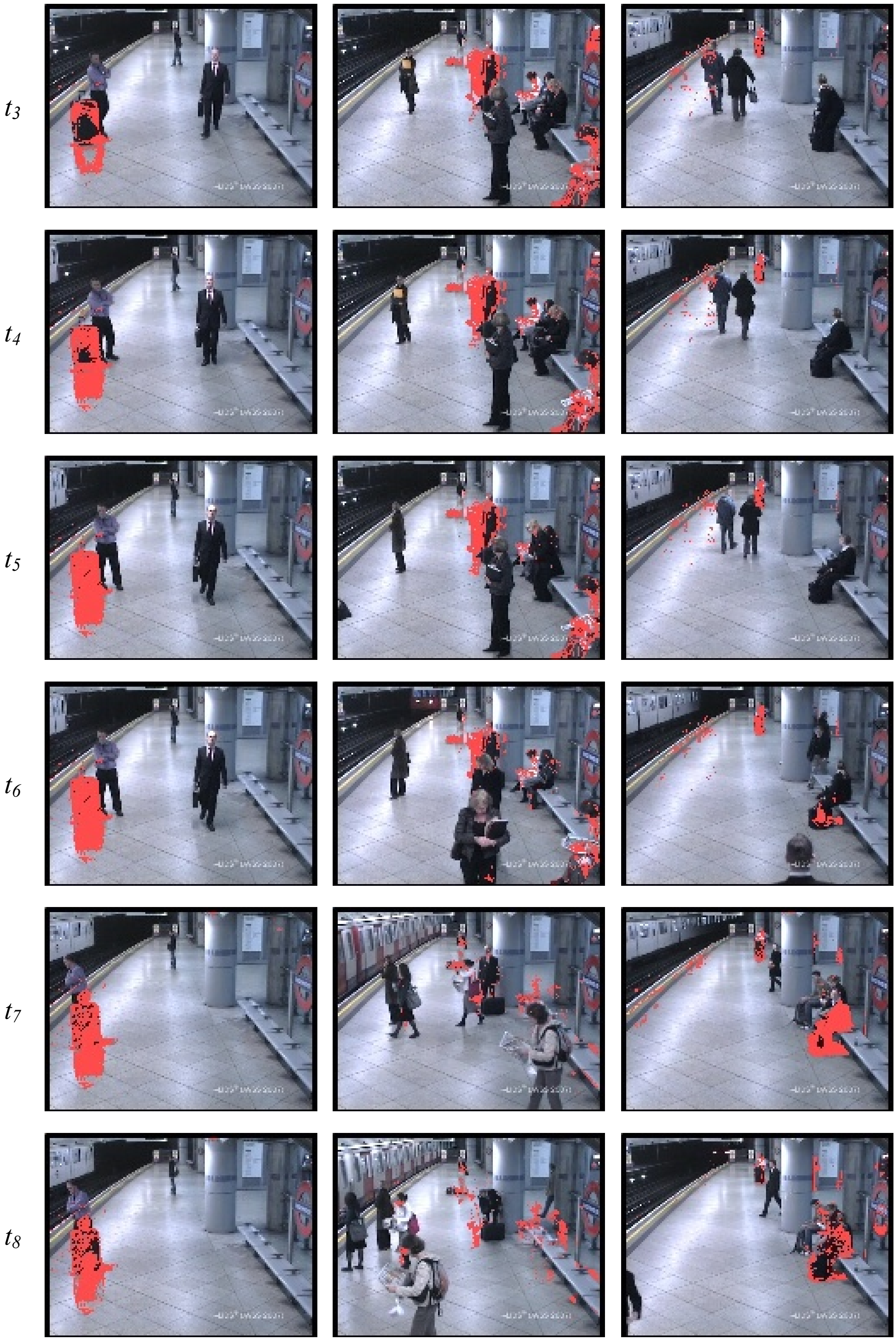
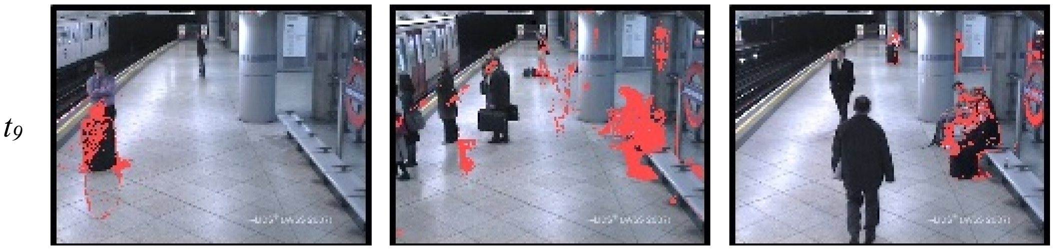


Figure 21 (cont.)



(a) Result for *AB-Easy*

(b) Result for *AB-Medium*

(c) Result for *AB-Hard*

Figure 21: Abandoned Object Detection results for *i-LIDS* dataset [23]

In Figure 21 (a), frame at t_0 is the first frame that object is left. After that time frame, we wait 300 frames to decide whether the pixel belongs to abandoned object or not. Algorithm start to detect pixels that belong to abandoned object at time t_1. At time t_2-t_6, number of pixels that belongs to abandoned object increases and abandoned object is almost masked. After time frame t_6, number of pixels for abandoned object start to decrease since pixels of abandoned object started to mix both long-term and short-term background.

In Figure 21 (b) and (c), luggage is left at time frame t_0. In this case, left luggage is not close camera so we detect it as a small object. Also, there are some noisy in the environment due to the illumination and some person moves after waiting some time. Therefore, we detect some pixels as pixels of abandoned object. At time frame t_6, abandoned luggages are detected but we also get some false detection due to people sitting on the bench or standing without moving much.

Besides to *i-LIDS* dataset [23], we used 2 dataset videos that we captured with different screnerio. Each dataset consists of 3 videos. So, we tested abandoned object detection for 6 more videos and results are shown in Figure 22 and Figure 23.

Figure 22 (cont.)

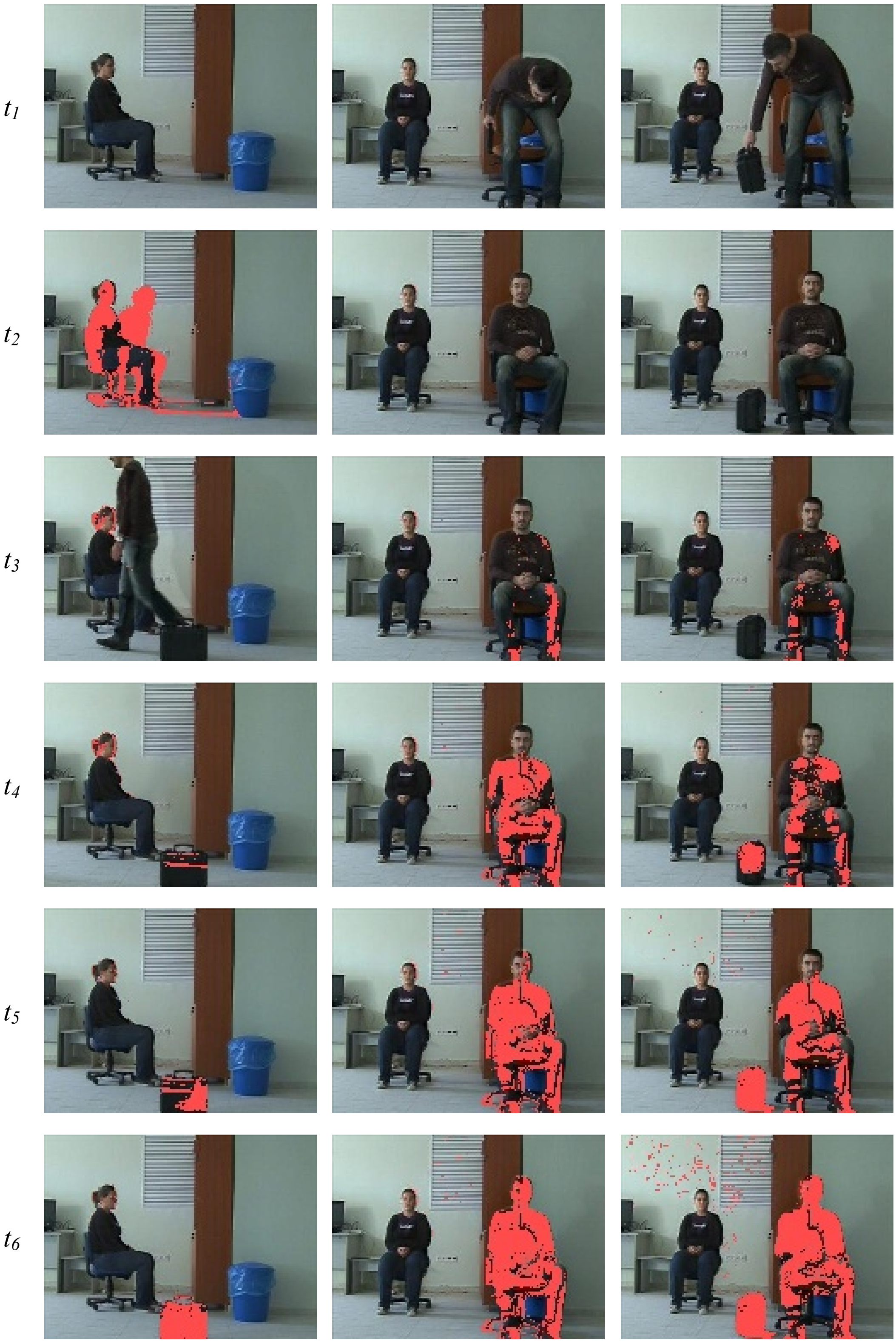
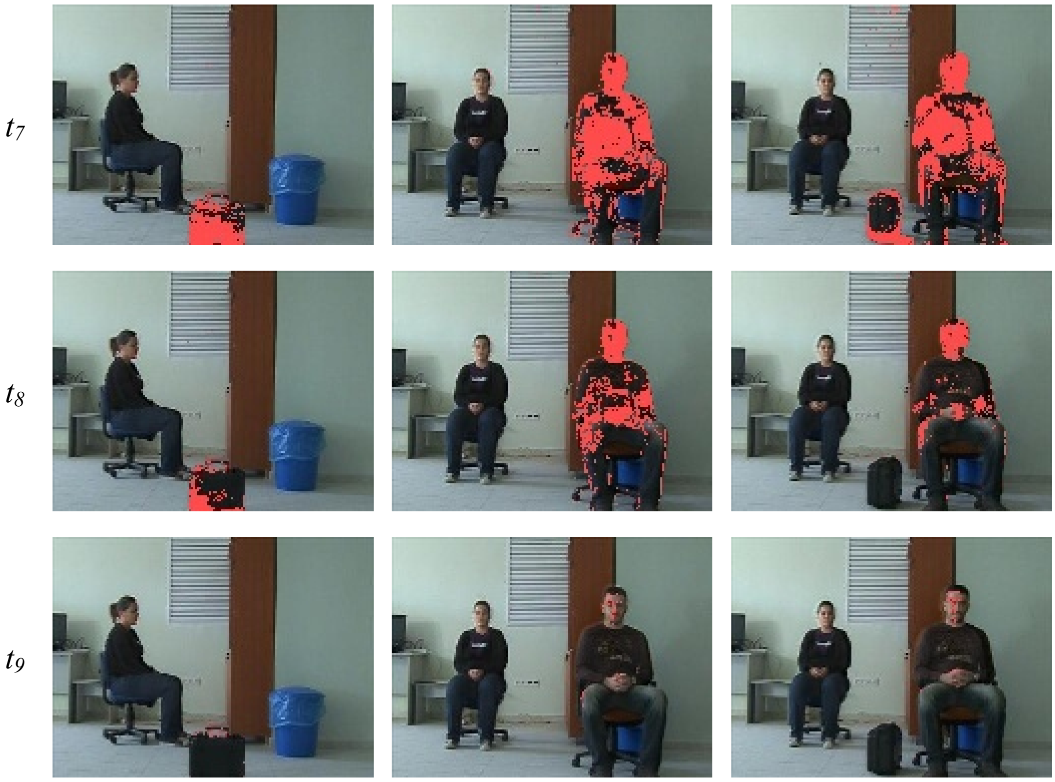


Figure 22 (cont.)



(a) Results for Video1 of Dataset1

(b) Results for Video2 of Dataset1

(c) Results for Video3 of Dataset1

Figure 22: Results of Abandoned Object Detection with Videos that we captured

In Figure 22, we tested abandoned object detection algorithm with different scenarios. We generate a scenario that only luggage is detected as an abandoned object for Video1, only human is detected as an abandoned object for Video2, and both luggage and human are detected as an abandoned object for Video3. In these scenarios, background is displayed at time frame t_0. Algorithm started to detect abandoned object at time frame t_3-t_4. After time frame t_6, detected pixels began to decrease since these pixels became background for both long-term and short-time foreground models. For Figure 22 (a), we get false detection since lady in image moved between time frames t_0 and t_1. So, person became foreground for long-term foreground model but background for short-term model at time t_1.

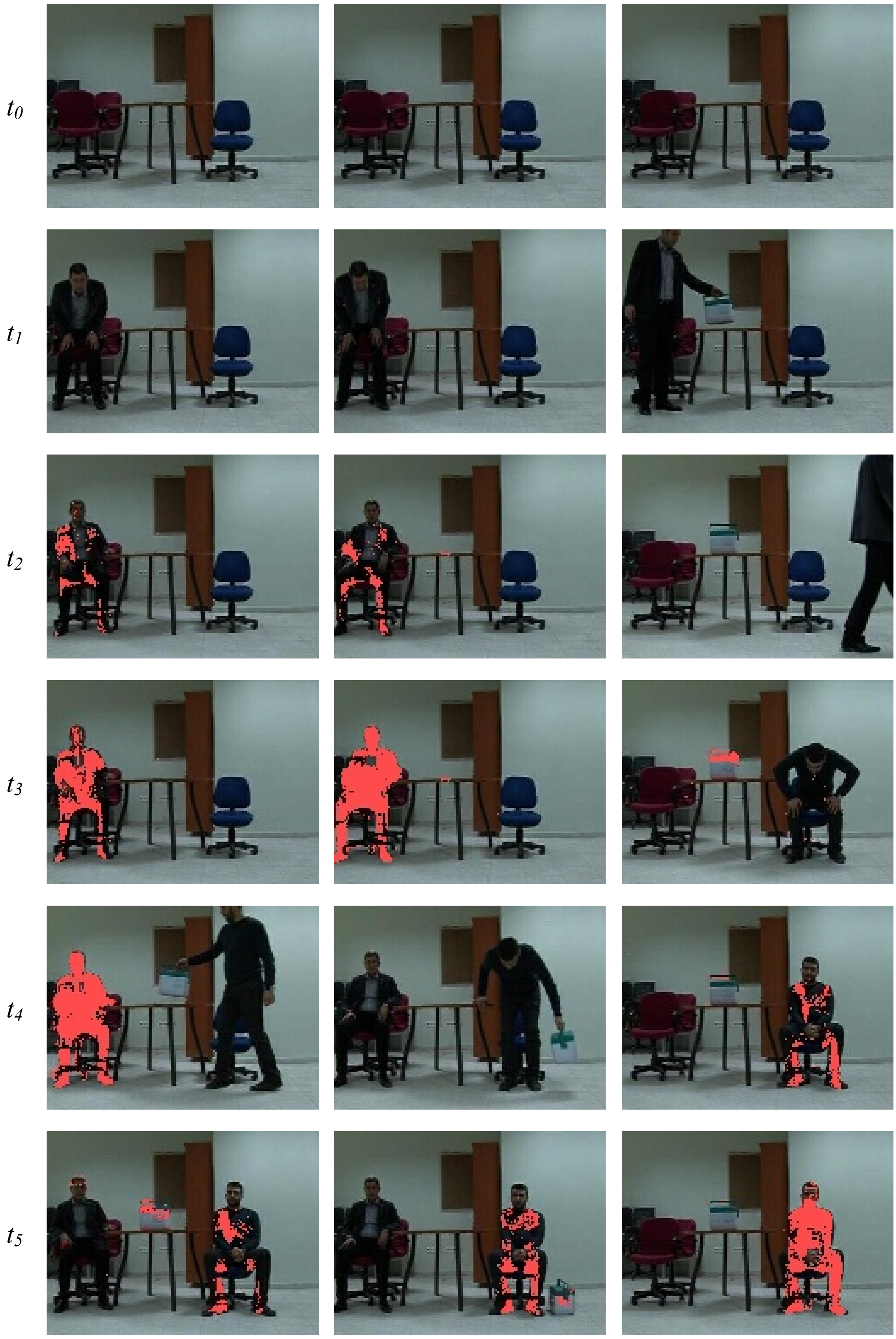
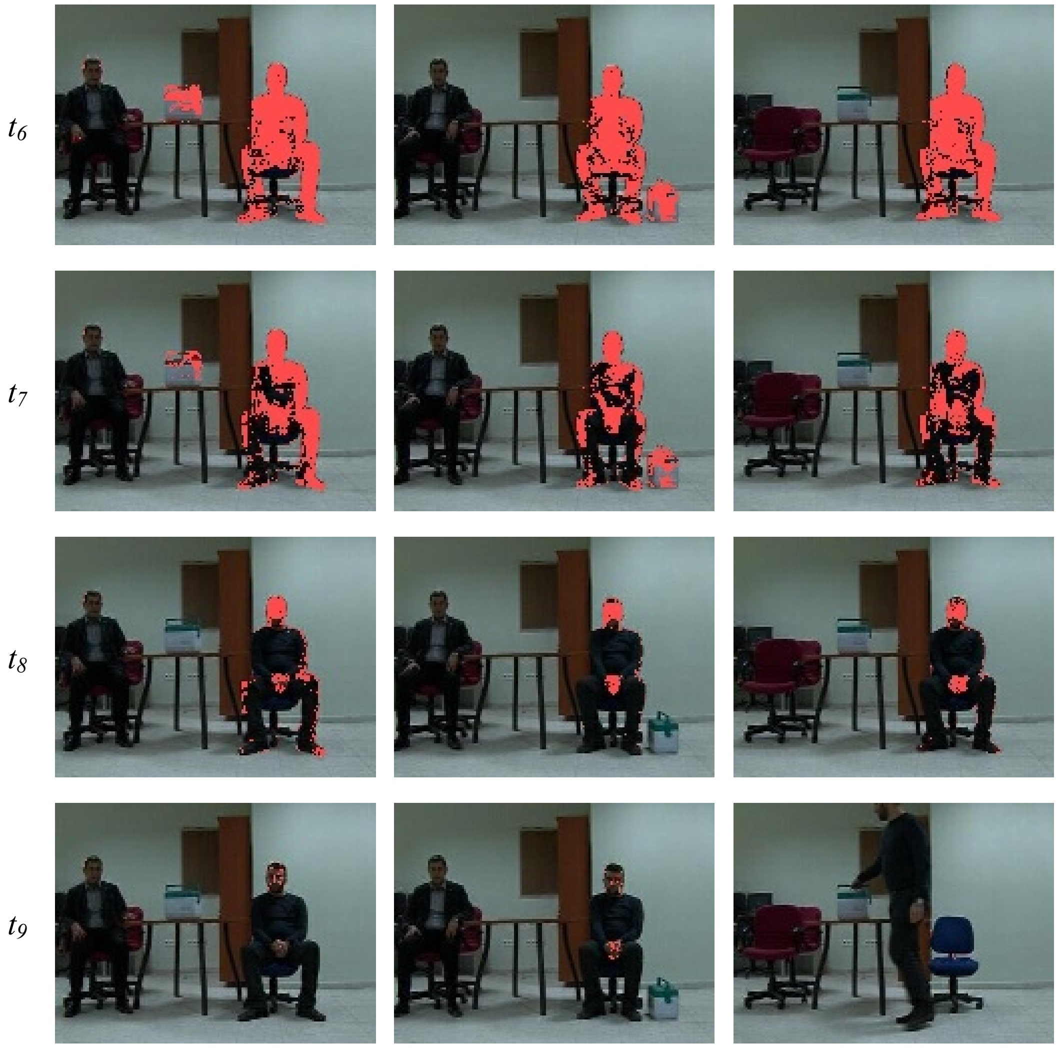

t_0
t_1
t_2
t_3
t_4
t_5

Figure 23 (cont.)



(a) Result for Video1 of Dataset2

(b) Result for Video2 of Dataset2

(c) Result for Video3 of Dataset2

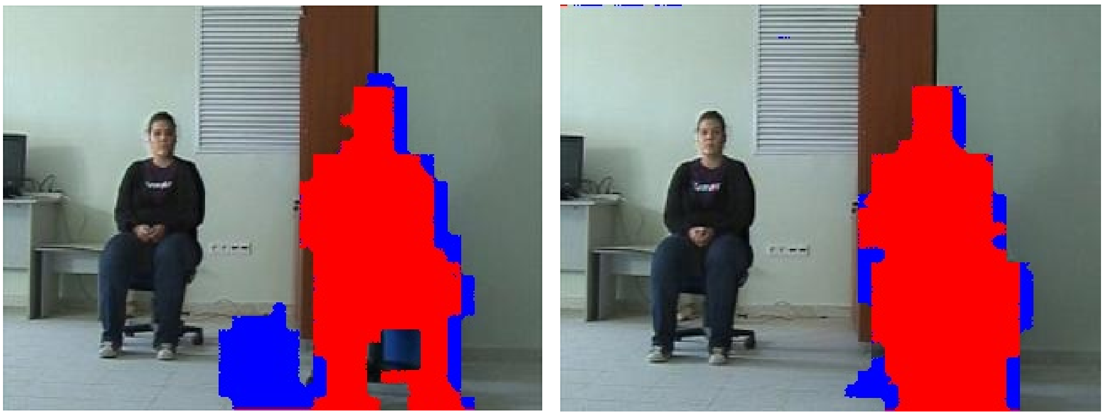
Figure 23: Results of Abandoned Object Detection with Videos of Dataset2 that we captured

In Figure 23 (a) and (b), human came to environment at time frame t_1. Then, algorithm detected human as an abandoned object between t_2-t_4. At time frame t_4, new person with his bag came to environment. Algorithm detected this guy and his bag as an abandoned object between time frame t_5-t_6. After time frame t_6, number of pixels that belongs to foreground started to decrease. In Figure 23 (c), bag was left in the environment at time frame t_1. At time frame t_3, new person came to environment and bag was detected as an abandoned object. Human was detected as an abandoned object between time frame t_4-t_6. After time frame t_6, number of pixels that belongs to foreground started to decrease since these pixels were background at both long-term and short-term foreground models. As you notice, only green part of bag was detected as an abandoned object in Figure 23 (a) and (c) since color for other parts of bag is very close to background color. Therefore, algorithm could not detect other parts of bag in Figure 23 (a) and (c). However, almost all pixels of bag were detected as an abandoned object in Figure 23 (b) since bag was on the ground and colors of bag was not close to background in Figure 23 (b).

After evaluating abandoned object detection, we can discuss the whole proposed algorithm. We tested whole algorithm with 6 videos that we capture. During the implementation, we faced the error explained in section 4.5 around person detected as abandoned object. This error can be measured as follows:

$$E_i = N_w \big/ \left(N_p + N_w\right) \tag{5.1}$$

Where E_i is the error percentage at ith frame, N_w is the number of pixel that belongs to abandoned object but detected with false aliveness features, and N_p is the number of pixel that belongs to abandoned object detected with true aliveness features. Figure 24 shows these errors.



(a) Discrimination Error from Video 3 of Dataset1

(b) Discrimination Error from Video 2 of Dataset1

Figure 24: Object Discrimination Result for person with error

In the sample shown in Figure 24 (a), blue and red colored pixels represents nonliving and living objects respectively. We measured that there are 1690 pixels that belong to person but detected as nonliving and 10862 pixels that belong to person but detected as living in Figure 24 (a). The ratio of area detected as nonliving to living is E = 1690/ (10862 + 1690) = 0.13, 13%. We removed these pixels by using the algorithm explained section 4.5.

Figure 25 and 26 show the results of proposed approach.

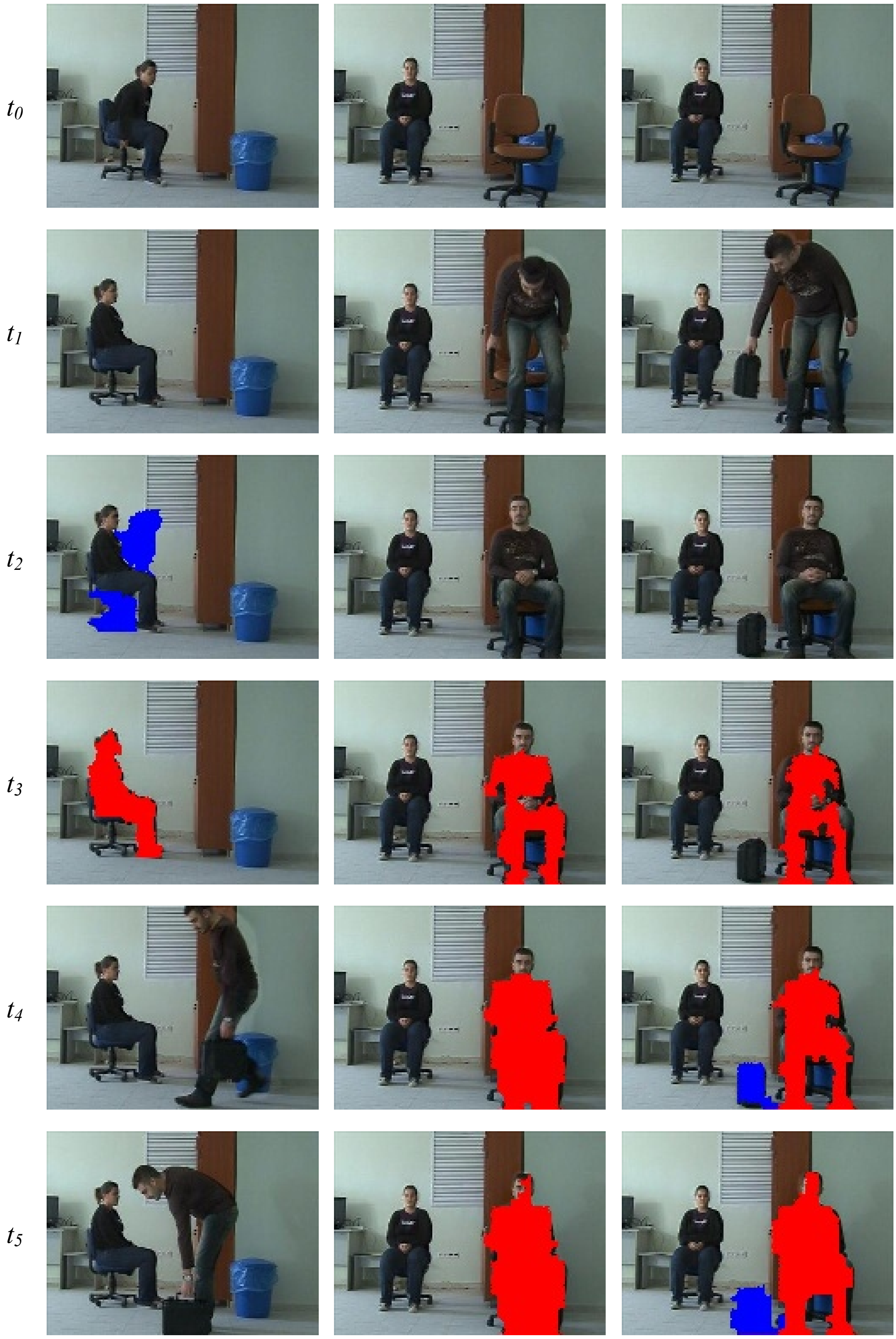
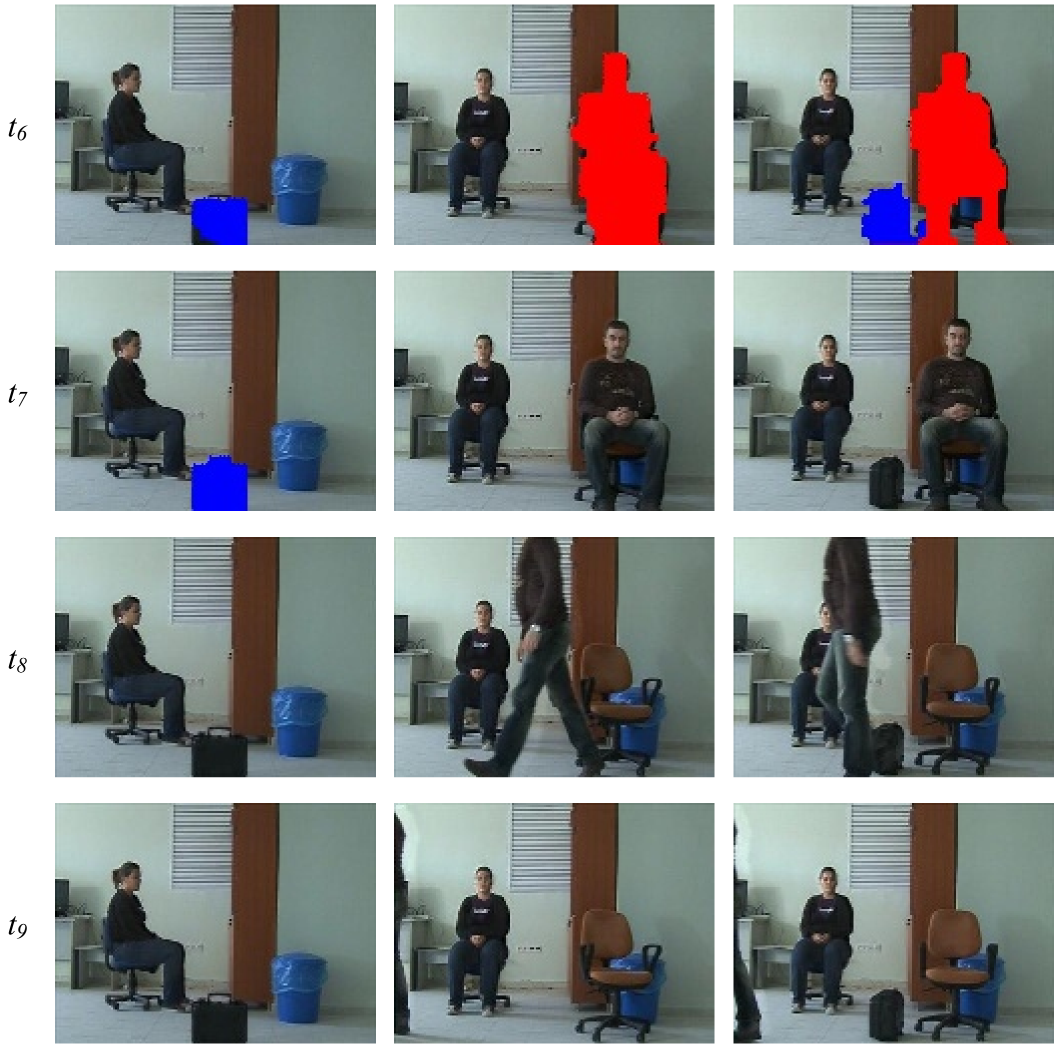


Figure 25 (cont.)



(a) Results for Video1 of Dataset1

(b) Results for Video2 of Dataset1

(c) Results for Video3 of Dataset1

Figure 25: Result of Proposed Solution with Videos that we capture

As shown in Figure 25 (a), human moves around chair between frame time t_0 and t_1. Therefore, at frame time t_2, pixels belongs to foreground at time frame t_0 was detected to nonliving abandoned object. However, there is no such a foreground object showed at time frame t_2. This false detection was generated in only 1 frame due to our thresholding mechanism explained in section 4.4. This false detection can be removed by providing object persistency in visible domain. To provide object persistency, detected object should be observed for *N* adjacent frame. Algorithm detected human as an abandoned living object at frame time t_3 due to movement between frame times t_0 and t_1. Luggage was detected as an abandoned nonliving object at frame time t_6 and t_7. Then, this luggage was mixed to background for both long-term and short-term foreground model. In Figure 25 (b), human was detected as a living object between time frame t_3 and t_6. In Figure 25 (c) human was detected as a living abandoned object but luggage was detected as a nonliving abandoned object. Figure 25 (c) also shows discrimination of abandoned object according to object's aliveness since both luggage and human was detected as abandoned objects at the same time frames but luggage is detected as an nonliving and human was detected as a living object in Figure 25 (c).

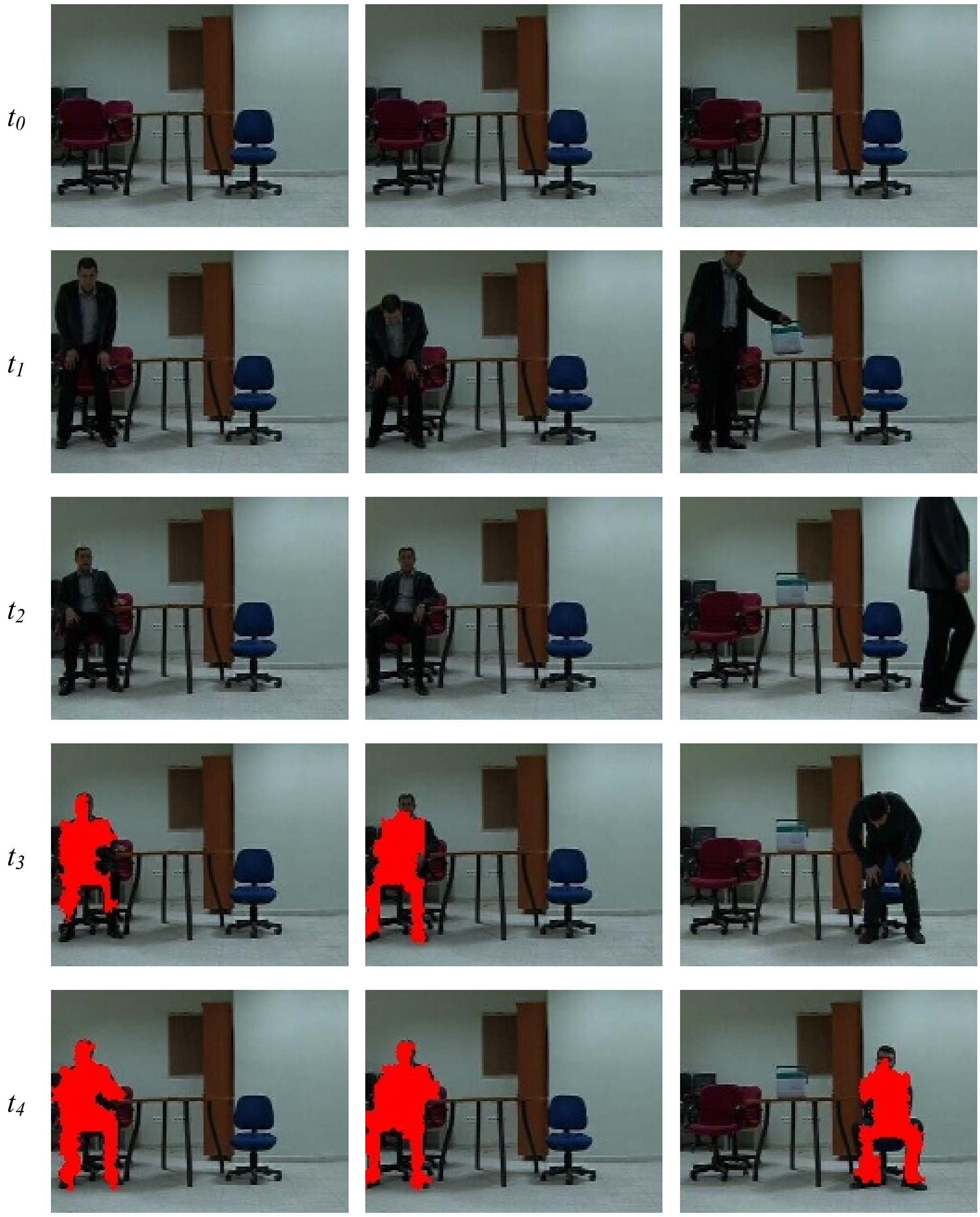
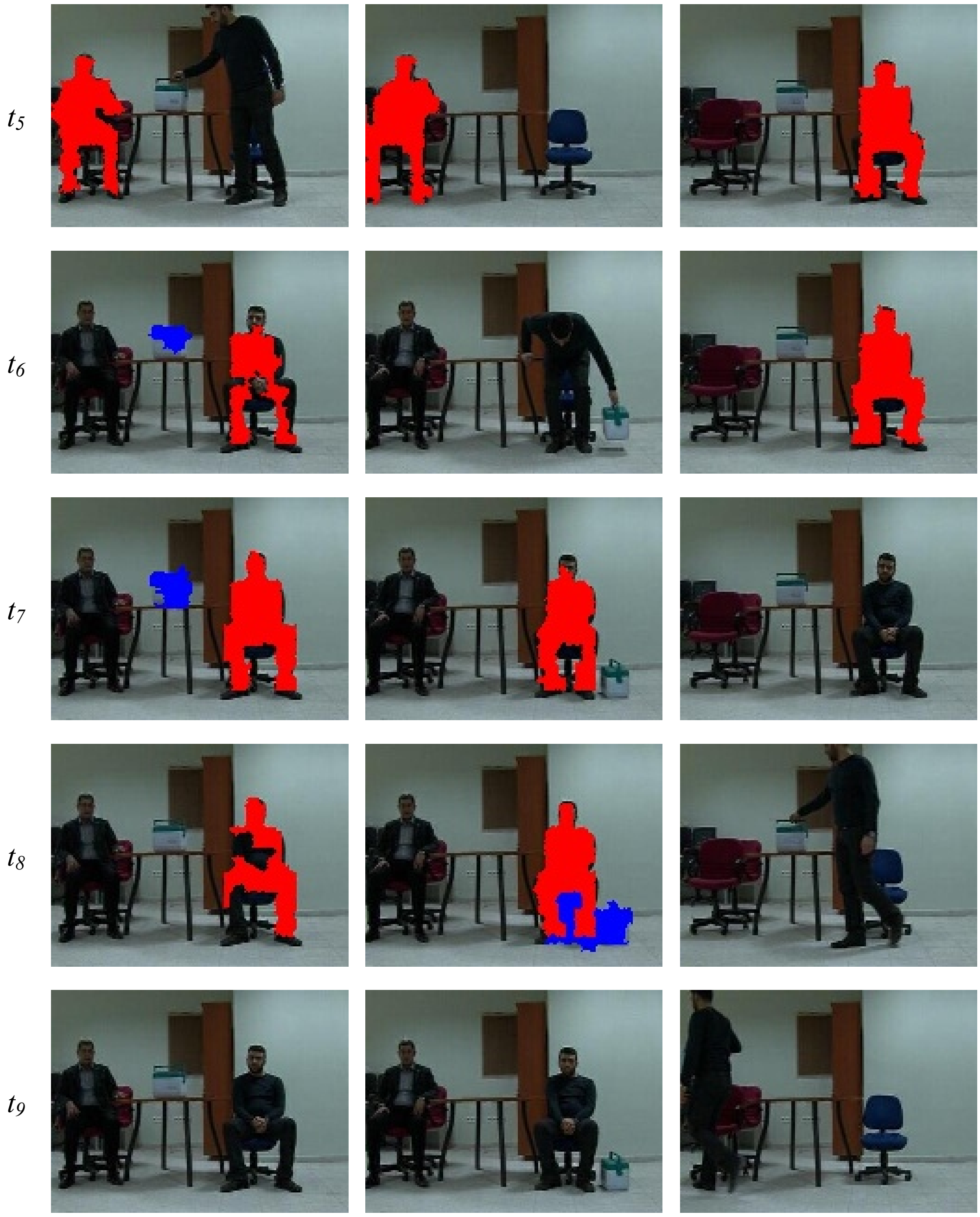


Figure 26 (cont.)



(a) Results for Video1 of Dataset2

(b) Results for Video2 of Dataset2

(c) Results for Video3 of Dataset2

Figure 26: Result of Proposed Solution with Videos of Dataset2 that we capture

In Figure 26 (a), proposed algorithm started to detect human as an abandoned living object at time frame t_3. At time frame t_5, new person came to environment with his bag. Person in right side of image was detected as an abandoned living object between time frame t_6-t_8. Bag that belongs to person detected as an abandoned nonliving object. In Figure 26 (b), proposed algorithm detected human as an abandoned living object between time frame t_3-t_5. At time frame t_6, new person came to environment with his bag. Algorithm detected both person and his bag at time frame t_8. As you notice, bag was detected as an abandoned nonliving object since almost all pixels belongs to bag was detected as an abandoned object in Figure 26 (b). Also, extra region between two legs of human was detected as an abandoned nonliving object. Normally, our algorithm removes this noisy part by using the method explained in section 4.5. However, this region and bag were detected as the only one object. Therefore, equation 4.6 was not satisfied for this case and this region was detected as a part of abandoned nonliving object. In Figure 26 (c), bag was left in the environment at time frame t_1. However, it was not detected as an abandoned nonliving object due to bad illumination and color for some part of bag is very close the color of background. At time frame t_3, new person came to environment and he was detected as an abandoned living object between time frame t_4-t_7.

Our algorithm detected abandoned objects and discriminated them as a living object or non living object. Only abandoned object detection provides to detect temporal objects in the background scene but our algorithm detects abandoned objects with aliveness feature. Therefore, our algorithm can be used more effectively than only abandoned object detection algorithm to find suspicious packets left behind in the environment.

In this thesis, we assume that number of pixels that belong abandoned object is greater that 1% of whole image, 320x240x0.01=768. This provides to eliminate small noisy pixels that can be detected as an abandoned object due to the illumination changes. Therefore, we can find only real abandoned objects in the background scene. However, this is not obstacle for detecting small objects. If we want to detect small objects, we first need to adjust parameters. In the proposed solution, we discard the detected object if number of pixel for object is less than 1000 and density of object calculated in equation 4.4 is less than 0.4. To detect and discriminate small objects, we first need to adjust these parameters. Then, object segmentation algorithm can be improved for small objects' segmentation. Also, we may need better thermal camera that shows small object' temperature very well.

The last thing we can discuss is performance of system. This system is not designed for the real-time applications. Therefore, we load all frames into memory to reduce the run-time of operations. Otherwise, we need to access memory for every frame with file operations and it is really time consuming operation. Therefore, Memory requirement for this system is high.

However, most parts of approach can be used in real-time applications. Because, operation time for background subtraction, abandoned object detection and data fusion is really short. But segmentation operation is really time-consuming. It should be faster for real-time applications. Segmentation is implemented in Matlab but Matlab use C++ DLL files for functions we used for segmentation. Also, we implemented object segmentation part in C++ but we used Matlab since it gives better result in shorter time. To improve segmentation, more heuristic algorithms can be used. Table 2 shows us performance results for proposed system.

Table 2: Performance Results of Proposed Algorithms

Algorithm	**Number of Frame Used for testing**	**Avarage Run time (sec)**
Background Subtraction	1500	0.40
Abandoned Object Detection with dual foregrounds	1500	0.16
Object Segmentation	251	60.18
Data Fusion	251	0.63

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this thesis, we propose an algorithm to detect abandoned object with aliveness discrimination. Our algorithm is based on four parts: Background subtraction, Abandoned Object Detection by using dual foreground approach, living object extraction from thermal image, and Feature level fusion between thermal and visible images. We used Improved Gaussian method [9] for background subtraction. Then, we create two background models by using this background subtraction method: long-term and short-term. We used these two background models for abandoned object detection by using dual foreground approach [20]. Algorithm for abandoned object detection is based on the idea that abandoned objects are the temporal static objects in the background. This algorithm discriminates abandoned object from background by using dual foregrounds that have different learning rate. To extract living object from thermal images, we used LOI method that brightens brighter pixels and darkens darker pixels. After obtaining different feature from visible and thermal domain, we fused this information to get more detailed information about abandoned object. We showed that the false alarms due to people sitting or standing without moving motion could be eliminated with the proposed algorithm.

Test of proposed method is really challenging for us since we faced image registration problems. Since FOV of cameras are different, we made a crop operation on images to set same FOV for both thermal and visible images. We used *i-LIDS dataset* [23] and videos that we captured with different scenarios to test abandoned object detection part of proposed algorithm. Videos that we captured were also used to test whole proposed algorithm. Our results show that the proposed method has low false detection in most of cases. This proposed method is expected to be a base for further research.

In the future, the image segmentation part can be improved since proposed algorithm is used for real-time application. However, this is not enough to use this system in the real-time applications. Thermal camera is too expensive so we cannot use lots of thermal camera to secure crowded environment such shopping malls and airports for real-time applications. Therefore, for now, thermal camera can be placed to capture wide area of the environment and this will reduce the number of thermal camera used for this purpose. Thermal camera is mostly used in military applications. In the future, thermal imagery technologies will be widely using in surveillance applications and this will lower the price of thermal cameras. Also, tracking utility can be added to system to detect who left abandoned object accurately. By using the heat signatures obtained from thermal domain could be used as a feature for tracking the people and the objects they carry separately to deduct further information. Automatic image registration can be added to system if we adjust both thermal and visible camera capture video from same source. This will give us more accurate results.

These algorithms may be adapted to work in different platforms in the future. For example, algorithms can found nonliving abandoned object and send alert to operator’s hand-held palms with wireless network. This system may also be integrated into existing security systems.

REFERENCES

[1] Austin Richards, “Thermal imaging: how far can you see with it”, Flir Technical Note [Online] [Cited: 15 Jan 2010]
www.flir.com/uploadedFiles/ENG_01_howfar.pdf

[2] Piccardi, Massimo., "Background Subtraction Techniques: a review." s.l. : IEEE International Conference on Systems, Man and Cybernetics, 2004. 0-7803-8566.

[3] Wren, C., Azarhayejani, A., Darrell, T. A.P., "Pfinder: real-time tracking of the human." s.l. : IEEE Trans. on Patfern Anal. and Machine Intell. vol. 19, no. 7, pp. 78g785, 1997.

[4] Stauffer, C. and Grimson, W.E.L., "Adaptive background mixture models for real-time tracking." s.l. : IEEE CVPR 1999, pp. 24&252, June 1999

[5] Elgammal, A., Harwood, D. and Davis, L.S., "Non parametric model for background subtraction." June 2000. Proc. ECCV 2000. pp. 751-767.

[6] Han, Bohyung, Comaniciu, Dorin and Davis, Larry., "Sequential Kernel Density Approximation Through Mode Propagation:applications to background modelling." January 2004. Proc. Asian Conf. on Computer Vision.

[7] Seki, Makito, et al., "Background Subtraction based on Cooccurrence of Image Variations." 2003. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’03), 1063-6919/03. pp. 65-72.

[8] Xu, Zhifei, Shi, Pengfei and Yu-Hua Gu, Irene., "An Eigenbackground Subtraction Method Using Recursive Error Compensation." [book auth.] Lecture Notes in Computer Science. *Advances in Multimedia Information Processing - PCM 2006.* s.l. : Springer Berlin / Heidelberg, 2006, pp. 779-787.

[9] Zivkovic, Z., "Improved adaptive Gaussian mixture model for background subtraction," *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol.2, no., pp. 28-31 Vol.2, 23-26 Aug. 2004

[10] Davis, J. W. and Sharma, V. 2007, "Background-subtraction using contour-based fusion of thermal and visible imagery', *Comput. Vis. Image Underst.* 106, 2-3 (May. 2007), 162-182

[11] Davis, J.W.; Sharma, V., "Robust Background-Subtraction for Person Detection in Thermal Imagery," *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on* , vol., no., pp. 128-128, 27-02 June 2004

[12] J. Davis, V. Sharma, "Background-subtraction in thermal imagery using contour saliency", International Journal of Computer Vision 71 (2) (2007) 161–181.

[13] S. Russell, P. Norvig (Eds.), Artificial Intelligence: A Modern Approach, Prentice Hall, 2003.

[14] Ju Han, Bir Bhanu, "Fusion of color and infrared video for moving human detection", Pattern Recognition, Volume 40, Issue 6, June 2007, Pages 1771-1784, ISSN 0031-3203

[15] Krotosky, S.J.; Trivedi, M.M., "Person Surveillance Using Visual and Infrared Imagery," *Circuits and Systems for Video Technology, IEEE Transactions on* , vol.18, no.8, pp.1096-1105, Aug. 2008

[16] S. J. Krotosky and M. M. Trivedi, "On color, infrared and multimodal stereo approaches to pedestrian detection," *IEEE Trans. Intell. Transport. Syst.*, vol. 8, pp. 619–629, Dec. 2007.

[17] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Comp. Vis. Pattern Recogn.*, vol. 1, pp. 886–893, 2005.

[18] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines." 2001 [Online]. Available: http://www.csie.ntu.edu.tw/ cjlicn/libsvm

[19] Conaire, C.O.; O'Connor, N.E.O.; Cooke, E.; Smeaton, A.F., "Comparison of Fusion Methods for Thermo-Visual Surveillance Tracking," *Information Fusion, 2006 9th International Conference on* , vol., no., pp.1-7, 10-13 July 2006

[20] Fatih Porikli, Yuri Ivanov, and Tetsuji Haga, “Robust Abandoned Object Detection Using Dual Foregrounds,” EURASIP Journal on Advances in Signal Processing, vol. 2008, Article ID 197875, 11 pages, 2008. doi:10.1155/2008/197875

[21] Rudi Heriansyah, S.A.R. Abu-Bakar, Defect detection in thermal image for nondestructive evaluation of petrochemical equipments, NDT & E International, Volume 42, Issue 8, December 2009, Pages 729-740, ISSN 0963-8695

[22] W. Burger, M.J. Burge, “*Principles of Digital Image Processing”*, Undergraduate Topics in Computer Science, DOI 10.1007/978-1-84800-195-4_2, © Springer-Verlag London Limited, 20 09

[23] i-LIDS dataset for AVSS 2007. [Online] [Cited: 19 Jan 2010.] http://www.elec.qmul.ac.uk/staffinfo/andrea/avss2007_d.html.

[24] *OpenCV Documentation and FAQs.* [Online] [Cited: 13 March 2009.] http://opencvlibrary.sourceforge.net/.

[25] Gary, Bradski and Kaehler Adrian., *Learning OpenCV.* s.l. : O'Reilly, September 2008. 978-0-596-51613-0.

APPENDICES

APPENDIX A: SPECIFICATIONS OF SONY HDR-FX1 HDV HANDYCAM

Hardware

- Light/Flash : N/A
- Docking Station : N/A

Audio

- Recording Format : Stereo
- Microphone : Yes (Built-in)
- Dolby® Digital Output : MPEG1 Audio Layer 2-Stereo (HDV), PCM (DV)

Convenience Features

- Multiple Language Display : Yes
- Remote Control : Yes
- Tilting : N/A

Inputs and Outputs

- Analog Audio/Video Output(s) : Yes (Mini Plug)
- Analog Audio/Video Input(s) : Yes (Mini Plug)
- Digital Audio/Video Output(s) : Yes (via i.LINK®)
- Digital Audio/Video Input(s) : Yes (via i.LINK®)
- USB Port(s) : N/A
- Component Video (Y/Pb/Pr) Output(s) : Yes
- HDMI™ Connection Output(s) : N/A

- Headphone Jack : Yes (Stereo Mini)
- Microphone Input : Yes (Stereo Mini)
- i.LINK® Interface : Yes

- LANC Terminal : Yes (Stereo Mini)
- S-Video Input(s) : Yes
- S-Video Output(s) : Yes
- Remote Jack : N/A
- Active Interface Shoe : Yes (Cold)

Video

- Analog-to-Digital Converter : Yes (Signal Convert with DV only)
- Video Recording System : Real-Time HD Codec Engine, HDV/DV Recording
- Video Signal : NTSC color, EIA standards

General

- Imaging Device : Three - 1/3" 16:9 Advanced HAD™ CCD
- Pixel Gross : 1120K
- Video Actual : 1070K Pixels
- Video Resolution : Full HD 1440x1080
- Still Actual : N/A
- Recording Media : MiniDV Cassette (sold separately)
- Recording and Playback Times: HDV: up to 60 min. (with DVM60 cassette), DV: SP: up to 60 min., LP: up to 90 min.

Service and Warranty Information

- Limited Warranty Term : 1 Year Parts; 90 Days Labor

Optics/Lens

- Lens Type : Carl Zeiss® Vario-Sonnar® T
- 35mm Equivalent : 32.5-390mm (Camera Mode), 40-480mm (4:3 TV Mode)
- Aperture : f1.6-f2.8
- Digital Zoom : N/A
- Filter Diameter : 72mm
- Focal Distance : 4.5-54.0mm
- Focus : Full Range Auto, Manual (Ring), One Touch
- Progressive Shutter Mode : N/A

- Shutter Speed : 1/4-1/10,000 (AE Mode)
- Minimum Illumination : 3 Lux
- Optical Zoom : 12X
- Exposure : N/A
- Resolution : N/A

Software

- Supplied Software : N/A
- Operating System Compatibility : N/A

Weights and Measurements

- Weight (Approx.) : 4 lbs 6.5 oz (2000g) main unit only; 4 lbs 10 1/8 oz (2.1 kg) including the NP-F570 rechargeable battery pack, DVM60 cassette and lens hood
- Dimensions (Approx.) : 5 15/16" x 7 1/8" x 14 3/8" (151 x 181 x 365mm)

Power

- Battery Type : InfoLITHIUM® with AccuPower™ Meter System (NP-F570)
- Power Requirements : 7.2V (battery pack); 8.4V (AC Adaptor)
- Power Consumption (in Operation) : 7.4W/8.0W/8.4W (VF/LCD/VF LCD)

Convenience

- Image Stabilization : Yes (Optical)
- Low Light Capability : N/A
- Memory Stick® Pro Media Compatibility : N/A
- Still Image Mode(s) : N/A
- Digital Picture Effect(s) : N/A
- Fader Effect(s) : Black, White
- Picture Effect(s) : CineFrame™ Recording and Shot Transition
- Movie Mode(s) : MPEG2
- Scene Mode(s) : Picture Profile (upto 6 pre-set conditions)
- USB Streaming : N/A
- PictBridge Compatible : N/A
- Easy Operation : Assignable Buttons, End Search

- Slide Show Mode : N/A
- White Balance : Auto, A/B Preset, One-Push

Display

- LCD Screen : 3.5" (250K Pixels Wide Precision Hybrid SwivelScreen™ LCD Display)
- Viewfinder : Wide (16:9), Color (252K Pixels)

APPENDIX B: FEATURES OF OPGAL EYE-R640

Features	Description
Focal Plane Array	microbolometer
Spectral Range	8-14 µm
Number of Pixels	640x480
Frame Rate	50/60 Hz Max
Pixel Size	25 µm
Analog Video Output	CCIR or RS-170 (PAL or NTSC)
Digital Video Output	USB2 or LVDS
Remote Control	RS 232 or RS 422 or USB2 (optional)
Control Operation	Video Polarity, NUC, Zoom, Freeze
Operating Voltage	7-9 VDC 12VDC (optional)
Power Consumption	≤ 3.5 W (basic unit)
NETD	≤ 70° mK @ f/1 lens
Operating Temperature Range	-30°C to +60°C
Environmental Qulification	MIL STD 810E IP 65 , IP67 (optional)
Dimensions (without lens)	122x122x100 / (HxWxL) 100x100x80 (optional)
Weights	< 1700 g

Optics focal length (mm)	**Field of View (HxV) (Deg)**
12	67.4 x 50.5
20	43.6 x 33.0
35	26.0 x 19.6
50	18.2 x 13.8
75	12.2 x 9.2
100	9.2 x 6.8
150	6.2 x 4.6