INTEGRATION OF THERMAL AND VISIBLE IMAGERY FOR ROBUST FOREGROUND DETECTION IN TELE-IMMERSIVE SPACES

BY

MILES J. JOHNSON

B.S., Cornell University, 2002

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Aerospace Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2007

Urbana, Illinois

Adviser:

Adjunct Professor Peter Bajcsy
Assistant Professor Timothy Bretl

Abstract

The creation of tele-immersive environments involves taking parts of the real world, or scene, and synthesizing them in a virtual world. These parts are called *objects of interest*, and the collection of these objects together form the *foreground* of the scene. In particular, tele-immersive systems are designed to facilitate communication and collaboration between people. Therefore, the central objects of interest in a tele-immersive system are these people, the things they jointly manipulate, and the tools they need to perform this manipulation. This thesis develops a multi-modal image fusion framework to detect these objects of interest. The framework consists of blob extraction, depth estimation, and coordinate transformations to integrate visible and thermal IR imaging modalities, and results in multi-modal foreground object detection. Experimental results with a prototype tele-immersive system show that fusion of visible and thermal imagery enables robust foreground detection in unstructured environments. The contribution of this work lies in designing the integration framework, prototyping hardware and software components, and evaluating quantitatively the multi-modal foreground detection for a set of standard scenarios.

Acknowledgements

This thesis would not have been possible without the guidance and support of my advisors Dr. Peter Bajcsy (NCSA) and Prof. Tim Bretl (Aerospace Engineering). Their combined advice shaped this thesis and brought focus to my work.

I would also like to thank the TEEVE teams in the UIUC CS Department and UC Berkeley. In particular, I would like to thank Prof. Klara Nahrstedt, Wanmin Wu, Zhenyu Yang, Muyuan Wang, Hossein Mobahi and Dongyun Jin for their tremendous support.

Table of Contents

List of Figures

Chapter 1

Introduction

The invention of the telephone in 1876 suddenly allowed two people, even separated by a vast distance, to talk with each other as if they were in the same room. Video conferencing systems in the twentieth century further allowed two people to see each other as if through a looking glass. Today, tele-immersive systems promise to let us step into that looking glass, and feel as if we share physical space. In contrast to traditional virtual reality, these systems create virtual environments by rendering *real objects* captured from images of the *real world*.

This chapter presents an overview of tele-immersive systems (Section 1.1) and discusses some of the current challenges to their implementation (Section 1.2). In particular, we focus on the problem of how to decide which objects in a visual camera image should be included in the virtual environment (Section 1.3). Our approach will fuse information from a second type of camera, a thermal infrared camera, which has a number of potential benefits (Section 1.4).

1.1 Overview

Emerging techologies in computer vision and interactive multimedia applications are placing our society on the verge of being able to experience realistic, life-like, virtual environments on a day to day basis. Following the telephone, internet and video conferencing, tele-immersive systems are the next evolution of individual and group remote communication and interaction.

Tele-immmersive systems integrate networked virtual reality (VR), real-time video, and significant computing and processsing resources. Together, these create an immersive virtual environment that provides a foundation for communication, interaction, and analysis. In contrast to traditional VR, tele-immersion places an emphesis on communication between people. Currently, a typical setup involves human subjects in geographically separated locations interacting in a common virtual environment.

Potential application domains for this technology include:

- remote medical diagnosis and therapy

- distributed decision making through integrated immersive interaction (e.g. distributed teams analyzing scientific data)
- social behavior and networking research (e.g. understanding of complex 3D group behavior)
- remote and distributed education (e.g. learning, training, exploration based on 3D environments)
- distributed artistic performance and interaction (e.g. distributed dance choreography)
- entertainment (e.g. immersive interaction with intelligent environments in movies and games)
- virtual travel and tourism (e.g. architectural walk-throughs and historical virtual tours)
- distributed sport activities and training (e.g. interactive golf swing training)

As the above list demonstrates, the development and application of tele-immersive systems involve multi-disciplinary participaction and expertise, ranging from psychophysical research to low-level vision processing algorithms. This is due to the wide spectrum of system components and multiple goals of tele-immersive environments. The end goals of tele-immersion are centered around enhancing the user experience, and enabling new experiences that have not been possible before. In order to meet these goals, there are many low level infrastructure and implementation challenges that need to be addressed, as explained in the next section.

1.2 Challenges in Tele-immersive Systems

There are two general aspects of tele-immersion research. First, tele-immersive systems perform the task of absorbing, or cloning, parts of the real world, making them available in the virtual environment. Second, tele-immersive systems provide the opportunity to create a rich virtual environment, free from the laws and bounds of reality, which enables users to interact with each other, or complex data sets, in ways that are not possible using traditional physical and computer interfaces.

To enable both of these aspects of tele-immersive systems, there are fundamental technology requirements/needs such as [35, 26, 15]:

- Sensor network development: arrays of sensors must be utilized to capture desired aspects of the real world and import them into the virtual environment. This involves camera array calibration, wireless sensor calibration, etc. The main challenge here is how to handle large arrays of heterogeneous imaging sensors.

- User interface: what kind of displays (computer monitors, goggles, etc.) does one use to interface between the user and the virtual environment. Also, what kind of interactive tools are implemented to allow users to interact with and manipulate the environment. This area also involves the development of visualization algorithms.
- Local environment calibration: (this is an extension of camera calibration and user interface) How do you configure and align local environments so that a coherent virtual world is constructed. (Do the users share a virtual space, or do they just view a nearby virtual space? What are the objects of interest – other users or visualizations of data?)
- Access to high performance computing resources: the virtual environment will often benefit from access to computing resources to both generate and filter the data of interest.
- Networking: transmitting massive amounts of data between remote locations, handling heterogeneous network protocols UDP, TCP, etc. What infrastructure should be used to handle message passing and distributed shared memory.

From a software development perspective, there are two approaches to tele-immersion research: application development and core development [26]. Application development focuses on domain-specific tele-immersion tools, while core development focuses on the software and hardware infrastructure necessary for resource access and hardware communication throughout the tele-immersion system. The current system level challenge of tele-immersion technology is to integrate all of the above in a coherent, extensible system that allows developers to write software for domain-specific tele-immersion applications.

Often, research in tele-immersion is driven by specific applications or domains. For example, the OptIPuter project development is driven by the growing demands of Earth science and bioscience data analysis [15]. Currently, the OptIPuter project is focusing on networking infrastructure, and defines their near-term goal: "... to increase visualization power tenfold or more through a new architecture for distributed information infrastructure, one that optimizes the use of screens, clusters, storage and networks in parallel."

Other projects are approaching tele-immersion research in a different way. One approach is based on leased dedicated network connection and access to high performance computing (HPC) resources as implemented between the University of Pennsylvania and the University of North Carolina. Another approach is to limit the spatial content and use custom hardware as prototyped in the Coliseum system [6]. The approach taken by research groups at the National Center for Supercomputing Applications (NCSA), the University of Illinois at Urbana-Champaign, and the University of California - Berkeley is to integrate commercial off-the-shelf components, providing

flexible and expandable spatial content. These projects have their roots in earlier work of stereo-based 3D reconstruction [5, 43, 14, 31, 34, 22]. The work presented in this thesis builds off of the system maintained by the TEEVE research groups at UIUC and UC-Berkeley.

Finally, there is a large body of related research that tackles issues important in tele-immersion research from a human-computer interface point of view. This research, which began with the intention of making existing tools more efficient, is being performed in the areas of computer supported collaborative work (CSCW) and groupware. These bodies of research have a strong focus on human behavior, both individual and social, and how various interfaces and technologies affect human perception and interaction [12].

Due to the complexity of these interdependent challenges, existing systems that provide immersive communication services are expensive and tailored to very specific environments. Our work attempts to make progress towards a *portable, robust, and inexpensive* system that people can use in their daily lives. Such a system has the following attributes:

- portable non-intrusive hardware (this also implies simple and robust multi-sensor calibration)
- robust real-time reconstruction and transmission of real 3-D data using off-the-shelf components
- inexpensive and easy to use system that is affordable by a significant portion of society

Working to make progress towards realizing robust tele-immersive environments can come in three forms: (1) design and utilization of better sensing, (2) working on underlying infrastructure and other low-level networking and vision algorithms (e.g. stereo reconstruction), and (3) working on higher level scene understanding research that attempts to more efficiently and intelligently utilize the hardware and infrastructure that currently exist. This thesis takes the first approach, and focuses on making a tele-immersive system robust to *complex* and *dynamic environments* through the fusion of multi-modal vision information.

1.3 Foreground Detection in Tele-immersive Systems

Portable and robust tele-immersive systems will be required to handle diverse and loosely controlled environments, such as homes, offices, conference rooms, etc. One problem such environments pose is a difficulty in extracting the useful, or interesting,

portion of the real environment to be synthesized, or reconstructed, in the virtual space. This problem of *foreground detection* is common in many computer vision tasks. However, in immersive environments, a user's perception of the environment is a critical part of their experience. Thus, the performance of a foreground detection algorithm contributes significantly to the overall performance of a tele-immersive system.

As described above, tele-immersion emphasizes the importance of a user's perception of the information of interest. Further, tele-immersion involves a transformation of real world objects into a virtual space (possibly containing purely virtual objects). Viewed in this context, foreground object extraction and tracking has an additional benefit. Successful foreground extraction not only acts as a pre-processing filter for more complicated computations but also allows one to *control* what elements of a scene are visualized, or incorporated, into a tele-immersive environment.

Separating foreground from background in images is a common task in many computer vision and image processing applications. In real-time vision applications that utilize stereoscopic camera systems, accurate foreground detection significantly increases the performance of the system. When dealing with stereoscopic imaging systems, foreground detection plays a major role as a pre-processing step [8], [32]. Further, in the area of real-time object detection and tracking, foreground object extraction plays a significant role in reducing processing time. For example, in moving object detection, where processing-heavy algorithms are used to estimate the object's position and speed, one would like to focus on interesting portions of the image, rather than heavily processing parts of the image that will later be discarded.

Our motivation for integrating thermal cameras with our teleimmersive system began when we were looking closely at components of the system in order to optimize performance. We observed that having a large, and poorly modeled, background significantly slowed down system operation (by up to a factor of 2). This is mainly due to the fact that stereo correlation is being attempted on a larger portion of the field of view. In other words, the common objects of interest in tele-immersive systems typically represent a small fraction of the entire field of view.

In addition to decreasing computation by allowing processor heavy operations to focus only on interesting regions within the scene, keeping track of coherent foreground regions can enable prediction of future object poses and motions. This would also decrease computational requirements by allowing more intelligent selection of disparity windows, resulting in more efficient 3D modeling of foreground objects. These advantages of foreground object detection reflect an interesting theme: the more one knows about the types of objects in a scene, the less work has to be performed to understand the scene's structure or meaning.

Separating an interesting foreground object from a complex scene remains quite a challenge. Many systems have been devised to solve this problem, from the ubiquitous

method of background subtraction, to complex feature-based methods utilizing, for example, multi-resolution wavelet filters. Because of the technology available, and because of the similarity with our own biological vision, most human-scale computer vision research focuses on systems operating in the visible light wavelengths. However, in industrial and military domains, systems are often encountered which use other related technologies such as synthetic aperture radar, sonar, ultrasonic, near infrared, and thermal infrared. These additional modalities are used to increase the capability of pattern recognition systems. This thesis will utilize an additional *spectral* modality and develop a foreground detection system integrating visible and thermal infrared imagery.

1.4 Thermal Infrared Imaging

Visual and thermal cameras provide fundamentally different information. Where visual cameras primarily measure how materials reflect light, thermal cameras primarily measure temperature. These differences of content mean that a combination of visual and infrared images can provide more information about a scene than either modality used alone.

While conventional digital cameras use CCD (charged coupled device) sensors to measure electromagnetic energy from 0.4 μm to around 1.0 μm wavelengths, multiple technologies are becoming available that measure longer wavelengths from 1 μm to 14 μm [47]. These longer wavelengths typically define what we consider the infrared portion of the spectrum. This range is decomposed into four regions: near infrared (NIR), shortwave infrared (SWIR), midwave infrared (MWIR), and longwave infrared (LWIR). The NIR and SWIR portions of the spectrum are still primarily sensitive to reflective material properties, similar to visible cameras. In the longer wavelength regions, however, energy detected from objects becomes dominated by thermal emission. Thus MWIR (3.0 μm to 5.0 μm) and LWIR (8.0 μm to 14.0 μm) define the thermal infrared spectrum. This thesis will focus on LWIR imaging, thus terms such as infrared (IR) or thermal image will always refer to LWIR image acquisition.

There are three types of benefits that IR imaging can provide tele-immersive systems: (1) IR can enhance image processing tasks at a low level (e.g. human foreground detection), (2) IR can allow tele-immersion users to perceive temperature in the virtual environment using visual or tactile feedback, and (3) IR can fundamentally enhance material and object classification.

IR low-level image processing

Assumptions about objects are explicitly and implicitly used throughout the entire field of computer vision and image processing. An example from visible image filtering:

objects in an image provide the high spatial frequency component while illumination consists of lower spatial frequencies [36]. Another flavor of this is: objects will tend to produce small regions with a large intensity gradient, while other areas will have a relatively small gradient. These assumptions help in object segmentation by allowing one to find edges in the image, and assume that intensity values at edge locations generally reflect the intensity of an object, allowing a threshold to be defined based on the estimated object intensities. Further, it is from basic assumptions like this that many sophisticated vision algoritms (in visible wavelengths) derive their inspiration and motivation.

We will not focus on this heavily, but note that IR imaging provides a fundamentally different view of a scene. Because illumination and visually reflective information generally does not effect IR measurements, we must always keep in mind that the shapes, structures and intensities we see in thermal images are created by different phenomena. For example, IR images do contain textures and gradients, but these are not based on visible illumination, and instead derive from the complicated heat transfer between internal and external heat sources and the skin, clothing, or other materials that are part of a physical object.

It is also interesting to note that it is possible for features in visible and thermal wavelengths to be highly correlated. In human face imaging for example, because of the structure and composition of our faces, regions such as eyes, noses, and mouths are easily recognizable in both visible and thermal wavelengths. A more subtle and loosely correlated example is when thermal texture implies the likelihood of visible texture on the human body. Because of how clothing lies on our bodies, some areas are in more direct contact with our body than others. This leads to uneven heat transfer from our bodies to our clothing. Simultaneously, because our clothing is uneven, it often generates small shadows which often represent a large portion of the texture we see in visible wavelength images.

While IR images are robust to illumination changes, they also present new challenges. When imaging the human face, for example, while one no longer has to consider what lights are on or off, one should consider the surrounding environment (is it summer or winter), physical activity (is the person at rest or jogging), and psychological activity (is the person relaxed or excited). In general, however, IR images allow us to relax many restrictions normally placed on a visible spectrum algorithm that are primarily due to issues of incident illumination on an object. Face detection in low light environments is a good example of this – instead of attempting to adjust a face detection algorithm because of poor illumination, one can rely on IR imagery to perform consistently in nominal or poor illumination.

IR perception in tele-immersive systems

Another benefit of adding thermal imagery to a tele-immersive systen is that it can provide another layer of information to the user, enhancing the immersive experience.

Temperature quite often plays an important role in our awareness and perception of our environment. A few examples of how thermal information can provide a more immersive experience are:

- a mechanical engineer remotely inspecting some machine operation or simulation – direct visual/thermal feedback could lead to more efficient design prototyping and analysis
- a physical trainer remotely observing a patient undergoing physical therapy – visual/thermal feedback can lead to more natural interaction and understanding between doctor and patient.

IR based material classification

Finally, in addition to low-level processing and allowing another mode of perception for the user, thermal imagery can also be a great asset in material classification problems. The emitted thermal energy in the LWIR spectrum has a strong dependence on the bulk properties of a material such as specific heat, heat generation rate, density, volume, etc. Thus IR imaging can be utilized to more directly measure these properties of an object [33]. When combined with visible imagery, this essentially extends the possible feature space during object/material classification. For example, [33] and [30] describe methods to specify a relationship between an object's thermophysical properties (thermal conductivity, thermal capacitance, emissivity, etc), scene parameters (wind temperature, incident solar radiation, etc), and the sensed IR image. They use this relationship to define invariant features that will remain constant for an object throughout variations in the scene parameters, and hence enable robust classification. Further, thermal IR imagery can be used to detect abnormalities in an environment. These abnormalities might represent, for example, deviations in a person's mood [37], or hazards [4].

Chapter 2

System Hardware and Configuration

Required infrastructure for a tele-immersive system includes everything from networking to stereo camera rigs. This chapter presents the camera hardware system used in this thesis (Section 2.1), which is a combination of stereo clusters from the TEEVE project in the CS department at UIUC, and thermal cameras from the Image Spatial Data Analysis group at NCSA. This chapter also provides an overview of the software configuration controlling the operation of these cameras, which invovles image acquisition, synchronization, and software development tools (Section 2.2). Given this set of cameras, knowledge of their relative positions and orientations is critical to the algorithms used to perform 3D reconstruction, and foreground detection. Thus, this chapter concludes with a description of the automatic multi-camera calibration techniques used in this thesis (Section 2.3).

2.1 Hardware Setup

Tele-immersive systems currently depend on camera hardware to provide all of the information that will be available in the virtual environment. Thus, factors such as framerate, resolution, synchronization, and color all fundamentally effect the immersive experience. Further, hardware issues often determine whether or not a vision system is portable and easily re-configurable.

In this thesis, we will consider an environment containing one *trinocular stereo cluster* and one *thermal infrared camera*. The stereo cluster contains three grayscale and one color Dragonfly digital camera from Point Grey Research Inc. The thermal camera is a ThermoVision A10 uncooled microbolometer, which detects thermal energy in the LWIR (7.5 to 13.5 microns) wavelengths. (This camera is produced by FLIR Systems, Inc., and was previously called the Indigo Omega camera.) Figure 2.1 shows two stereo clusters and one thermal infrared camera.

The trinocular stereo cluster is a modular unit, and in full deployment, there could be many of these clusters spatially distributed to provide the desired coverage over a large area. For example, in the UIUC TEEVE lab, ten clusters are used to cover roughly 180 degrees (Figure 2.2). It would be ideal for there to be a thermal camera paired with

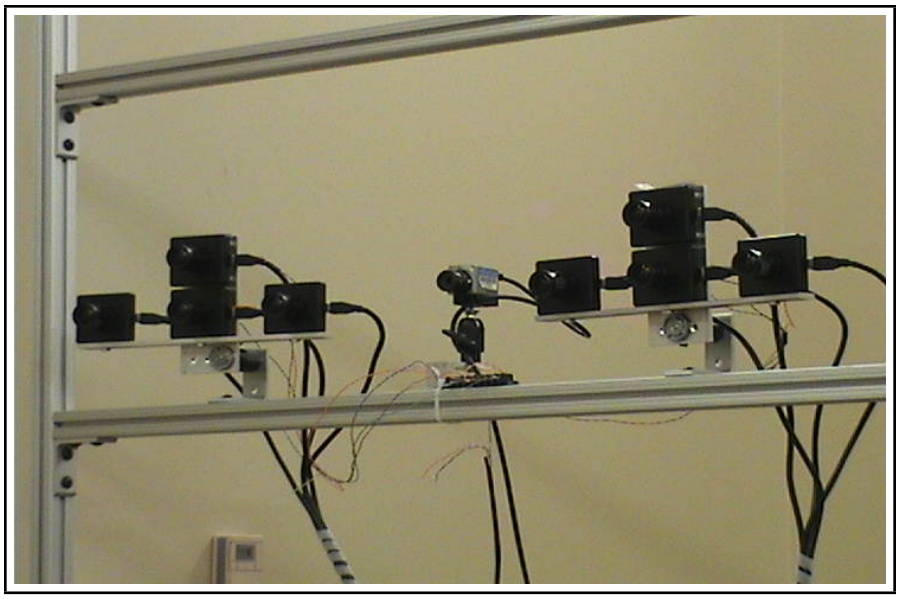


Figure 2.1: Camera Hardware: Two trinocular stereo clusters and one thermal infrared camera in the Collaboration Lab at NCSA. Note that only one stereo cluster was used throughout this thesis.

each stereo cluster – the closer two cameras are, the more similar their perspectives of the scene are. However, due to the current price of thermal cameras ($\sim \$10,000$), this quickly becomes cost-prohibitive. As the technology progresses, however, prices on thermal infrared cameras will drop, and these cameras will move closer to being an affordable off-the-shelf commodity (in comparison, the four Dragonfly cameras which form one cluster are roughly $\$700$ per camera). In the meanwhile, one can consider using a relatively sparse set of thermal infrared cameras to provide support to a denser array of stereo clusters.

2.2 Software Configuration

In our prototype system, the visible (three grayscale and one color) images are captured using the Point Grey Research camera drivers on a multi-core desktop computer. These four cameras are synchronized using an external trigger connected to the GPIO pins on the Dragonfly cameras. The external trigger is created using parallel port communication from a standard desktop computer.

Thermal images are captured using the Libdc1394 Linux/Mac OS X drivers on a laptop (Intel Core 2 Duo). Synchronization between the thermal and visible cameras is performed using timestamping. The timestamps themselves come from the host computer clocks, which are synchronized using the Network Time Protocol (NTP). We found that this leads to synchronization accuracies between visible and thermal host computers that are typically within 100 ms.

Figure 2.2: TEEVE lab at the University of Illinois. Ten clusters are mounted such that they cover roughly 180 degrees.

All capture, processing, and visualization software is written in C++. Development occured under Windows XP using Microsoft Visual Studio 2003 and 2005, as well as under Mac OS X using the Gnu Compiler Collecion (GCC). Many of the algorithms discussed below also utilize functionality from the OpenCV computer vision library (see details in Chapter 4). For details on the general system architecture of the TEEVE project, see [49].

2.3 Camera Calibration

Vision algorithms that perform 3D reconstruction primarily rely on knowledge of the camera positions and orientations with respect to some reference frame. Further, as we will see later (Chapter 4), this knowledge also plays a fundamental role in our proposed foreground detection system. Note that if one cares about absolute power incident on the imaging sensor (in the visible or thermal modalities), then radiometric calibration must also be performed. A demonstration of determing absolute temperature from image intensities in thermal cameras is shown in [4].

Camera calibration is the process of determining the geometric and optical parameters which describe the transformation from an object in the world to it's image detected by the camera system. These parameters are usually grouped into *intrinsic* and *extrinsic* parameters. Intrinsic parameters describe quantities that are effected by the optical and

electrical components of a camera: (a) focal length, (b) pixel aspect ratio, (c) principle point, (d) lens distortion, etc. Extrinsic parameters describe the geometric position and orientation of the camera with respect to the world. In practice, this turns out to be a translation vector which points to the origin of the world frame with respect to the camera; and a rotation matrix, which describes the orientation of the world frame with respect to the camera.

These parameters are derived from a simple model of a camera: the pinhole perspective projection model. In this model, the aperture of the camera is infinitely small, causing all light to pass through one point, the optical center, before hitting the detector. In the ideal pinhole camera model, a 3D point p with coordinates $\mathbf{X} = [X, Y, Z]^T$ in the camera reference frame is related to it's image $\mathbf{x} = [u, v]^T$ through the following equation [28]:

$$\mathbf{x} = \begin{bmatrix} u \\ v \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X \\ Y \end{bmatrix} \tag{2.1}$$

The coordinates $\mathbf{X_0}$ of the point p relative to the world frame are related to $\mathbf{X}$ by a rigid body transformation:

$$\mathbf{X} = R\mathbf{X_0} + \mathbf{t} \tag{2.2}$$

where R is a 3×3 rotation matrix and $\mathbf{t}$ is a 3×1 translation vector.

Written using homogeneous coordinates, Equation 2.1 becomes:

$$Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.3}$$

When acquiring images with a real camera, this ideal pinhole perspective camera model no longer holds. We have to deal with a pixel array coordinate system, the image reference frame, where the origin is not at the optical center, and length units are in pixels instead of metric units. Thus, Equation 2.3 is typically rewritten as:

$$Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.4}$$

where s_x, s_y represent scaling factors, s_θ represents a skew factor, and (o_x, o_y) are the pixel coordinates of the principle point.

In homogeneous coordinates, the rigid body transformation from $\mathbf{X_0}$ to $\mathbf{X}$ can be rewritten as:

$$\mathbf{X} = \begin{bmatrix} R & \mathbf{t} \\ 0 & 1 \end{bmatrix} \mathbf{X_0} \tag{2.5}$$

Putting all of this together and reorganizing matrices, we get the final form for a perspective camera model:

$$Z\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & s_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix} \tag{2.6}$$

or in matrix form [28]:

$$Z\mathbf{x} = K\Pi_0 g\mathbf{X_0} \tag{2.7}$$

Here K represents the intrinsic calibration parameters, and g the extrinsic parameters. These are often combined to form one perspective projection matrix P relating world 3D coordinates to image pixel coordinate:

$$\lambda\mathbf{x} = P\mathbf{X_0} \tag{2.8}$$

Here, λ, an arbitrary scalar value, replaces Z because often the actual depth Z of an object is not known, and $P = K\Pi_0 g$.

Camera calibration typically refers to the process of solving for the *unknown* parameters encoded in P. This is typically performed using a set of images which capture multiple views of a calibration target. Calibration targets are objects where some amount of spatial information regarding the structure is known *a priori* (e.g. a planar chessboard pattern, or a cube covered with a regular grid of dots or lines). One then captures many images of this target as the target moves through different positions and orientations. In traditional calibration procedures, one would then manually click on the features in each image (e.g. chessboard corners), allowing the software to compare the detected features with the known geometry of the target. Typically, a linear least squares estimate of the calibration parameters is then found, followed with nonlinear optimization to refine the estimate (for more details see [42, 51]).

The above traditional calibration process becomes extremely difficult and time consuming when attempting to calibrate a large array of cameras. For example, in the UIUC Teleimmersion lab, there are 40 cameras that need to be calibrated at a time. These types of systems pushed the development of more "user friendly" calibration techniques, which significantly simplify the data acquisition and user input in the calibration process.

In our prototype system we used the method presented in [41] to calibrate the four visible cameras and one thermal camera. In [41], one simply moves a point light source (e.g. a small flashlight or a modified laser pointer) through the environment, while all visible cameras are capturing synchronized images. The calibration algorithm detects points as seen by each visible camera, and computes transformation parameters of P. In order to make this method work with both visible and thermal cameras, we

simply constructed a pole with an open bulb flashlight attached to the end. The pole is used to exclude the warm human from the field of view of the thermal infrared camera. When the flashlight is turned on, it quickly becomes much warmer than room temperature, thus both the visible and thermal cameras can readily detect the same point source. In this manner, one effectively reduces the computational requirements of feature matching between multiple images, because there is only one point light source in each frame captured during calibration. This method can be compared to [39], where a standard checkerboard pattern is used to calibration both thermal and visible cameras. In [39] the checkerboard pattern is heated with a flood lamp, and the grid is detected because of the different emissivity between black and white regions. We believe that Svoboda's method is more user friendly, although a thorough comparison of the these two methods was not performed.

The method presented in [41] then uses an iterative procedure that performs projective reconstruction using epipolar geometry, RANSAC analysis to remove outliers, projective depth estimation, euclidean stratification, and non-linear distortion estimation. Notice that automatic camera calibration is not a trivial task, and state-of-the-art methods such as [41] combine major results from many areas of computer vision.

Figure 2.3 shows the results of camera calibration in terms of reprojection errors for all cameras and the 3D configuration of the cameras and detected flashlight points. The camera labeled number 5 is the thermal camera, and contains the largest error, although the reprojection error is still less than 1 pixel. This is an expected calibration result because of the low resolution of the thermal camera with respect to the visible cameras. Note that even though two stereo clusters were calibrated, only one cluster was used throughout this thesis. One stereo cluster plus one thermal IR camera make up the fundamental hardware unit used in this thesis. During calibration, however, the numerical optimization process converges more quickly (and reliably) when cameras with different pointing angles and positions are used - thus we calibrated with two clusters, but only used one.

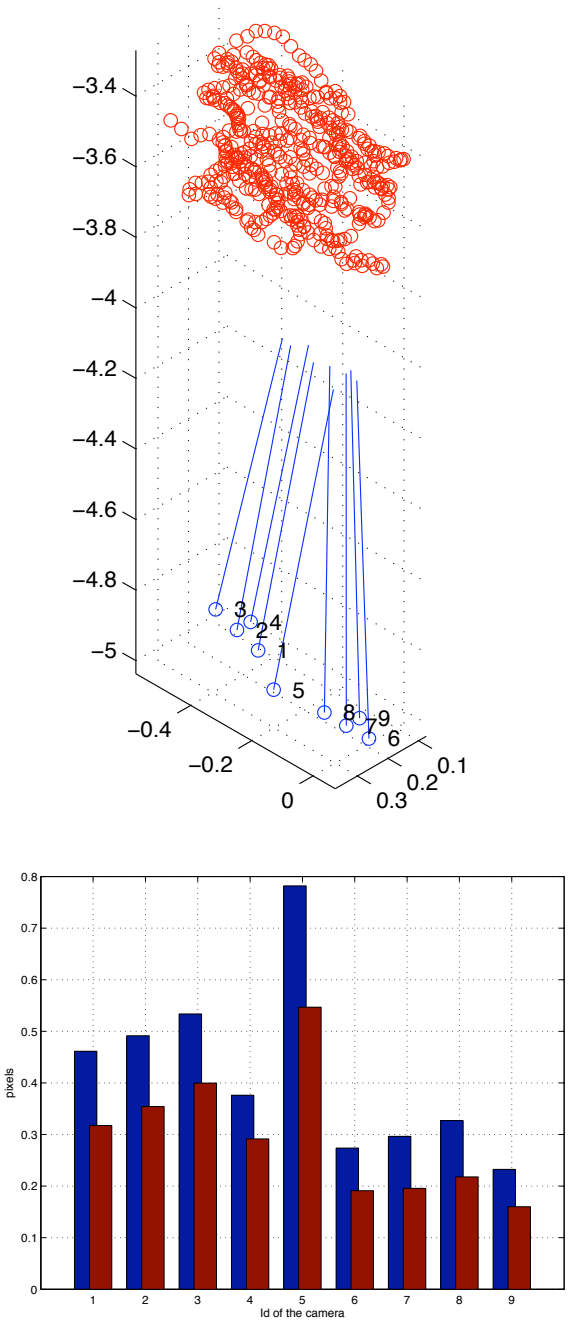


Figure 2.3: Camera calibration results. These plots are examples of output produced by the system presented in [41]. On the top, the 3D structure of the cameras and detected flashlight points are shown. On the bottom, final reprojection errors are shown. The thermal camera has the largest error, which is expected because of its low resolution.

Chapter 3

Problem Formulation

A tele-immersive environment combines objects of interest from geographically distributed camera images to create an interactive virtual world. In this chapter we describe typical objects of interest in a tele-immersive system, and discuss the features of these objects in a camera image (Section 3.1). Certain assumptions about the image make it easier to find objects of interest (Section 3.2), but these assumptions often break down in practice (Section 3.3). In particular, five characteristics of real scenes cause problems in the current TEEVE system (Section 3.4): changing illumination, moving foreground objects (causing shadows), moving background objects, lack of contrast between foreground and background objects, and lack of contrast between different foreground objects. We propose a method of fusing information from visible and thermal infrared cameras that can solve all five of these problems (Section 3.5).

3.1 Objects of Interest in Tele-immersive Systems

Tele-immersive systems are meant to facilitate communication, cooperation, and interaction between people. Therefore, the central objects of interest in a tele-immersive system are these people, the things they jointly manipulate, and the tools they need to perform this manipulation.

For example, consider a tele-immersive system designed to allow a tennis coach in Flushing Meadows, New York, to instruct a student in Champaign, Illinois. First, it is important that each player can observe the other – their appearance, their expressions and body language, their movement, and in particular their physical relationship as reconstructed in a virtual space. Second, it is important that each player can observe the other's tennis ball – its trajectory is what the coach is teaching the student to more effectively manipulate. Third, it is important that the each player can observe the other's tennis racket – this racket is the tool with which each player manipulates the ball, so its properties (such as length, head size, or weight) have an impact on each player's behavior. Moreover, all of these objects must be displayed in a common reference frame, a single tennis court that may differ in appearance from the actual one in either of the two locations.

Notice that it may not always be so easy to define objects of interest in a tele-immersive system. In the previous example, imagine that the student has either a physical or mental disability, and requires a wheelchair to move around. This wheelchair does not directly manipulate the tennis ball (like the student's racket), but it indirectly allows this manipulation to occur. Since the wheelchair is fundamental to understanding the student's motion and behavior, it should be viewed as an object of interest, and included in the student's virtual reconstruction.

For a specific application, objects of interest may have particular characteristics that influence the design of our computer vision algorithms. For example, both visible and thermal infrared cameras can capture a variety of low-level features such as reflectance, color, and temperature, as well as high-level features such as edges and texture. Although these features vary widely in general, they are distributed more narrowly in images of tennis players. For this application, people traditionally wear short-sleeved shirts and shorts that are predominately white, the ball is usually yellow and spherical, and the racket is about arm's length with a head of characteristic texture.

Of course, even within a particular application, some image features may vary. For example, even if a tennis player's clothes are predominately white, they may exhibit many different textures and highlight colors. Further, the player's average temperature can change because of physical exertion, or even because of insulated clothing. Similarly, depending on whether the court surface is hard, clay, or grass, the tennis ball might be a slightly different color, texture, or size.

So far we have focused on facilitating sports-related activities (like the tennis coaching example described in this section) with our own tele-immersive system, TEEVE. In particular, this thesis looks at a prototype system in which we ignore tools (like the tennis racket or the wheelchair) and assume that the object to be manipulated is always spherical. Broader applications, such as demonstrating the assembly of a complex part or performing collaborative medicine, are of future interest.

3.2 Typical Simplifying Assumptions

Foreground detection plays a role in a wide variety of vision applications. Generally, practical solutions to the problem of foreground detection involve making assumptions about the scene. These assumptions are made in order to reduce the complexity of the problem, allowing an algorithm to focus on a smaller set of expected measurements.

One way to look at this situation is that the complexity in the problem is derived from the fact that a natural scene deviates from an *ideal* scene. We think of the ideal scene as one which, in general, exhibits characteristics that make the problem of foreground detection more straightforward, while still being physically possible (e.g. a physically

impossible ideal background would contain material that absorbed light in all wavelengths, leaving the foreground object completely isolated in the observed image).

An ideal scene can be defined as having the following properties:

In the visible wavelengths:

- background materials have non-reflective surfaces: materials follow perfect Lambertian diffusion. This reduces specular reflections, and makes predictive illumination modeling possible.
- background materials are generally dark: this would enable one to more easily use simple intensity based models to segment a brighter foreground object from the dark background. In addition, any shadows cast on the background would tend to blend in to the dark background (in contrast to a pure white wall, for example).
- scene illumination is constant (non-changing) over time: this allows one to learn a background model and eliminate the possibility that the model can change over time (see later discussion on background subtraction techniques).
- foreground object is uniformly illuminated by diffuse lighting: this would cause a foreground object to maintain a roughly constant intensity in the observed image by reducing reflectance changes due to changes in the relative position of the subject and light sources. This also minimizes the effect of shadows cast on background (in contrast to a directional spot-light).
- scene lighting should exhibit a constant power spectrum: this would ease radiometric/photometric calibration across distributed labs (for example, this would enable similar physical object colors, like "red", to appear the same color in the integrated virtual environment).
- there is an intensity differential between background and foreground objects: this is similar to the above dark background discussion. The idea here is that, in an ideal scene, the foreground would have distinct reflectance with respect to the background.

In the thermal infrared wavelengths:

- background objects should maintain constant temperatures over time: this allows one to learn a background model, and interpret any changes in the observed image as signifying foreground object motion.
- background materials are non-reflective in thermal infrared wavelengths: surfaces such as bare or anodized metals act like mirrors, causing serious reflections.

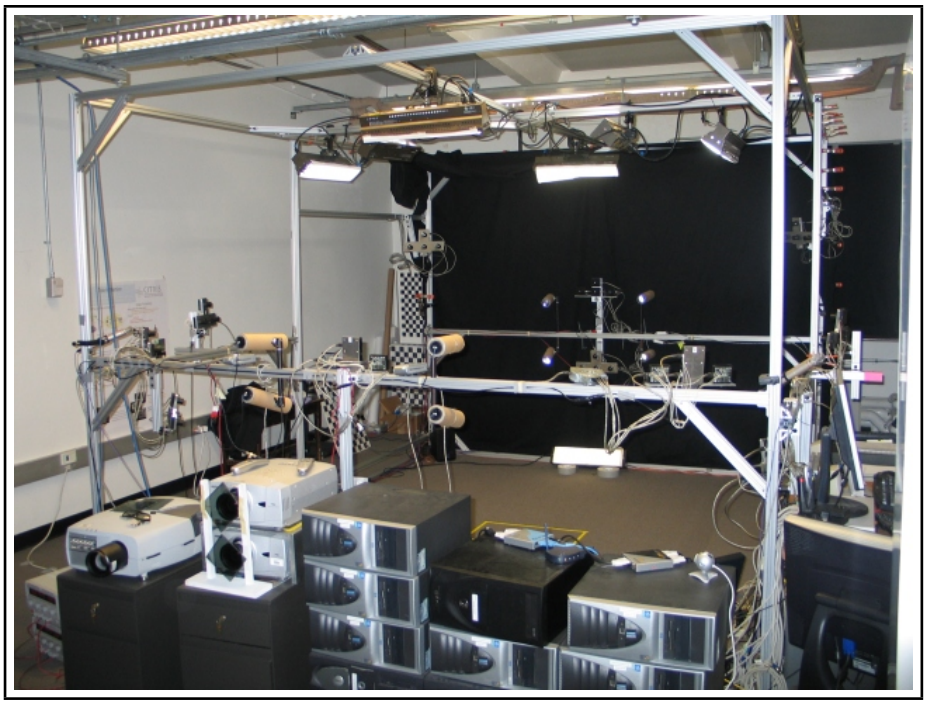


Figure 3.1: Tele-immersion lab at UC Berkeley. Black curtains are used to provide a dark, uniform, background against which brightly dressed human subjects are easily discernable. Also, special illumination is provided by the mounted lights on top of the frame.

- there is a temperature differential between background and foreground objects: similar to the discussion above for visible wavelength scenes, an ideal scene would consist of a foreground that has a distinct temperature with respect to the background (this, for example, would allow one to use simple thresholding for object detection).

In such ideal environments, visible and thermal infrared images provide a very efficient means of extracting information about foreground objects of interest. For example, background subtraction techniques can almost instantly give you knowledge of moving foreground regions. In addition, data association (object tracking) becomes straightforward because more complicated foreground models can be learned and used reliably over time for classification and recognition tasks.

Many current systems attempt to control the environment such that it approaches the above ideal (note the black curtains and complex illuminatio in Figure 3.1). However, in doing this, one quickly approaches the point where cameras, lights, and displays must be positioned *just right* in order for the system to function well, if at all, which does not meet the requirements of portability.

3.3 How Assumptions Break Down in Practice

Scenes encountered in normal office environments differ in almost all aspects from the ideal environment outlined above (in both visible and thermal wavelengths). In addition, as mentioned in Section 3.1, tele-immersive systems involve objects of interest that typically fall outside of the ideal scene definition. The following descriptions showing how assumptions break down in practice apply to both indoor and outdoor scenes. Current tele-immersive systems, however, operate within indoor environments, thus our examples will focus on characteristics of indoor scenes.

In the visible wavelengths,

- backgrounds are complicated, containing many different materials of various colors and reflectivities;
- illumination is not constant over time, nor is lighting uniform for all possible foreground object poses (for example, certain room lights can be turned on or off, and lights are more likely to be directional);
- there is, in general, no significant color/intensity differential between foreground and background objects.

In the thermal wavelengths:

- background temperatures are, in general, not constant (computers can be turned on/off, air conditioning or heating can be turned on/off, etc);
- background materials often consist of reflective materials, such as metal cabinets, chairs, etc).
- many foreground objects typically do not exhibit a temperature differential with respect to the background (for example, most inanimate objects in a room will be in thermal equilibrium with the rest of the room)

These characteristics of real scenes typically encountered in tele-immersive systems (indoor office environments), cause specific problems in 2D images. In the following section, we describe a subset of these problems which the methods developed in this thesis will focus on solving.

3.4 Current System Challenges

The current baseline TEEVE system uses static background subtraction in the grayscale image to extract foreground objects from the scene. While computationally efficient,

background subtraction methods only encode weak knowledge about the scene. In other words, the information encoded in the difference image – the result obtained by subtracting the background image from the current frame – is not very good at discriminating between background and foreground (see Section 4.2 for more details). Thus, in non-ideal scenarios, which are common in practice, foreground object detection accuracy can be significantly degraded. Specifically, problems arise that are due to:

- changing illumination
- moving foreground objects casting shadows on the background
- moving background objects
- lack of contrast between foreground and background objects
- lack of contrast between different foreground objects

Problems caused by changing illumination

Changing illumination is a cause for difficulties in many computer vision systems. The general effect that this has on the existing foreground detection system is that occurances of mis-classified regions increase. This occurs because after the lighting conditions have changed, regions in the background no longer match the background model originally developed. This effect can occur under small lighting changes when dealing with non-lambertian or specular materials, whose surface reflectivity properties make it even more difficult to predict what will happen under various lighting conditions.

A typical example of time varying illumination can be seen when office lighting changes throughout the day because of sunlight entering windows. This demonstrates drastic but gradual changes in lighting. More rapid changes can also occur when, for example, room lights are turned on or off. Figure 3.2 demonstrates the effect that a simple lamp can have. In this case, a small lamp was turned on just outside the field of view of the cameras. This causes the appearance of large portions of the scene to change with respect to the static background image acquired before the lamp was on.

Problems caused by moving foreground objects causing shadowing

In addition to illumination problems caused by the light source changing intensity or spatial configuration, problems can also be caused by the general illumination environment changing because physical objects in the scene are moved, causing shadows and reflections in the background to change. Figure 3.3 demonstrates this problem. In this example, the room lighting was not changed at all. However, when the human foreground object enters the scene, shadows are cast on the background objects. Also, shiny surfaces, such as the bottom of the computer monitor, change appearance even though they are not under direct shadowing.

Problems caused by moving background objects

When a static background model is used for foreground detection, moving objects will likely cause variations in pixel intensity over time. This is due to non-lambertian surface properties as well as color and texture variations in the object and the surrounding scene. A common example of this in our tele-immersive laboratory is the situation where a computer monitor is in the scene. We would like this computer monitor to remain part of the background even when it is displaying dynamic content (for example, a full screen animation or movie). As Figure 3.4 shows, this is not possible in the current tele-immersive system.

Problems caused by lack of contrast between foreground and background objects

When using a simple background subtraction scheme to detect interesting objects, a variety of problems arise because of this scheme's limited discriminating power.

One example of this problem can be seen when a foreground object has regions that exhibit similar image intensities as the background. (Note that even though background subtraction is applied in a grayscale image, this same phenomenon will generally occur in a color image.) This leads to false negative classification because, as far as the background subtraction technique can discern, if the pixel looks similar enough to the background model, it must be the background. This scenario is illustrated in Figure 3.5.

Problems caused by lack of contrast between different foreground objects

Similar to the difficulty in discriminating objects from the background model, background subtraction also has difficulty in discriminating foreground objects *from each other*. In our tele-immersive system, we want the ability to choose which types of objects should be included in the foreground. This functionality would not only give us the ability to actively include classes of objects in the foreground, it would also allow us to effectively filter out common objects in the scene that we do not want in the virtual environment, such as office accessories including phones, cabinets, etc.

For example, consider a tele-immersive application which requests that only red objects will be considered interesting for reconstruction. A green object entering the scene can register the same "distance" from the background model as the red object. In this case, background subtraction will consider both objects foreground. In other words, the fundamental symptom of this limitation is false positive classification.

In our prototype system developed in this thesis, we have chosen shape as the primary feature for inanimate objects of interest. Figure 3.6 demonstrates that the current system cannot distinguish between a spherical object and a rectangular one.

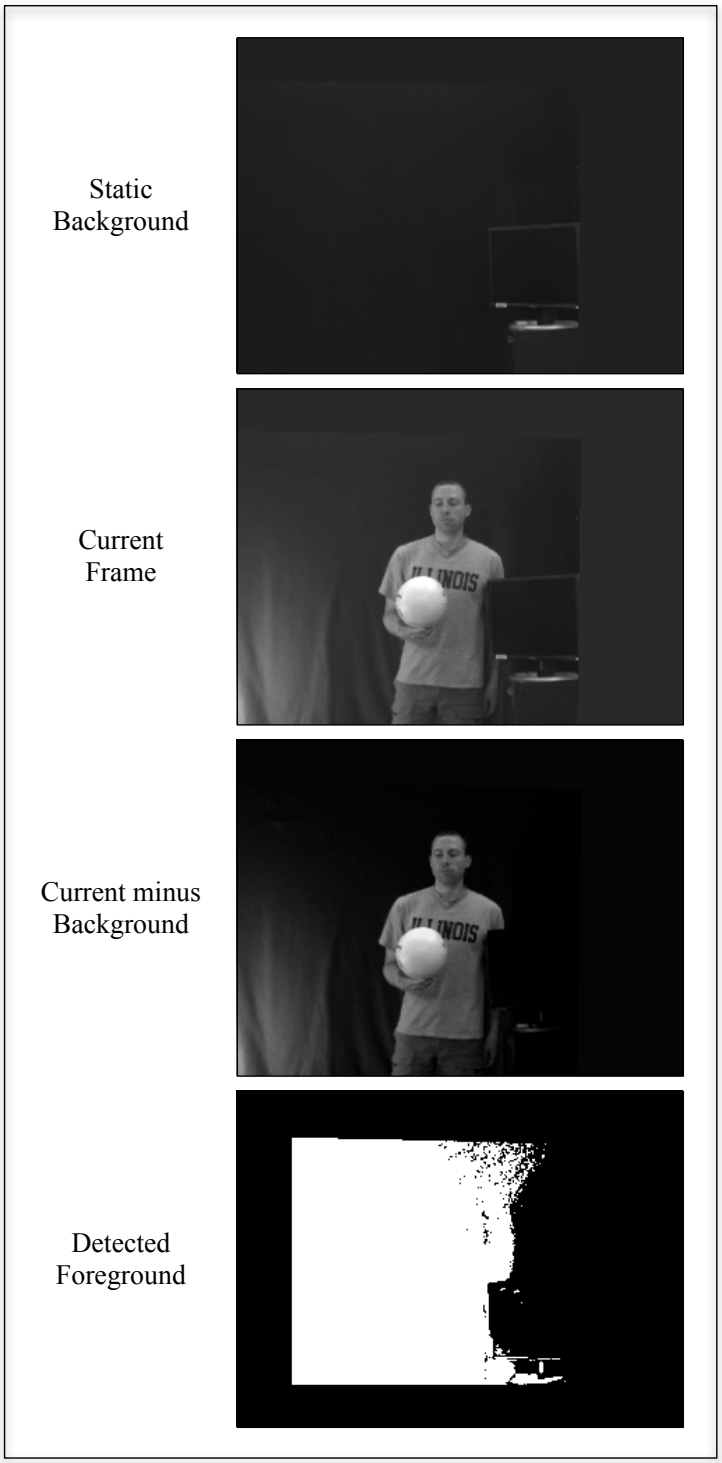


Figure 3.2: Problems caused by changing illumination: In this sequence, a small lamp was turned on, causing large portions of the background to be seen as foreground because their pixels changed intensity under the additional illumination of the lamp.

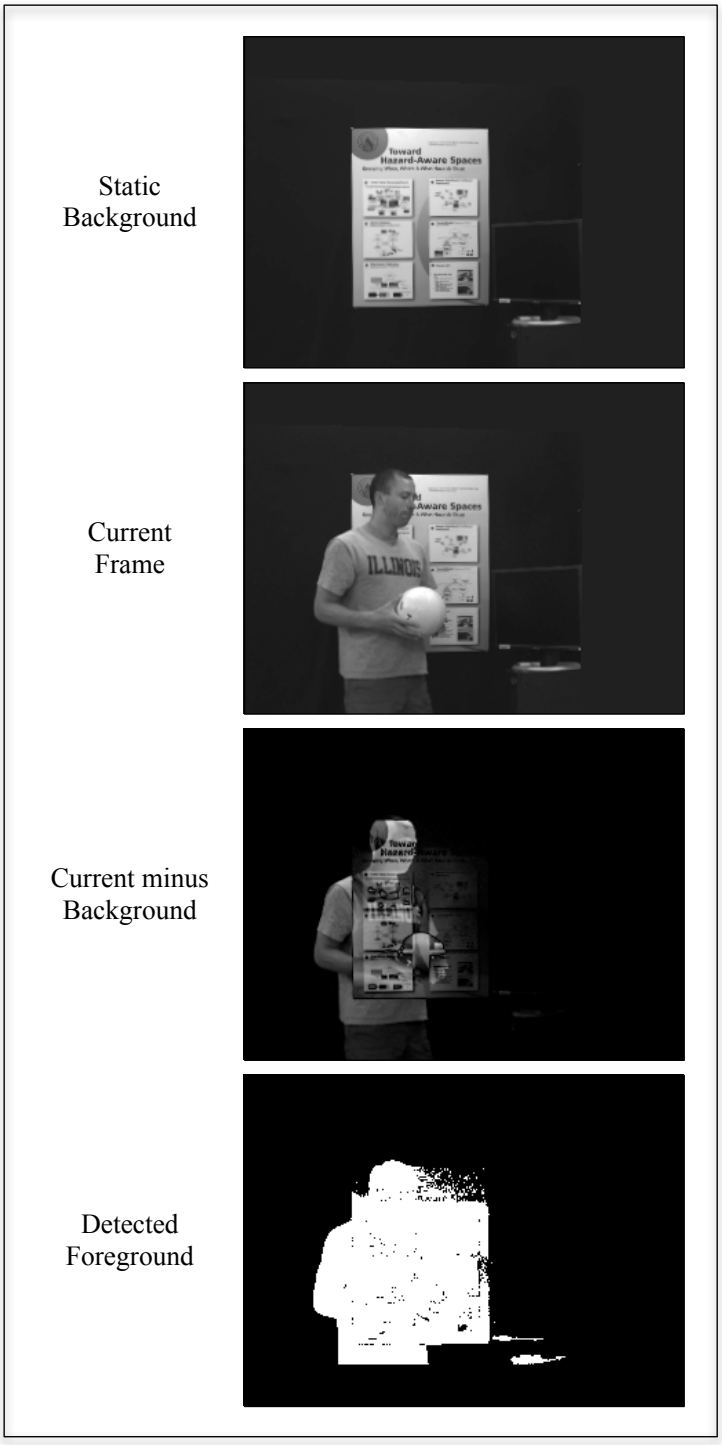


Figure 3.3: Problems caused by moving foreground objects: Moving foregrounds cast shadows and change the general scattering environment of the scene. In this sequence, background objects are mistaken as foreground due to their pixel intensities changing because of the shadowing.

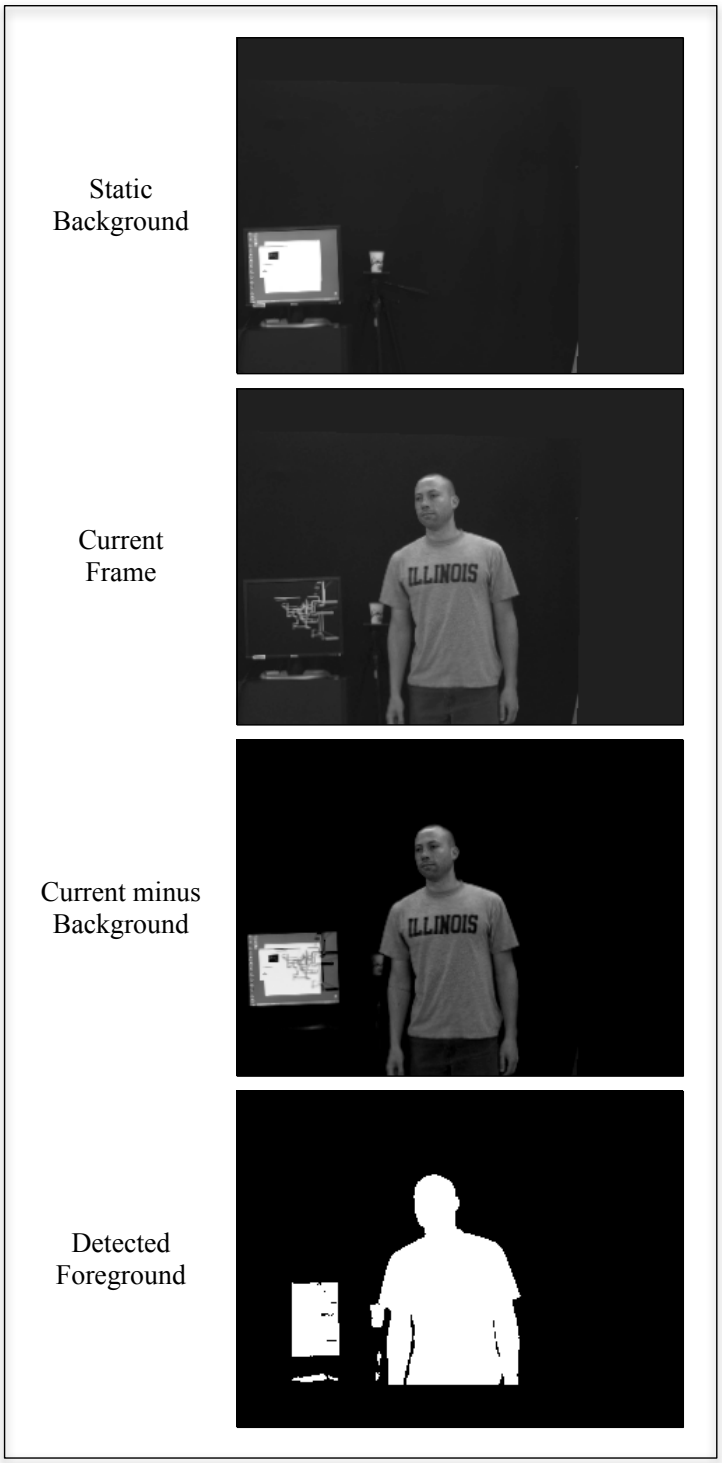


Figure 3.4: Problems caused by moving background objects: In this example, the computer display (a background object) has changed appearance between the acquisition of the background and the current frame. Thus, the current system treats these changed pixels as foreground.

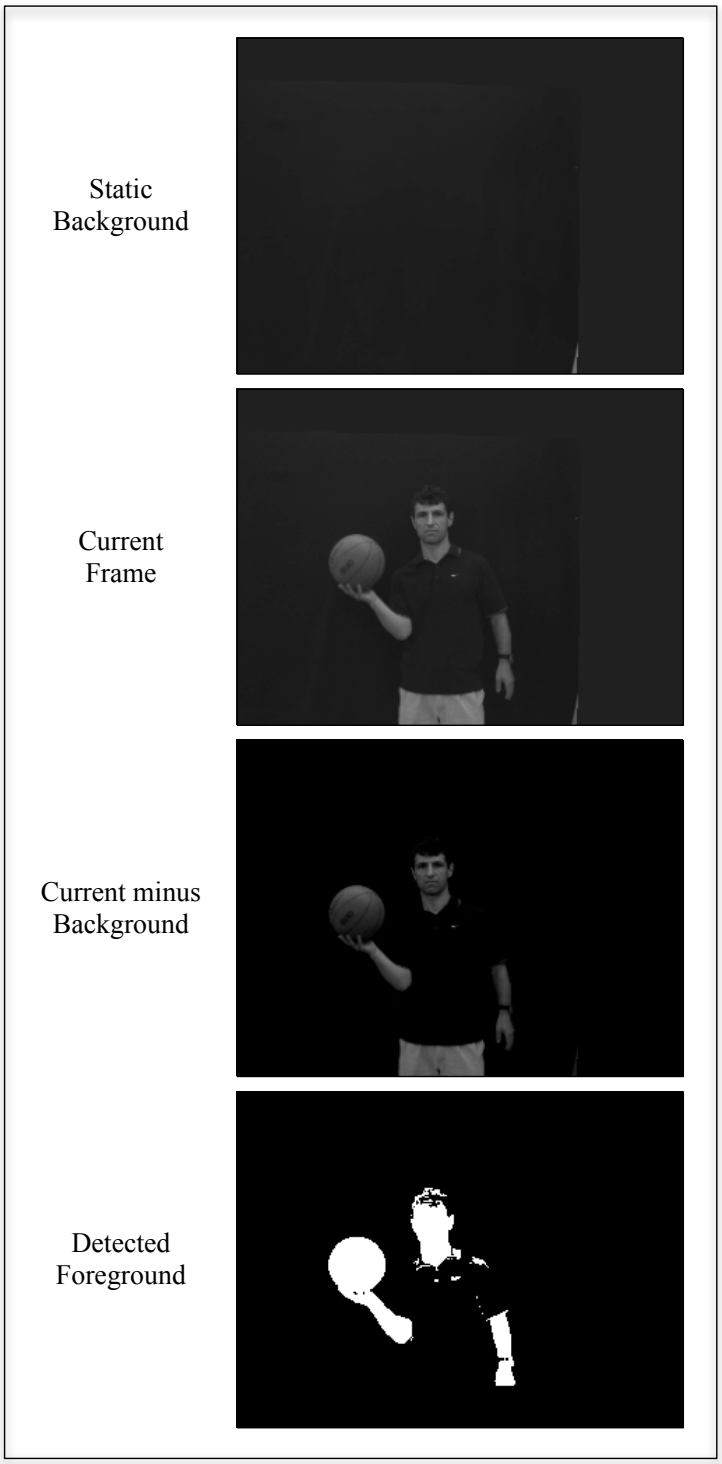


Figure 3.5: Low contrast between foreground and background: In this case, the visible modality has a difficulty classifying foreground objects that have intensities (or colors) that closely match the background model. In this example, the current system does not detect large portions of the human subject because he is wearing a dark shirt.

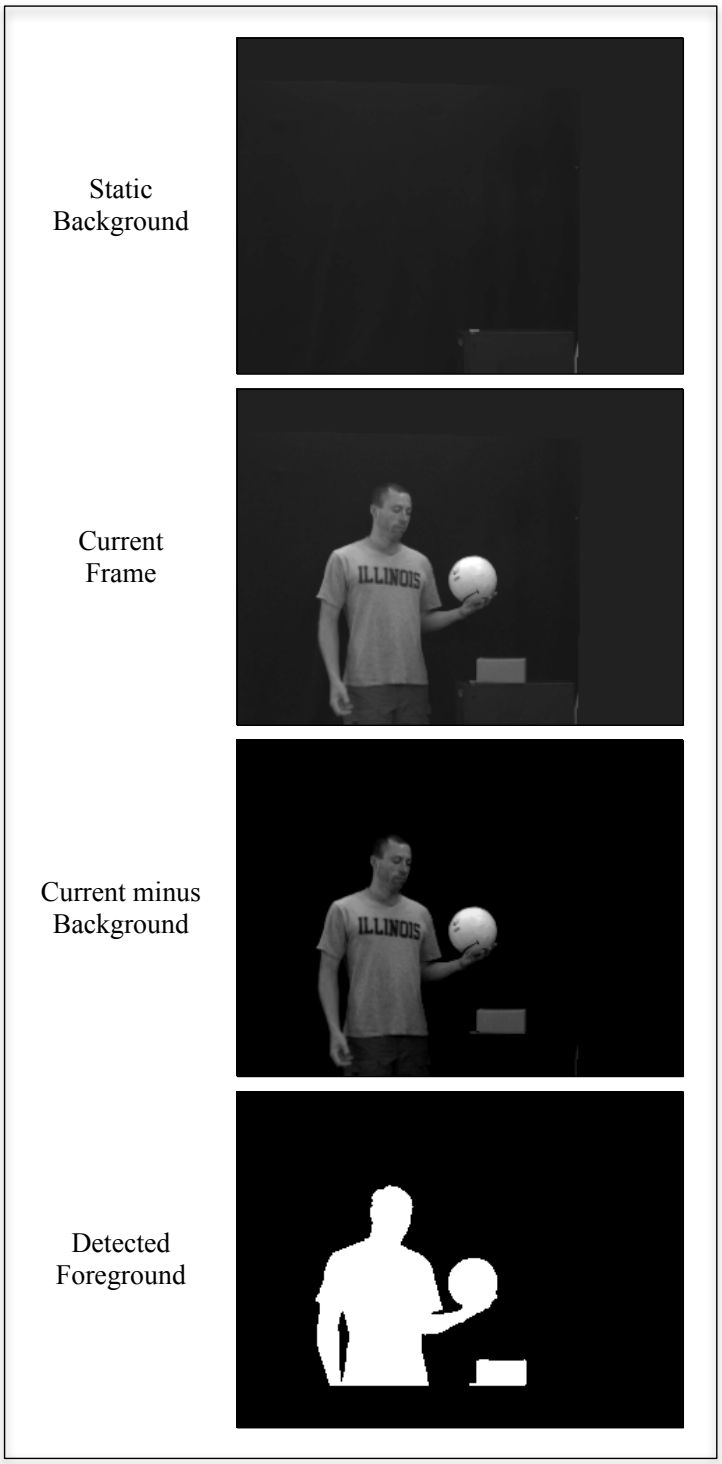


Figure 3.6: Low contrast between different foreground objects: The current TEEVE system cannot discriminate between object classes based on shape, color, etc. In this particular example, the existing system cannot distinguish between the object of interest (the ball), and a rectangular object, thus classifying them both as foreground.

3.5 Our Approach

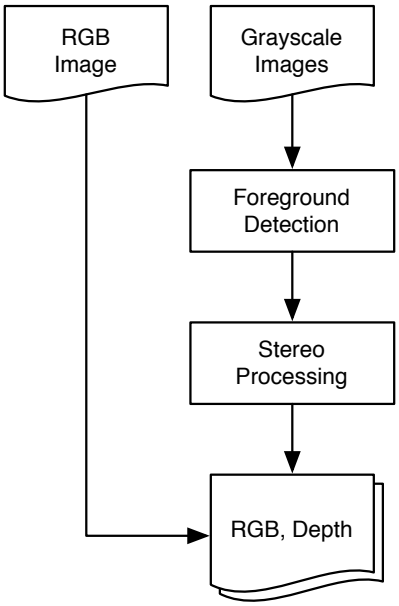
The creation of tele-immersive environments involves taking parts of the real world, or scene, and synthesizing them in a virtual world. These parts are called *objects of interest*, and the collection of these objects describe the *foreground* of the scene. As described above, it is often a difficult problem to extract these objects from the background. The approach taken in this thesis is to increase the feature space used to detect these objects by adding an additional imaging modality: thermal infrared imagery. This thesis will present a method of fusing visible and thermal imagery to perform foreground detection. Through fusing thermal infrared and visible wavelength information, we will demonstrate more robust foreground extraction in the presence of complex and dynamic backgrounds than using visible wavelength information alone.

Multi-modal image fusion was chosen as a framework for this work because each sensing modality on its own has benefits and drawbacks (Sections 3.1, 3.2 and 3.3). In other words, neither modality alone can completely solve the problem of robust foreground detection for the objects of interest in tele-immersive systems. Our fusion process will attempt to maximize the benefits of each modality by intelligently fusing their information, overcoming the limitations of each modality alone. Figure 3.7 shows the overall processing pipeline of the current TEEVE system and the modifications that our approach will introduce.

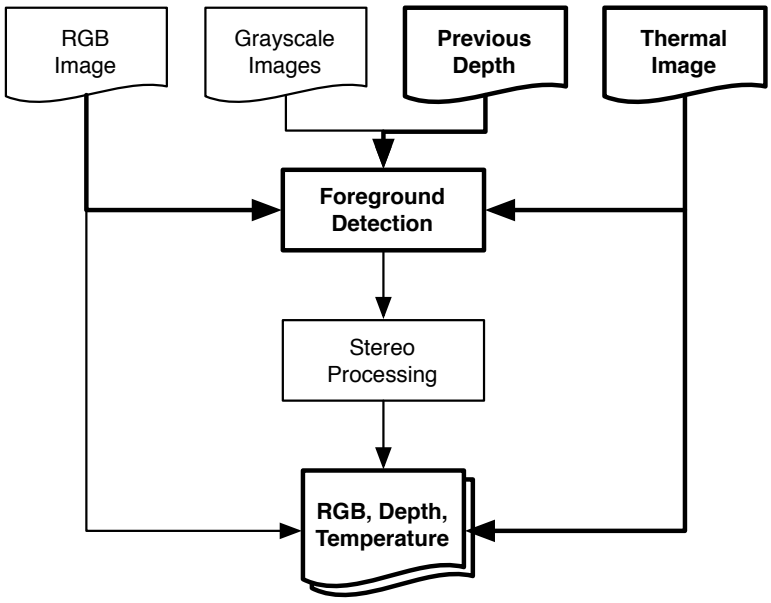
An example illustrating the benefit of combining visible and thermal information is a scene containing a person and a large computer display. In this scenario, we want to detect the human, but keep the display in the background. This is an extremely difficult case to consider. Both the human and the display will appear warm in the thermal image. Both the human and the display will appear different than the visible wavelength background model (for example, the display can be playing a full-screen video). However, when combining information from both thermal and visible images, one can see that the display does not change its spatial configuration in the thermal image, while it does change it's spatial structure in the visible wavelength. In this way, one can classify the display in a different category than the human.

Our fusion algorithm for foreground detection will consist of the following high level components:

- preliminary feature detection in visible and thermal images
- alignment, or registration, of visible and thermal information
- foreground object detection utilizing aligned information



(a) Current TEEVE system based on visible spectrum imaging only.



(b) Proposed TEEVE system based on visible and thermal IR spectrum imaging and information fusion.

Figure 3.7: High level overview of existing and proposed system: Bold blocks in the bottom figure highlight the modifications this thesis proposes.

The detected foreground from our fusion system will then be input to the next stages of the TEEVE reconstruction pipeline, which will proceed to:

- use stereo processing to reconstruct the 3D structure of detected objects of interest in real-time
- transmit this reconstructed information to remote sites, and similarly receive 3D streams from remote sites
- render an integrated virtual environment in which the subjects from all sites can interact.

Chapter 4

Fusion of Visible and Infrared: Methodology and Algorithms

In this chapter, we present our method for performing visible and thermal image fusion for foreground detection. Our system utilizes the fact that some objects of interest are best measured in one modality over the other. In particular, we use thermal imagery to primarily detect human subjects, and visible imagery to detect inanimate objects of interest. However, despite this sensor preference for a given type of object, our framework also allows for other types of information sharing between modalities.

Our feature level fusion algorithm has four components: initial feature extraction (Section 4.2), depth estimation (Section 4.3), information alignment (Section 4.4), and object detection (Section 4.5). Figure 4.1 shows how these components fit together. Before describing each component, we will first discuss related work in the area of thermal and visible image fusion (Section 4.1).

4.1 Related Work on Sensor Fusion

The need for sensor fusion arises in almost all robot and system control problems. Sensor fusion is generally the process of combining multiple observations of possibly different physical phenomena to produce a better estimate of some of those phenomena than any one measurement alone could provide. For example, in inertial measurement units, one commonly measures position, angular speed, and acceleration. The acceleration alone could be integrated to provide position, but fusing those integrated values with direct measurements will most likely reduce noise (due to filtering/integrating). This technique proves to be very useful when sensors are noisy or have intermittent failures (for example, GPS in urban environments).

The general objectives of sensor fusion are twofold. First, sensor fusion combines multiple sources of data to form a more complete or robust representation of an object. Second, fusion can reduce the multidimensionality of a multi-sensor measurement to an informative and compact representation. Another way to view sensor fusion is that it is a method of inferring some variable(s) from multiple sources of measured information.

Sensor fusion systems can operate at many different levels: at the raw data level, at a feature level, and at a decision level [18]. Note that our application calls for a pixel-

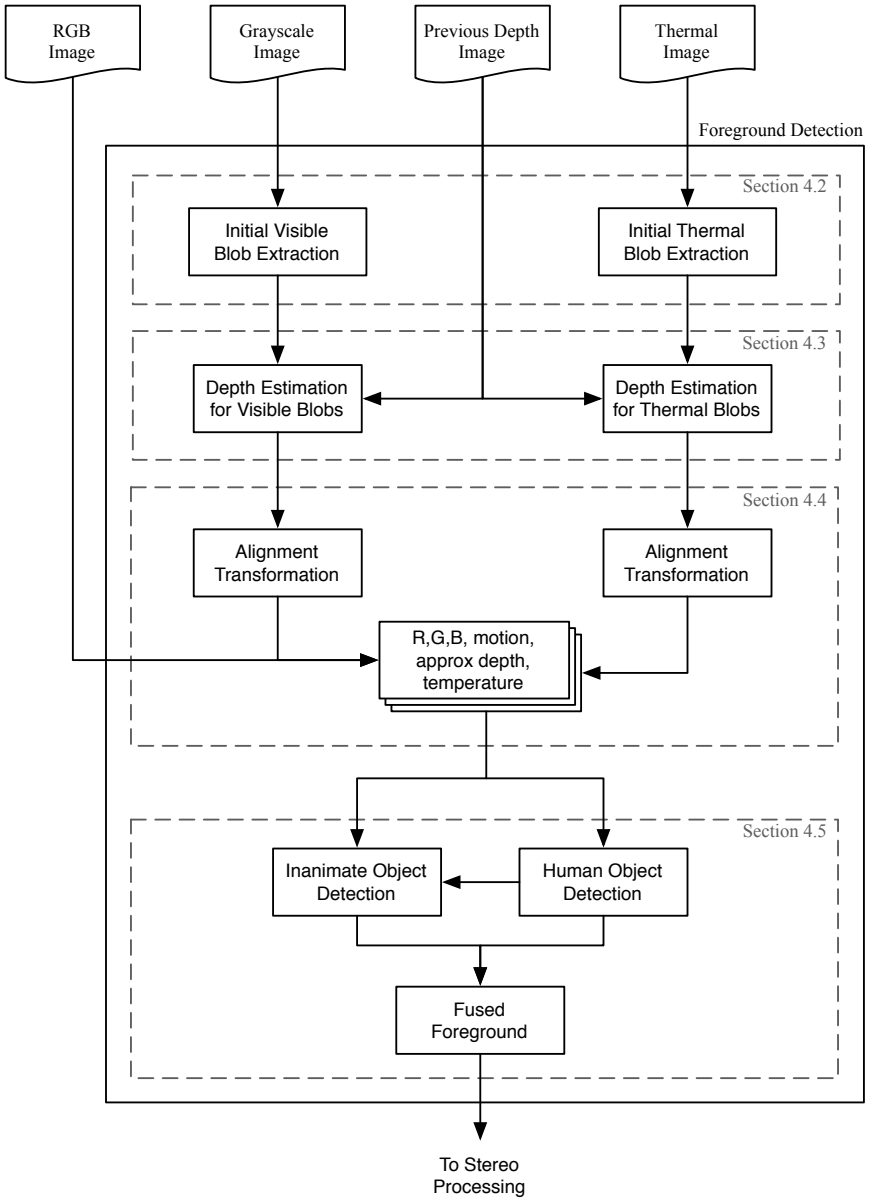


Figure 4.1: Block diagram of proposed foreground detection framework.

level result. However, how we arrive at that pixel-level result can vary. Fusion can be performed at the raw data level (after aligning infrared and visible pixels), or fusion can be performed at the feature level, which is then propagated back into the pixel level. This all requires a careful understanding of how features relate to pixel regions. Another way to look at the problem is in terms of the structure of the final result.

Thermal and visible image fusion for face recognition

A major application that utilizes thermal and visible image fusion is face detection. [1, 20, 45, 17, 7, 23] all use infrared imagery to increase robustness of face detection. These works show that standard face recognition tools work in either visible or thermal wavelengths, and that thermal imagery can result in superior detection performance across various lighting conditions and facial expressions. The techniques presented in these works primarily focus on a pixel-level fusion that results in an enhanced image containing information for further pattern recognition and classification.

[17, 7, 23] utilize wavelet transformations to perform visible and thermal fusion given registered images. A fundamental benefit of using wavelet transformations is that they decompose images into features that are prominent at different scales. The basic outline of these fusion methods is to (a) decompose the image into coefficients representing information at various scales, (b) fuse coefficients from visible and thermal images, and (c) use inverse wavelet transform to obtain the fused image.

In [23], the discrete wavelet transform (DWT) was used to fuse visible and thermal infrared images (registered) for illumination invariant face recognition. The discrete wavelet transform transformed visible and thermal images into two components, approximation and details components. These components are represented by their respective wavelet coefficients, approximation coefficients W_ϕ, and details coefficients W_ψ (see [23] for details). The fusion method consists of using a weighted average of the wavelet coefficients of the thermal and visible images.

$$W_\phi = \alpha_1 W_\phi^{vis} + \beta_1 W_\phi^{IR} \tag{4.1}$$

$$W_\psi = \alpha_2 W_\psi^{vis} + \beta_2 W_\psi^{IR} \tag{4.2}$$

The coefficients α_1 and β_1 define the weighting factors of the approximation coefficients, while α_2 and β_2 define the weighting of the detail coefficients. The final fused image is a result of using the inverse DWT of the fused wavelet coefficients W_ϕ and W_ψ.

$$I_{fused} = IDWT\left[W_\phi, W_\psi\right] \tag{4.3}$$

Thus, the problem of image fusion comes down to choosing these α and β coefficients. In [23], these coefficients are chosen experimentally, by selecting a set of coefficient choices and comparing their performance in terms of accuracy of face recognition using a set of images from a previously acquired database. In [17] and [7], genetic algorithms

(GAs) are used to select an optimum fusion strategy to combine the wavelet coefficients derived from the visible and thermal images.

In [20], a different pixel level fusion methodology is used, which is inspired by the contrast sensitivity of the human visual system. In this system, saliencies, which represent how prominent a pixel is relative to its neighboring pixels, are computed in each image at multiple length scales. These visible and thermal saliencies are then compared. If the visible saliency is much larger than the thermal saliency for a given pixel, then the visible pixel will be retained in the fused image (i.e. the visible pixel will have a weight of 1, and the thermal pixel will have a weight of 0), and vice versa. If one is not higher than the other, an average weight of the two will be computed. Finally, the fused image is the resulting weighted average of the visible and thermal images.

In contrast to pixel level fusion, [1] also presents a system based on decision level fusion. In this system, visible and thermal face recognition modules operate separately, resulting in a measure of similarity between a probe and a gallery image. Here, decision fusion consists of refining the individual scores by either (a) taking their average, or (b) taking the maximum score. Results for both of these schemes are presented, but overall, decision fusion using the average matching score gave the best performance.

Thermal imagery for human body detection

[19, 8] both utilize thermal imagery for full human body detection. The method presented in [19] focuses on human silhouette fusion between visible and thermal images, while [8] utilizes thermal stereo and human head modeling to detect pedestrians.

While [8] does not currently fuse thermal images with visible, they do mention that in urban environments, buidings, cars, and other objects can lead to false positive detections, and that the inclusion of additional sensors is expected to increase robustness.

In [19], it is assumed that humans are the only moving objects in the scene. Thus, background subtraction techniques are used to find initial human silhouettes in each image. These initial silhouettes are aligned using genetic algorithms, and then fused using probabilistic strategies to obtain a more robust human silhouette extraction. These strategies, or rules are: the product rule, sum rule, max rule and min rule, which specify that a pixel location (u, v) belongs to a human silhouette S if:

$$\text{Product rule:} \quad P(c(u,v) \in S)P(t(u,v) \in S) > \tau \tag{4.4}$$

$$\text{Sum rule:} \quad P(c(u,v) \in S) + P(t(u,v) \in S) > \tau \tag{4.5}$$

$$\text{Max rule:} \quad \max\{P(c(u,v) \in S), P(t(u,v) \in S)\} > \tau \tag{4.6}$$

$$\text{Min rule:} \quad \min\{P(c(u,v) \in S), P(t(u,v) \in S)\} > \tau \tag{4.7}$$

where P represents the probability that a vector is in S, τ is a given threshold, c represents the color vector for the given pixel, and t represents the thermal value. The

probabilities are estimated using:

$$P(c(u,v) \in S) = 1 - e^{-\|c(u,v)-\mu_c(u,v)\|^2} \tag{4.8}$$

$$P(t(u,v) \in S) = 1 - e^{-\|t(u,v)-\mu_t(u,v)\|^2} \tag{4.9}$$

where μ_c represents the average background color vector, and μ_t represents the average background thermal value, which are computed beforehand from scenes which contain only background.

Relation to our tele-immersion application

The above applications of face detection and human pedestrian detection are related to our tele-immersion application in that humans and their activities are of central interest. The above applications of thermal imagery, and visible-thermal fusion differ our tele-immersive system in three fundamental ways. First, the image resolution required for tele-immersive systems lies in between that required for face detection and that required for outdoor pedestrian detection. This resolution range effects key factors such as: how often will a foreground object be in the field of view, occluded, and what scales of motion can be uniformly detected. Second, tele-immersion applications are currently focused on indoor environments. Thus, there are no solar radiation effects, nor seasonal variations. Third, the main difference between TEEVE application and face/pedestrian detection is that we are interested in partial and complete views of humans and inanimate objects (face detection is always partial, and pedestrian is always complete view).

However, because objects of interest in our system can exhibit a large variability at the pixel level while still belonging to the same class, this becomes extremely difficult. For example, reflectance and color of human clothing can exhibit a wide range of values, all of which should be classified as foreground. Similarly, in the thermal wavelengths, a given pixel may be warm because it belongs to a person (foreground) or a hot mug of coffee (background). In such scenarios, observing pixel level information is insufficient. In order to overcome this, more sophisticated features must be used, such as region or motion based features.

It is often stated that raw data fusion leads to the most accurate fused results [18]. The assumption here is that as one abstracts higher and higher level concepts, low-level information is invariably lost. For example, tracking a centroid of a region, instead of its pixel by pixel area. However, there are two assumptions here that lead to interesting implications: (a) the alignment of multisensor data contains small errors, and (b) the information one is interested lies at the pixel level.

Even when image registration is achieved, it is performed in the presence of simplifying assumptions such as: all features lie in a certain plane and regions can be mapped using planar homographies. In our application in particular, where a complex 3D articulated object is being viewed in close proximity, global feature registration is extremely

difficult. In contrast to dealing with face recognition problems, it is no longer the case that local regions can be treated as planar. The entire 3-D pose is important. Also, because the visible and infrared cameras are looking at the human subject from potentially different viewing positions and angles, each image will, in general, observe different surfaces of the subject. Thus, it is conceivable that alignment errors could get large enough that feature matching leads to more accurate results than pixel level fusion.

Further, depending on the pattern classification task, pixel level information might not be an important component of the end product to be classified. In this case, pixel level fusion can actually be detrimental by causing reduced resolution after averaging and interpolation. In these cases, detecting higher level features in each sensor followed by feature matching and voting can outperform pixel level fusion.

It is interesting to note here that the class of objects of interest has a fundamental impact on the design of the fusion algorithm used to detect those objects. For example, in this proposed algorithm, we are limiting ourselves to be interested in the objects shape, specifically spherical objects. Thus, we are essentially allowing users to interact using these objects, despite their color, size, or motion. Thus, based on our consideration of pixel-level and feature-level fusion, our approach is to use feature-level fusion.

4.2 Initial Feature Extraction

The first stage of our algorithm consists of preliminary feature extraction in the visible and thermal images. The purpose here is to select regions to be aligned and further analyzed in subsequent steps of our fusion pipeline. To extract preliminary areas, we utilize a simple background subtraction technique.

As mentioned earlier, the goal of background subtraction is to isolate objects of interest from the background. That is, background subtraction is often used as a technique of foreground/background segmentation. Background subtraction techniques can vary in complexity, ranging from simple frame differencing to more complicated background modeling maintenance. In general, background subtraction relies on one having some model of the background. This model can be static or dynamic, and is subtracted from the current frame, resulting in a difference image (most often the absolute difference is the value computed).

One extreme example of background subtraction is blue screen chroma keying. This is a common technique in video production used to extract foregrounds. The background is usually a blue or green screen, whose color and intensity are assumed to be known *a priori*. However, in applications where a fixed color background is obtrusive or impractical, the common alternative is model-based background subtraction.

When a static background is assumed, one way to build a model of the background is to capture a series of N images, $B(k)$, and gather statistics such as average pixel value and average pixel difference [32]:

$$B_{ave} = \frac{1}{N} \sum_{k} B(k) \tag{4.10}$$

The average pixel difference within these N images of the background is then given by:

$$D_{ave} = \frac{1}{N} \sum_{k} (B(k) - B_{ave}) \tag{4.11}$$

This equation typically encodes low levels of noise, and possible small fluctuations of appearance in the background scene. In other words, D_{ave} provides information about how uncertain we are about the background model B_{ave}.

Given such a background model, a simple foreground classification can be performed by thresholding the difference D of the current frame at time k and the average background:

$$D(k) = I(k) - B_{ave} \tag{4.12}$$

The threshold is commonly a factor of D_{ave}. Thus the binary foreground mask M, is computed pixel by pixel in the following manner, where 1 represents foreground and 0 represents background:

$$M(x,k) = \begin{cases} 1 & \text{if } D(x,k) > \alpha D_{ave}(x) \\ 0 & \text{otherwise} \end{cases} \tag{4.13}$$

where x is a particular pixel location.

Note that model-based background subtraction schemes are basically designed to detect "movement" in the image. Assuming a static background, and constant lighting conditions, this movement will be solely due to physical motion in the scene. In the simple background subtraction scheme shown in Eqn. 4.10, motion is measured against a static time in the past.

Under some conditions, this method works extremely well. For example, when a dark, uniform, screen fills the entire field of view of the cameras, brighter foreground objects are accurately detected when they enter the scene. Dark backgrounds also tend to perform well because their image tends to be at the low end of the camera's dynamic range.

In general, however, this simple subtraction method is quite noisy, and is often insufficient for accurate foreground detection. This method breaks down when the environment and lighting conditions conspire to cause the background to be unstable over time. For example, if the system is within a small office environment, shadows, both direct and more complicated illumination changes due to reflections, can easily

cause the lighting on office walls or other background objects (tables, chairs, etc) to change drastically over time. Further, if large computer displays are in the environment and part of the background, there will clearly be ambiguity between motion in the display and physical motion in the local environment. This factor currently limits tele-immersive coverage to roughly 180 degrees because the cameras are confined to look at uncluttered areas of the environment.

More complicated versions of this method involve combining multiple filtering paths to attempt to combine motion and edge information [29]. However, even this more involved method will fail when the background in the current scene is no longer modeled by the originally constructed background model B_{ave}, D_{ave}.

So why not just set the threshold higher? The threshold used in the simple foreground classification could be increased or decreased in the attempt to perform a more accurate reliable classification. Increasing the threshold will reduce the occurance of false positive foreground regions, yet it will also reduce the detection of true foreground objects (see Figure 1 in [19]).

If the background is changing *slowly*, there is a common extension to the simple background subtraction algorithm, which keeps track of a moving average of the background [16, 48]. These methods treat a particular location in the background as a weighted average of previous values for that location.

In [48], a pixel-by-pixel background model is maintained and dynamically updated. A pixel is selected as a candidate to update the background model if it has not changed (beyond some threshold) for some amount of time. For selected pixels, the background model at that pixel location is updated using a convex combination of its previous value and the current value in the frame:

$$B_{i,j}(k) = \alpha I_{i,j}(k) + (1 - \alpha) B_{i,j}(k - 1) \tag{4.14}$$

where $B_{i,j}(k)$ represents the value of the background model at pixel (i, j) and time k. This formulation allows the background to adapt quickly or slowly depending on the weights. In order to deal with gross illumination changes, [48] also keeps track of the average value of the frame-to-frame difference. If this value exceeds a threshold, then one can assume that they should update their background model more quickly. Note that in this work, one assumes that they have to constantly be on guard against a changing background. That is, if a foreground object stays still it will eventually become part of the background! This is not generally desired in tele-immersive systems, where humans can be interacting and changing motion patterns throughout system use.

These shortcomings of even the more robust background subtraction methds, in combination with the desire to minimize computational burden, led us to utilize the simple background subtraction methods shown in Equations (4.10) through (4.13) for both visible and thermal image sequences.

Connected components

Once we have the preliminary binary mask image, $M(k)$, we perform connected component analysis. This region partitioning and labeling will allow us to keep track of properties for segmented regions in subsequent stages of our algorithm.

Connected component analysis is typically performed on binary or grayscale images. One of the major features of a connected component algorithm is the connectivity measure used. This method typically works by scanning an image and defining regions of adjacent pixels that share the same intensity value. A common technique to perform this scanning is also referred to as flood filling. Flood fill is an algorithm that takes a seed location, and expands into regions where pixels have the same intensity as the seed pixel. A variant in this algorithm is created by allowing the target value to float by comparing the current pixel with neighboring pixels, instead of the seed pixel.

Other methods include morphological filtering operators and contour detection. Our system utilizes a contour detection method. Here, contour detection is performed on a binary image, with each region defined by a contour labeled as a particular connected component.

At this point, we have two sets of regions, $S^{vis} = \{S_1^{vis}, \ldots, S_n^{vis}\}$ and $S^{ir} = \{S_1^{ir}, \ldots, S_m^{ir}\}$, where n is the number of connected components found in the visible image, and m is the number of connected components found in the thermal image.

4.3 Depth Estimation

From the previous step of our fusion algorithm, we have obtained preliminary regions defined in both the visible and thermal images. Our goal in this step is to estimate the depth of each pixel in these preliminary regions. Knowing the depth of each pixel in a region gives us an estimate of the 3D structure of that region. This structure is later used to align regions from the visible and thermal images in a common reference frame (Section 4.4).

In order to maintain real-time performance, this stage utilizes depth maps computed in the previous time step by the TEEVE pipeline. This results in an estimation process because of two factors. First, between time $k-1$ and the current time k, the objects of interest will generally have changed position in the scene. Thus, our depth map no longer perfectly corresponds to the current region of interest. Second, the depth map is noisy and has missing data due to lack of successful stereo correlation in the previous time step. This missing data can be due to lack of texture in the stereo images, from occlusion of the object in one frame of the stereo images, or because the feature appeared differently in each stereo frame because of non-ideal material properties.

Depth in thermal frame

The first step in estimating depth for a preliminary region S is to align the depth map with the reference frame that S is in. This requires that we both align the depth map that results from the TEEVE stereo processing, and make sure that the depth values are represented in the correct reference frame. Because the depth map is defined in the visible image coordinate system, no additional work is necessary to align it with visible regions S^{vis}. However, we will need to align the depth map with thermal regions S^{ir}.

To perform this alignment, we first convert the depth map into a list of 3D coordinates, one for each pixel in the depth map containing a valid depth value (i.e. a non-zero depth value). For a typical depth value λ^{vis} at pixel location $\mathbf{x}^{vis}$, we invert the intrinsic calibration matrix K (from Equation 2.7) to recover the 3D coordinates $\mathbf{X}$ in the grayscale camera reference frame:

$$\lambda^{vis}(K^{vis})^{-1}\mathbf{x}^{vis} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{X} \tag{4.15}$$

where $\mathbf{x}^{vis} = [u, v, 1]^T$, $\mathbf{X} = [X, Y, Z, 1]^T$, and K^{vis} is the 3×3 matrix containing the intrinsic calibration parameters for the visible camera (see Section 2.3 for details). Note that in our prototype system, the grayscale camera reference frame is treated as the world reference frame, thus $\mathbf{X} = \mathbf{X_0}$ (this is the center grayscale camera of the trinocular stereo cluster). Otherwise, we would have to transform $\mathbf{X}$ through a rigid body motion to get from the visible camera frame to the world frame.

Next, we re-project the 3D world point into the thermal image:

$$\mathbf{x}^{ir} = P^{ir}\mathbf{X} \tag{4.16}$$

where P^{ir} comes from calibration (see Equation 2.8). At this point, we have a list of pixels in the thermal image, and their associated depth $(\mathbf{x}^{ir}, \lambda)$. However, the depth value is still with respect to the visible camera. To find the depth of the pixel $\mathbf{x}^{ir}$ with respect to the IR reference frame, we use the rigid body transformation:

$$\mathbf{X}^{ir} = R^{ir}\mathbf{X} + \mathbf{t}^{ir} \tag{4.17}$$

where R^{ir} and t^{ir} represent the extrinsic calibration parameters (see Equation 2.6). Now, the Z-component of $\mathbf{X}^{ir}$ gives us the correct depth of the thermal object with respect to the thermal camera.

At this point, we have aligned the depth map from the previous time step with both the visible and thermal frames. These are represented by images $I^{vis}_{\lambda_{old}}$ and $I^{ir}_{\lambda_{old}}$. These are subscripted old because we will next use this depth information (which was computed in the previous time step) to estimate the *current* depth of each pixel in regions S^{vis} and S^{ir}.

Depth estimation for regions S^{vis} and S^{ir}

One key intention in this phase is to implement a fast depth estimation for current blobs. If time were not an issue, many interesting methods could be employed to more accurately and robustly estimate depths for the current time step. For example, a blob could be considered a union of smaller regions (e.g. derived from depth and location based k-means clustering). These regions could then be matched between time steps, and depth estimation would only occur between matched regions. This would reduce the chances of giving foreground objects incorrect depth estimates. Further, we could simply use stereo correlation to recompute the depths of the regions S^{vis} and S^{ir}. However, it is one of the goals of foreground detection to extract important parts of the scene which will then be input to the computationally intensive modules such as stereo correlation.

Our preliminary region depth estimation is actually a two stage problem. First there is a blob association problem: which blobs in the current image correspond to blobs in the previous image. Second, once one knows the blob correspondence, the task remains to estimate depth values for each pixel of the new blob. How this is done depends on how the blobs are modeled. If blobs are simply lists of pixels belonging to one foreground object (i.e. there is no higher level modeling of sub-components of a blob), then one method of performing this mapping is to simply take an average of nearby depths. For each pixel $\mathbf{x} \in S$ we can compute a depth:

$$\lambda_{new}(\mathbf{x}) = \frac{1}{m}\sum_{i}^{m} \lambda_{old}(\hat{\mathbf{x}_i}) \quad \forall \quad \hat{\mathbf{x}_i} \in nbhd(\mathbf{x}) \tag{4.18}$$

Here, $\hat{\mathbf{x}_i}$ are pixel locations where $\lambda_{old}(\hat{\mathbf{x}_i}) \neq 0$, and $nbhd(\mathbf{x})$ represents a local neighborhood of $\mathbf{x}$ (e.g. a circular region centered on $\mathbf{x}$). $\lambda_{old}()$ represents the aligned depth map from the previous time step.

Another simple approach, the one which we use in our prototype implementation, is to simply use the nearest non-zero depth from the previous time step.

$$\lambda_{new}(\mathbf{x}) = \lambda_{old}(\hat{\mathbf{x}}_{min}) \tag{4.19}$$

where,

$$\hat{\mathbf{x}}_{min} = \underset{\hat{\mathbf{x}}}{\operatorname{argmin}} \|x - \hat{\mathbf{x}}\| \quad \forall \quad \hat{\mathbf{x}} \quad \text{such that} \quad \lambda_{old}(\hat{\mathbf{x}}) > 0 \tag{4.20}$$

For a particular pixel location $\mathbf{x}$, this method will directly use the value from λ_{old} if it exists at that location, otherwise it will use the nearest value.

An approximate algorithm to compute these depth estimates can be obtained by simply keeping a list of previously seen depths, and upon arrival at a pixel location that does not have an associated previous depth, use the last seen depth. In this case, one can choose intelligent methods of scanning through the image array so that it is more likely

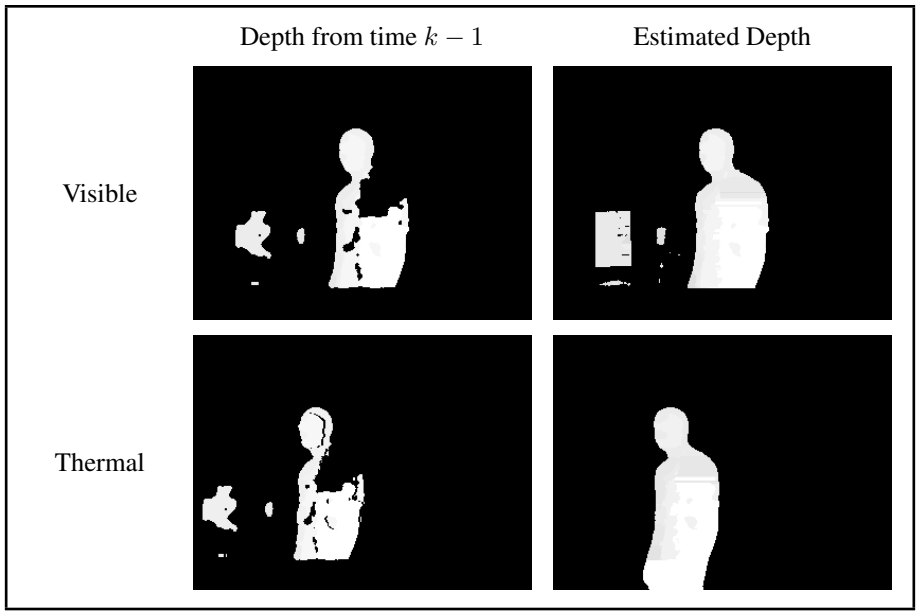


Figure 4.2: Depth estimation: Depth from time $k-1$ can be noisy and incomplete. We estimate depths for each pixel in preliminary areas of interest in both visible and thermal images.

that your last seen depth will be from the current neighborhood. For example, one would not want to scan row by row, starting at the left side of the image. This is an interesting topic in its own right. However, a thorough analysis is outside the scope of this thesis. For the purpose of our prototype system, we chose a scan method that empirically provided satisfactory results: scanning in a snake like pattern row by row. Figure 4.2 shows the results of this operation.

4.4 Information Alignment

Information alignment is one of the most critical stages of our foreground detection algorithm. The ability to correlate multi-modal observations of an object lets us move on to modeling and recognition tasks in a richer feature space. In this stage, we take preliminary regions S^{vis} and S^{ir} in the visible and thermal images, which now have associated depths for each pixel, and project these regions into the color camera reference frame.

Information alignment arises in various computer vision applications such as surveillance, remote sensing, and medical imaging [9, 24, 3, 25]. These techniques are designed to align two images so that features of interest in each image are transformed, or warped, to the same pixel locations in some reference frame – usually one of the original camera reference frames. In general, this process is required when an object is observed from more than one spatial location and orientation. For example, one could be acquring images from a large camera array, or capturing video sequences from one

moving camera. Note also that registration techniques can also be used to align data that has changed over time even though the sensor remained fixed. In many ways, this directly relates to the problem of image based object tracking.

Typical image alignment algorithms contain the following components: image pre-processing, feature extraction, feature matching, tranformation parameter estimation, post-processing of transformed image [9, 24]. This methodology works well in practice for many scenes, especially those that involve uncalibrated cameras, and require a simple transformation such as a rigid body, or affine transformation.

However, in our tele-immersive system we have two pieces of information that enable us to deviate from the above methodology: (1) camera calibration, and (2) depth maps derived from stereo processing. These two pieces of information provide us with a powerful capability. Knowledge of how far an object is from the camera (depth) combined with knowledge of where the camera is in a world reference frame (calibration) directly give us the 3D position of the object in that world reference frame.

Knowledge of depth and camera pose have also been utilized in related applications. For example, [45] uses camera calibration and 3D head pose information to register thermal IR and visible images for face detection. In [45], however, depth was estimated at only a few locations, or tie points. In contrast, our application projects entire regions from one image to another.

The use of depth information is attractive from another perspective as well. Because a single affine transformation is not sufficient for image registration of a complex scene with varying depth, we would like to use a transformation model that allows us to more accurately perform the registration. Here we note that if 3D structure of an object is completely known, then the full perspective projection using calibration information is the best transformation model we can use. However, given imperfect information (derived from calibration error or uncertainty in the object's 3D structure), the accuracy and utility of the full perspective projection decreases. Here we realize that an understanding of other registration techniques remains important, even when some knowledge of object depth is available.

We avoid matching features between thermal and visible images by utilizing camera calibration parameters and depth estimates of objects in the scene. In effect, the estimated depth of an object acts as a proxy for feature matching. This makes sense intuitively as well: the original depth map is created by performing stereo correlation, which is in effect an attempt at dense feature matching.

The specific operation of our alignment process will be to transform the thermal, grayscale motion, and depth images into the color camera reference frame.

Recall that from camera calibration, we have intrinsic parameters K^i and extrinsic parameters R^i, and $\mathbf{t}^i$, where $i \in \{ir, vis, rgb\}$ represents a particular camera with

label ir representing the thermal camera, vis the grayscale visible camera, and rgb the color camera.

The projective transformation of a 3D point (in the world frame) onto an image plane is defined by the calibration parameters of the camera:

$$\lambda \mathbf{x} = P\mathbf{X}_0 \tag{4.21}$$

where $\mathbf{x} = [u, v, 1]^T$ is the pixel location in the image, $\mathbf{X}_0 = [X_0, Y_0, Z_0, 1]^T$ is the 3D location of the point in the world frame, and λ represents the (estimated) depth of the point in the camera reference frame (we stop using the new subscript for more convenient notation). Note that $\lambda \neq Z_0$ but the z-component of the 3D position in the camera reference frame. P is the 3×4 perspective projection matrix given by:

$$P = K\Pi_0 g = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & \mathbf{t} \\ 0 & 1 \end{bmatrix} \tag{4.22}$$

Now, we use this perspective projection to transform the regions S^{vis} and S^{ir} into the color camera reference frame.

First, given a pixel $\mathbf{x}^c \in S^c$, where $c \in \{vis, ir\}$, we compute the 3D coordinates $\mathbf{X}^c$ with respect to the camera by:

$$\lambda^c (K^c)^{-1} \mathbf{x}^c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{X}^c \tag{4.23}$$

Next, we use the rigid body transformation in Equation 2.2 to find the 3D coordinates $\mathbf{X}_0$ in the world frame (which is equal to the vis camera frame):

$$\mathbf{X}_0 = (R^c)^T \mathbf{X} - \mathbf{t}^c \tag{4.24}$$

Finally, we use the projection matrix of the color camera to project $\mathbf{X}_0$ into the color image reference frame:

$$\lambda^{rgb} \mathbf{x}^{rgb} = P^{rgb} \mathbf{X}_0 \tag{4.25}$$

Even though we could compute λ^{rgb} at this point, we can divide Equation 4.25 by λ^{rgb} to get a more direct expression for the pixel coordinates of $\mathbf{x}^{rgb} = [u, v, 1]^T$:

$$u = \frac{\boldsymbol{\pi}_1^T \mathbf{X}_0}{\boldsymbol{\pi}_3^T \mathbf{X}_0}, \quad v = \frac{\boldsymbol{\pi}_2^T \mathbf{X}_0}{\boldsymbol{\pi}_3^T \mathbf{X}_0} \tag{4.26}$$

where $\boldsymbol{\pi}_1^T, \boldsymbol{\pi}_2^T, \boldsymbol{\pi}_3^T \in \mathbb{R}^4$ are the three rows of the projection matrix P^{rgb} [28].

4.5 Object Detection

Now that the information from grayscale, thermal, and color images are aligned (in the color camera reference frame), we can proceed to extract feature vectors for the regions S^{vis} and S^{ir}. We will use these feature vectors to classify these regions as foreground or background. Finally, the union of all detected foreground regions will define the final foreground.

One approach would be to look at the pixel level. At this level, we have six pieces of information: grayscale difference from static visible background model, red value, green value, blue value, temperature difference from static thermal background, and approximate depth. One could use these components to build a feature vector in this six-dimensional space.

However, looking at typical objects of interest in our tele-immersive system, we can make the following three observations: (a) thermal information is typically a more reliable feature for discriminating humans from the background, (b) visible and color information is typically a more reliable feature for detecting inanimate objects at room temperature, and (c) using pixel-level information is more difficult than using higher level features for classifying objects of interest. For example, in the computer display scenario presented in Figure 3.4, it would be extremely difficult to classify the pixels of the display correctly without having more context. Thus, instead of focusing on the pixel level, we will continue analyzing higher level features.

Our approach in this module will be to use the candidate regions S^{vis} and S^{ir} that are now aligned with the color image reference frame. In general, we will gather statistics for each region, such as size, average grayscale distance from background model, average color, average temperatue difference from thermal background, etc. However, again, we note that thermal information leads to stronger features, compared to other visible wavelength features such as average color. In other words, we want to allow human subjects to wear a wide variety of clothing, and note that despite the choice of clothing, their thermal characteristics will remain stable.

Human Region Feature Extraction and Classification

In our prototype system, we assume that thermal information is the primary component necessary for human object detection. Thus, we will first analyze regions derived from the thermal image, and determine which regions S_i^{ir} represent human objects. For each of these regions, we define a feature vector $\mathbf{v}_i^{ir} \in \mathbb{R}^2$ composed of: region size and average temperature difference from thermal background. We found that these features are sufficient to distinguish between human subjects and other warm, but inanimate, objects typically found in indoor environments, such as computers, computer monitors, and lamps.

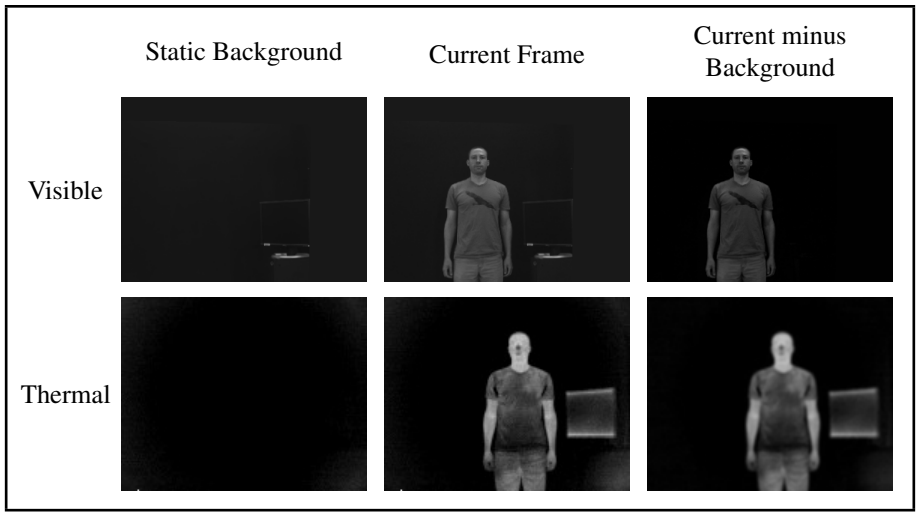


Figure 4.3: Background objects with changing temperatures: In this scene, a computer display was turned on in the middle of system operation. Thus, after background subtraction in the thermal frame, both the person and the display remain. Thus, higher level features such as shape and size are required to correctly classify the foreground (person) and the background (computer display).

This ability to distinguish between people and other warm objects is important because, similar to pixels changing due to object motion or illumination changes in the visible wavelengths, an object changing temperature survive the filtering of thermal background subtraction. Figure 4.3 demonstrates this scenario – when the background model is created, a computer was turned off. Later in the sequence, however, the computer and it's display have been turned on. Thus, in the thermal image, both the human subject and the computer monitor cause large temperature differences with respect to the background. This scenario requires that we be able to distinguish thermal regions using features other than temperature above the background.

In order to classify the feature vector $\mathbf{v}_i^{ir}$, we compare it to a model that represents the nominal "appearance" in this feature space of a human object: $\mathbf{v}_{model}^{ir}$. In our prototype system, we used empirical knowledge (obtained by qualitatively studying images from typical tele-immersive scenes, see Figure 4.4) to define this model:

$$\mathbf{v}_{model}^{ir} = \begin{bmatrix} 11520 \\ 84 \end{bmatrix} \tag{4.27}$$

Notice that the components of this vector represent different types of information: area and temperature. Area is specified in units of pixels, which can range from 1 pixel to the size of the image, $320 \times 240 = 76,800$ pixels. Temperatures, however, are specified in units of digital numbers (DN), which span the 14-bit image value range: 0 - $16,384$ DN. Thus, in practice we normalize each of these values: for area, we divide the measured blob area by the total image size; and for temperature, we divide the

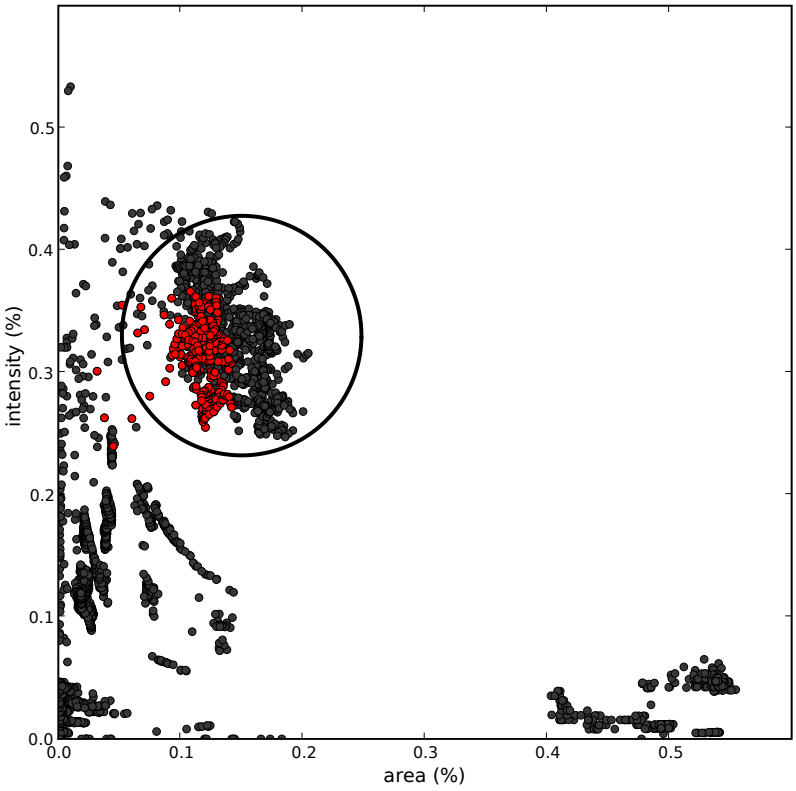


Figure 4.4: Thermal region features: This plot shows the computed thermal region features, normalized area and normalized intensity, from 2300 images collected during several experiments described in this thesis. The red (light) points represent images where we know there was only one warm human subject in the camera field of view. The dark circle represents the decision boundary used in thermal object-of-interest detection.

measured blob temperature by 255. This results in the normalized model:

$$\mathbf{v}_{model}^{ir} = \begin{bmatrix} 0.15 \\ 0.33 \end{bmatrix} \tag{4.28}$$

We then classify these features based on euclidean distance in the normalized feature space. If the features are within τ^{ir} of the model, we classify the corresponding region as foreground:

$$S_i^{ir} = \begin{cases} \text{foreground} & \text{if} \quad \|v_i^{ir} - v_{model}^{ir}\| < \tau^{ir} \\ \text{background} & \text{otherwise} \end{cases} \tag{4.29}$$

From qualitative analysis of test data (see Figure 4.4), we found that $\tau^{ir} = 0.09$ leads to good classification results. Note that the five red points outside of the decision

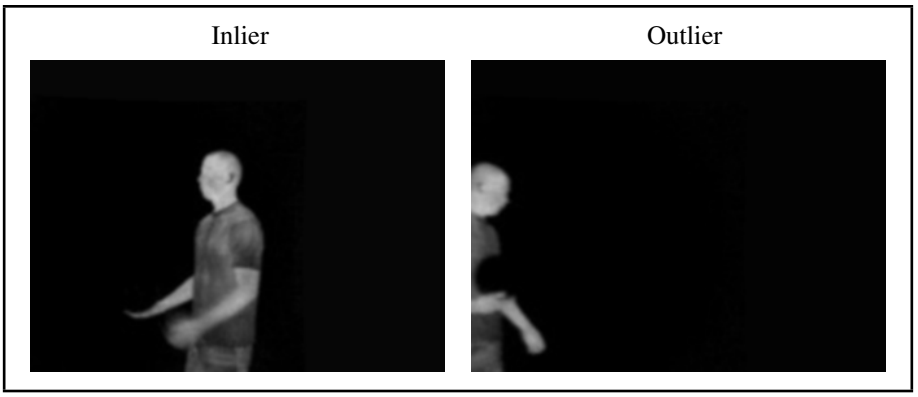


Figure 4.5: Inliers and outliers in feature space verification: The inlier sample shown on the left represents one of the many red circles which lie inside the decision boundary circle in 4.4. The outlier shown on the right represents one of the five red points which fall outside of the decision boundary.

boundary in Figure 4.4 are examples of when the human subject was only partially in the camera field of view. Thus, we treat these points as outliers. The rest of the red points represent a human subject fully within the camera field of view. See Figure 4.5 for a comparison of a typical inlier and a typical outlier.

Note that the decision boundary for this classifier is essentially a ball of radius τ^{ir} centered on $\mathbf{v}^{ir}_{model}$. Figure 4.4 shows thermal regions from 2300 images collected during our experiments projected into this feature space. Two immediate conclusions can be drawn from this plot. First, the simple feature space chosen leads to significant separation between different classes (human regions vs. non-human region). Second, despite the practical success of this feature space, we can see that our euclidean distance metric does not capture some of the structure that is present in the data. While this has worked well in our testing, note that it does not necessarily represent the *best* decision boundary. In particular, the two pieces of information in this feature space might not have the same relevance. For example, people in the scene might regularly appear the nominal size, but vary more widely in temperature.

Also, we note that these measurements can be used in a Bayesian classification framework. Using a technique similar to that in [19], shown above in Equations 4.8 and 4.9, we could estimate the probability that the measurement $\mathbf{v}^{ir}_i$ represents a human. If we learn, or subjectively select a prior, then the standard tools of Bayesian estimation can be used. Further, if we tracked objects over time, we could use recursive Bayesian techniques such as Kalman filtering. These avenues of development are extremely interesting, and left for future work.

Now that we have thermal regions classified as human foreground, we can utilize this information during classification of inanimate objects of interest (in our prototype, spheres).

Inanimate Object Feature Extraction and Classification

After thermal region classification, human regions are subtracted from the preliminary regions S_i^{vis} derived from the original grayscale image. This leaves regions which are *not* human foreground regions, and which should be further searched for inanimate objects of interest.

In general, these regions will be noisy, may contain holes, and will not correspond to edges or homogeneous regions in the color image. Thus, in order to extract inanimate objects of interest, we will search for high level features such as shapes or regions with distinct colors.

In our prototype system (where we are considering spherical objects), we use the Hough transform to find circles in the regions. The Hough transform is a general technique for detecting parameterized features in images. The simplest form of the Hough Transform, the Hough line transform, is commonly used to detect lines and line segments in images. The Hough line transform utilizes the normal form parameterization of a line:

$$x \cos\theta + y \sin\theta = r \tag{4.30}$$

For each point (x, y) on a given line, r and θ are constant.

The Hough line transform works by finding every possible structure that could contain a given point (x, y). The sample point (x, y) is often obtained by first finding edge features in the original image. In practice, one discretizes the line space (r, θ) before searching for possible structures. For example, one could choose to search for lines that have angles discretized in chunks of 10°. For each sample point (x, y), the Hough transform iterates through all possible angles, solving for r using Equation 4.30. Then, the evidence for a line parameterized by the resulting (r, θ) is incremented. Thus, an accumulator will be keeping track of "votes" for a particular (r, θ) location in the line space. After operating on every sample point, we can simply search the accumulator image for local maxima. These will be (r, θ) locations that represent lines passing through many sample points.

Similarly, circles can be detected using the Hough transform technique by using a parameterization for circular structures instead of lines:

$$(x - a)^2 + (y - b)^2 = r^2 \tag{4.31}$$

Here, r is the radius and (a, b) is the center of the circle containing (x, y). In this case, the Hough feature space is three dimensional, thus, the accumulator can be represented by a 3-dimensional image. In our prototype system, we utilize the HoughCircles function of the OpenCV computer vision library. This function utilizes the 21HT algorithm described in [50]. In this algorithm, edge direction information is used to reduce the storage and computational demands of the transform. This is achieved by decomposing the problem into two stages. The first stage is designed to find candidates for circle

centers using a 2-dimensional Hough Transform. The second stage is a 1-dimensional Hough transform to determine radii for candidate centers from the previous stage. Edge information is used by observing that the center of a circle must lie along the gradient direction of each edge point on the circle. Thus the common intersection of these gradients defines the location of the circle center. In the transform, a 2D accumulator builds evidence of centers, and local peak detection selects candidate center locations. The second stage of this method uses the candidate centers from the first stage and Equation 4.31 to solve for the radius of a edge point (x, y). These radii are used to construct a histogram of possible radius values for the given center candidate (a, b). Peaks in the radius histogram indicate the presence of circles.

Once we have these circular areas, we compute feature vectors containing statistics regarding those areas. Thus, for each circular structure C_j^{vis}, we create a feature vector $\mathbf{v}_j^{vis}$. In our prototype system, we create a simple feature vector composed solely of the ratio of the number of pixels in the intersection of S_i^{vis} and C_j^{vis} to the number of pixels in C_j^{vis}:

$$\mathbf{v}_j^{vis} = \frac{S_i^{vis} \cap C_j^{vis}}{C_j^{vis}} \tag{4.32}$$

In effect, this feature vector tells us how many pixels in the circular area C_j^{vis} have changed with respect to the background. As this ratio approaches 1, we can be confident that not only do we have a circular area, but also an area that has exhibited some motion with respect to the background. Note, however, that this general framework could also utilize features such as average color, average depth, or more detailed features such as color or depth histograms, entropy, texture, etc.

The use of this type of feature led us to the following classifier:

$$S_i^{vis} = \begin{cases} \text{foreground} & \text{if} \quad \mathbf{v}_i^{vis} > \tau^{vis} \\ \text{background} & \text{otherwise} \end{cases} \tag{4.33}$$

where τ^{vis} is a threshold determining how many pixels in the circular region must have exhibited change with respect to the background.

The decision boundary here is point in the interval $[0, 1]$. This type of decision boundary is appropriate here because deviations in this feature can be intuitively modeled as zero mean gaussian noise. The choice of τ^{vis} reflects the magnitude of the noise expected in the preliminary region detection. Note that this type of classifier also naturally rejects outliers resulting from errors in the Hough transform output. In our prototype system, we found that $\tau^{vis} = 0.7$ allowed robust circular foreground object detection.

Fusion Logic

At this point, we have two sources of regions classified as foreground. A decision fusion module uses these groups of regions to create a final foreground mask. Currently,

this consists of taking the union of regions classified as foregrounds in the previous feature extraction and classification component.

Another way to approach this task is to return to the pixel level information available at this point. All of the previous modules can be seen as having grouped pixels into membership classes. For example, the pixel location (i, j) can belong to a warm region, a circular region, a red region, or combinations of these. A more general (and sophisticated) decision fusion module could use this information to form a final foreground mask that utilizes more complicated (e.g. hierarchical) object models. For example, at this point the decision fusion could pick out objects that are circular but not green.

In general, however, this framework allows room for developing more sophisticated decision fusion modules. For example, one could attempt to extract high level context given the spatial relationship (or time history) of the two groups of foreground regions.

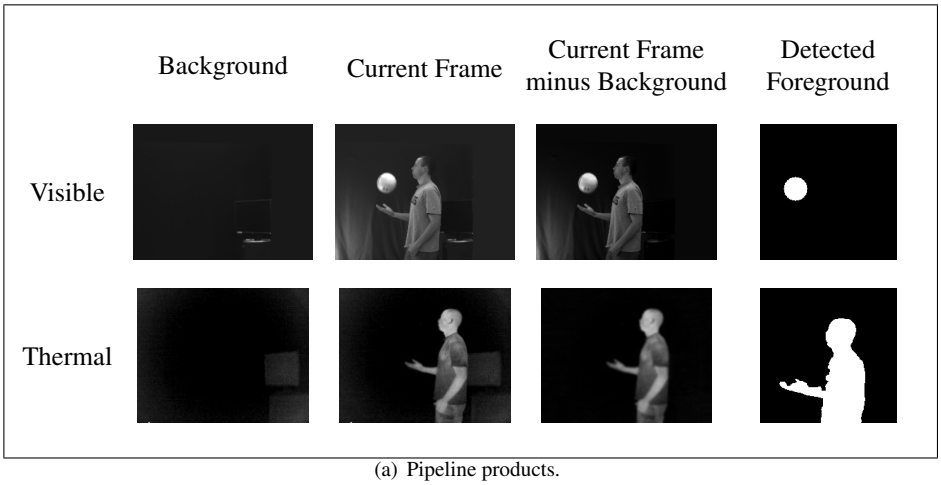
Chapter 5

Experimental Results

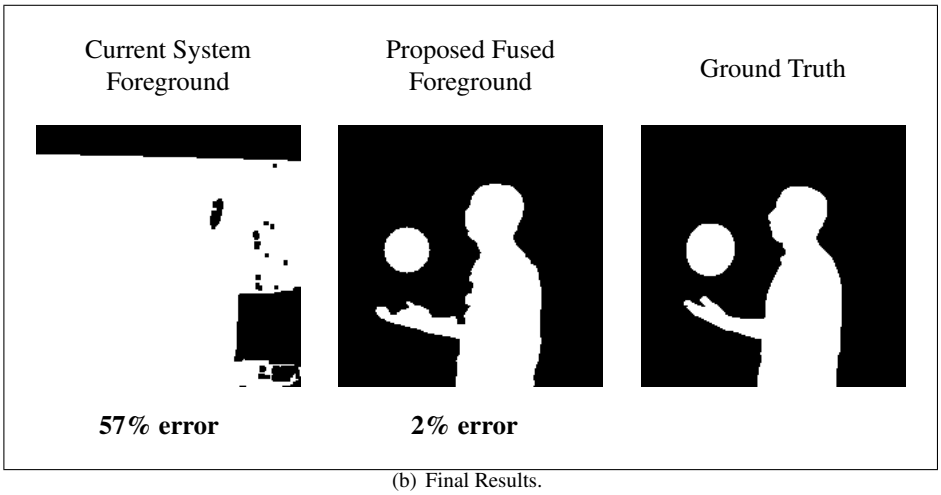
This chapter describes how our fusion algorithm was tested and validated for each problem presented in Section 3.4. These results demonstrate that visible and thermal fusion provide robust foreground detection in the presence of changing illumination (Section 5.1), moving foreground objects (Section 5.2), moving background objects (Section 5.3), lack of contrast between foreground and background objects (Section 5.4), and lack of contrast between different foreground objects (Section 5.5). Three additional experiments (Section 5.6) show that our algorithm maintains good performance both in an ideal scene and in scenes with multiple objects of interest. For all of these experiments, ground truth was collected by manually labeling parts of the image as foreground. Finally, we summarize these results and discuss their implications (Section 5.7).

5.1 Changing Illumination

This section deals with one of the most common challenges in computer vision: time-varying illumination. The problems that changing illumination can cause are described in Figure 3.2. In this scenario, normal room lights were on during the creation of the background image. Then, in the middle of system operation, an extra lamp was turned on. Problems occur because the new lighting conditions lead to changes in the background (and foreground) objects which are not modeled in the static background. Dynamically updating the background model would provide some robustness to gradual changes in room lighting. However, as mentioned in Section 4.2, this technique is not well suited for tele-immersive environments. Figure 5.1 shows the results of our system under this scenario. Thermal imagery is not sensitive to the lighitng change and is able to detect the person in the scene. Also, because our inanimate object detection emphasizes higher level features, in this case shape, the ball is correctly identified as an object of interest.



(a) Pipeline products.



(b) Final Results.

Figure 5.1: Changing illumination experimental results: This figure shows our results (compared to the existing system performance) in the presence of the problem described in Figure 3.2, where an additional lamp is turned on in the middle of the experiment. This experiment demonstrates that our thermal and visible image fusion exhibits robustness to lighting changes in the scene.

5.2 Moving Foreground Objects Casting Shadows

Moving foreground objects typically cast shadows on the background. This scenario, and it's effect on the TEEVE system are shown in Figure 3.3. Characteristics of the background can make the effect of these shadows more or less pronounced. In the ideal environment, the background would be dark, uniform, and non-reflective. Shadows cast on such a surface would not have much of an effect in the measured images. However, this background rarely occurs in real indoor environments. There are often large objects, such as lamps, doors, and cubicles that give a complex structure to the background. Further, there are often posters, pictures, or other materials that lead to textures in the background. These common background environments lead to more pronounced effects of shadowing. A complex 3D background will exhibit broken and non-uniform shadows. On the other hand, a colored and finely textured background will also cause problems during shadowing.

Again, because thermal imaging is not sensitive to visible reflectance and shading, the IR image is not effected by the shadows. Figure 5.2 shows that when the current TEEVE system breaks down because of shadows on a complex background, thermal images provide a stable modality for foreground detection.

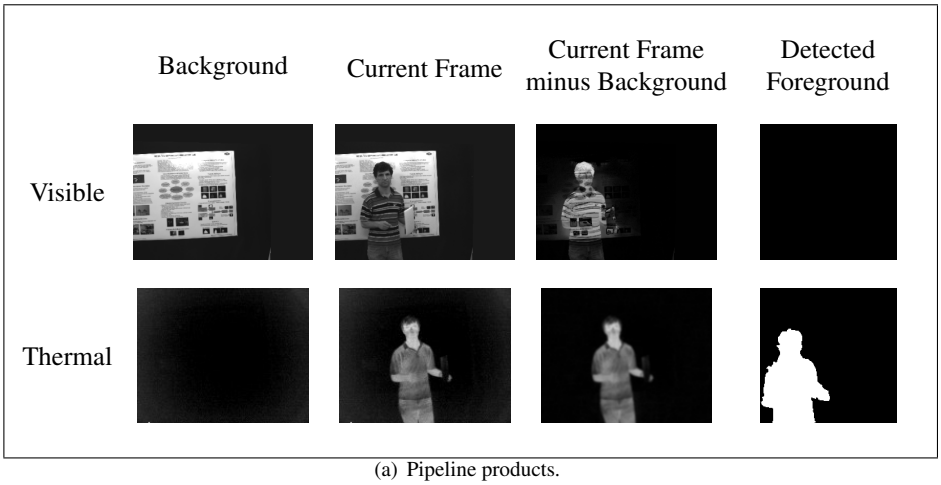
5.3 Moving Background Objects

Background objects can often move, or change appearance. For example, books or phones can be moved, or computer monitors can display changing content. The latter scenario is the focus of our experiment, and is described in Figure 3.4.

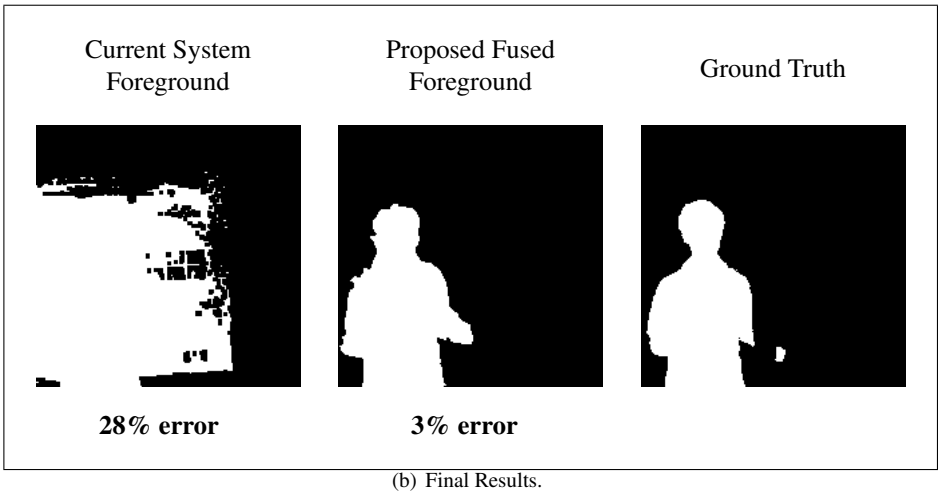
This scenario is particularly important in tele-immersive systems. These systems often deal with users viewing many different computer displays, which can be quite large (Figure 2.2). Even aside from these large displays, a normal office environment may have many such semi-static objects: desktop displays, windows, etc.

This scenario is also interesting because it illustrates a particular class of background object: a semi-static object. We call this a semi-static object because different portions of the same object can be changing or remaining fixed. For example, the outer casing of the monitor remains fixed (and value in the visible image does not change over time). However, the screen portion of the monitor is not static – in fact, it is likely to be changing rapidly over time. In the thermal image, a similar situation can arise. In this case, a portion of an object could be changing temperature over time, while other parts remain constant.

In our experiment, we show a computer display that has changed appearance between the acquisition of the static background and the current frame. The results shown in



(a) Pipeline products.



(b) Final Results.

Figure 5.2: Moving foreground object experimental results: This experiment demonstrates our results in the presence of the problem described in Figure 3.3, where shadows are cast on the background due to the foreground object's motion in the scene. Room lighting remained constant during this experiment, and the change of background pixel intensity is solely due to the human foreground blocking illumination. This experiment illustrates that visible and thermal fusion is robust to shadowing on textured backgrounds.

Figure 5.3 demonstrate that our fusion algorithm can filter out these moving background objects.

5.4 Low Contrast Between Foreground and Background

This experiment deals with the problem presented in Figure 3.5, where portions of the foreground object appear visually similar to the background.

This experiment illustrates the fact that when one attempts to control an environments background, for example, by painting it blue or black or using a curtain, then the users of that space will not be able to wear clothing of similar colors. Doing so would make them look more like the background.

In general, this problem can arise in uncontrolled environments as well. However, in an uncontrolled environment, it is less likely that large portions of a person's appearance will match the background.

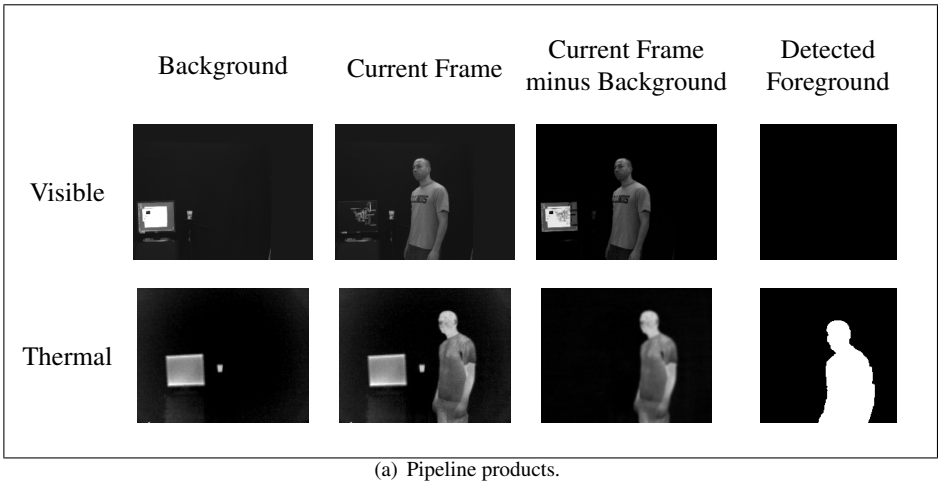
The results of this experiment are shown in Figure 5.4. These results illustrate that the color of the clothing a person wears is independent of their temperature profile. Thus, even in the presence of a lack of visual contrast, thermal imagery provides stable information for human foreground detection.

5.5 Low Contrast Between Different Foreground Objects

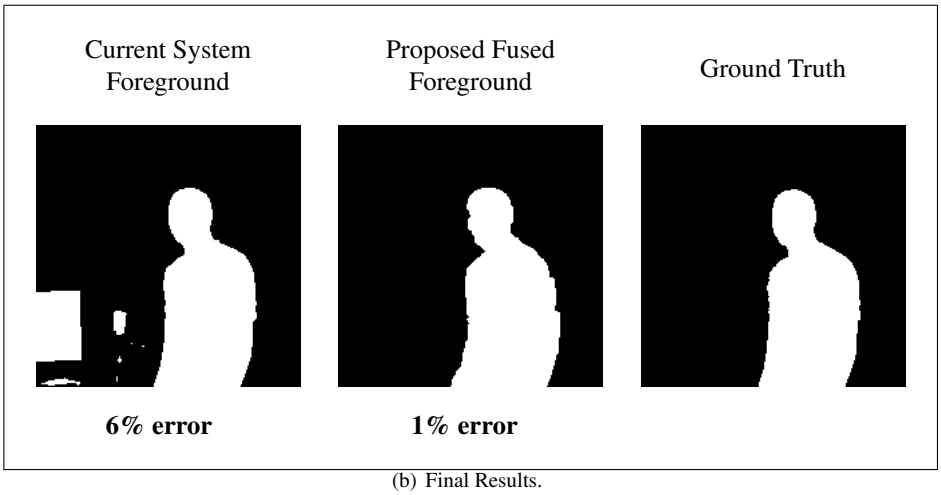
A critical element of our fusion framework is that it allows high level features to be used to describe foreground objects. This ability is important in tele-immersive systems because it gives us finer control over what objects are reconstructed in the virtual world.

When multiple objects have a similar visual (and thermal) brightness or color, we must use higher level features such as shape or size to distinguish them. This problem is depicted in Figure 3.6. As described in Section 3.1, our prototype system assumes that the objects being manipulated in the tele-immersive system are spherical.

Figure 5.5 demonstrates the results of this experiment. These results show that the object detection in the visible wavelengths successfully extract spherical regions from the rest of the potential foreground regions.

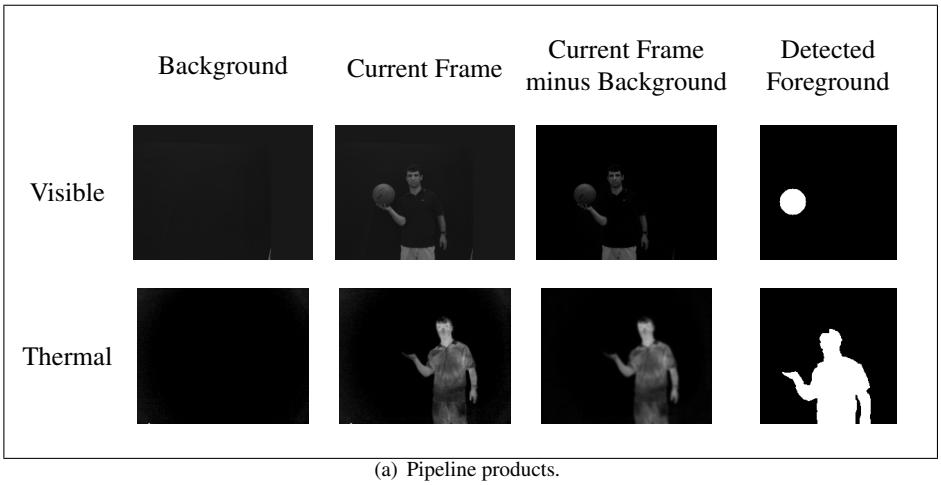


(a) Pipeline products.

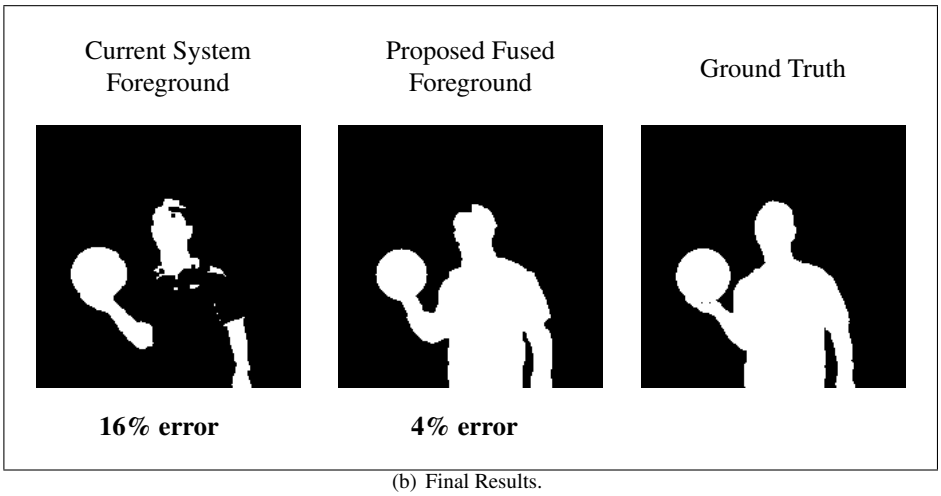


(b) Final Results.

Figure 5.3: Moving background object experimental results: This experiment demonstrates our algorithm's ability to filter out moving background objects (problem described in Figure 3.4). We capture a scene in which a computer display is changing over time. This is a case where we want the display to remain part of the background despite the fact that all of its pixels are changing with respect to the static background model. Our fusion algorithm successfully handles this scene, and does not include the computer display in the foreground.

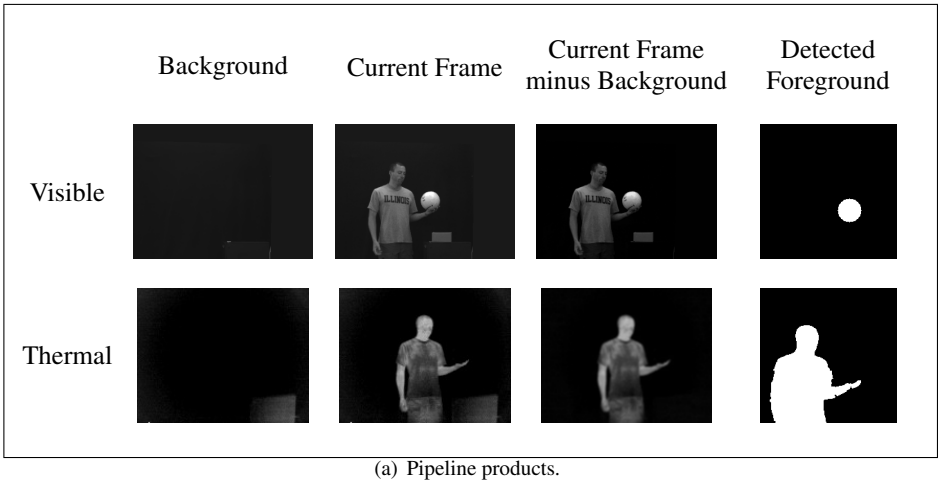


(a) Pipeline products.

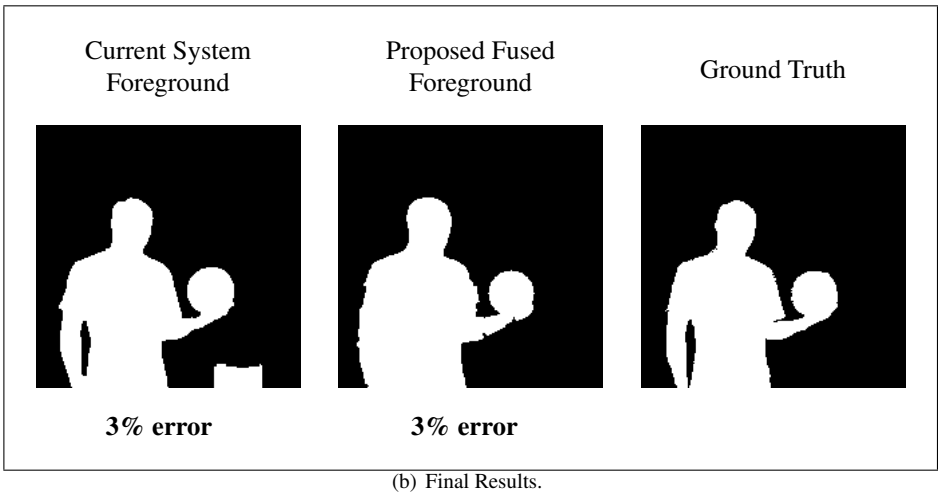


(b) Final Results.

Figure 5.4: Low contrast between foreground/background experimental results: This experiment demonstrates our fusion algorithm performance in the presence of problems that can occur when the foreground object has a similar color or brightness as the background (described in Figure 3.5). In this case, the current system fails to recognize portions of the person as foreground because of their dark clothing. Fusion with thermal information is able to fill in the missing information.



(a) Pipeline products.



(b) Final Results.

Figure 5.5: Low contrast between different foreground objects results: This experiment demonstrates the ability of our algorithm to distinguish between two objects with similar brightness and color: a soccer ball and a box (problem description in Figure 3.6). While the total errors are similar between the current TEEVE system and our proposed method, the qualitative improvement is quite noticeable.

5.6 Other Experiments

In this section, we present a few other experiments that, while they do not directly relate to the specific problems that this thesis focuses on solving, illustrate interesting results.

Ideal Scene

In this experiment, we created scene that attempts to reach the ideal – containing a dark, static background, no illumination changes, etc. This scenario represents the type of scene which leads to the best results in the current TEEVE system. Figure 5.6 compares the results of the current TEEVE system and our fusion algorithm. These results show that both algorithms perform roughly the same, with the TEEVE system giving slightly cleaner results.

Multiple Objects of Interest in Scene

In many human foreground detection and tracking systems, multiple people cause difficulties because of occlusion and complex interaction.

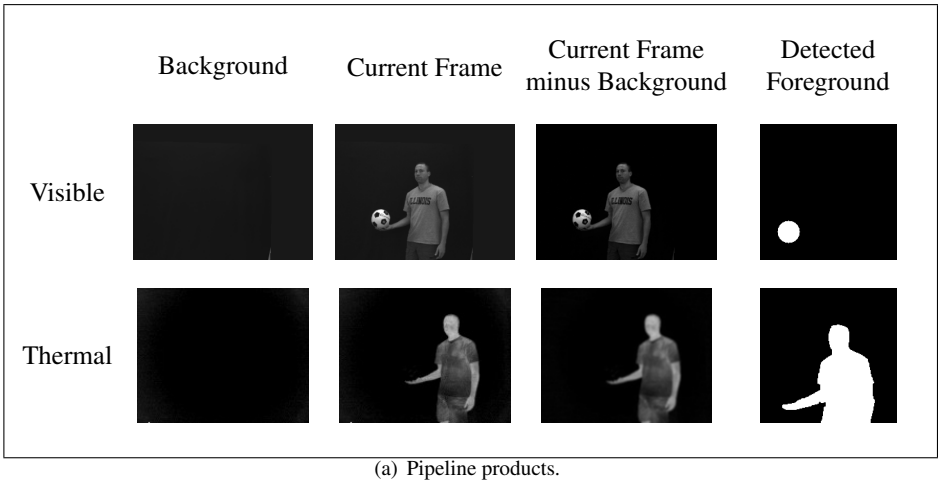
Our system currently does not attempt to distinguish between each individual (even if we might have separate connected components representing these individuals), and only attempts to maintain a general classification of foreground vs. background. Figure 5.7 demonstrates that our system can successfully extract multiple people in the scene. Similarly, Figure 5.8 demonstrates that our system can also extract multiple inanimate objects of interest.

5.7 Summary and Discussion

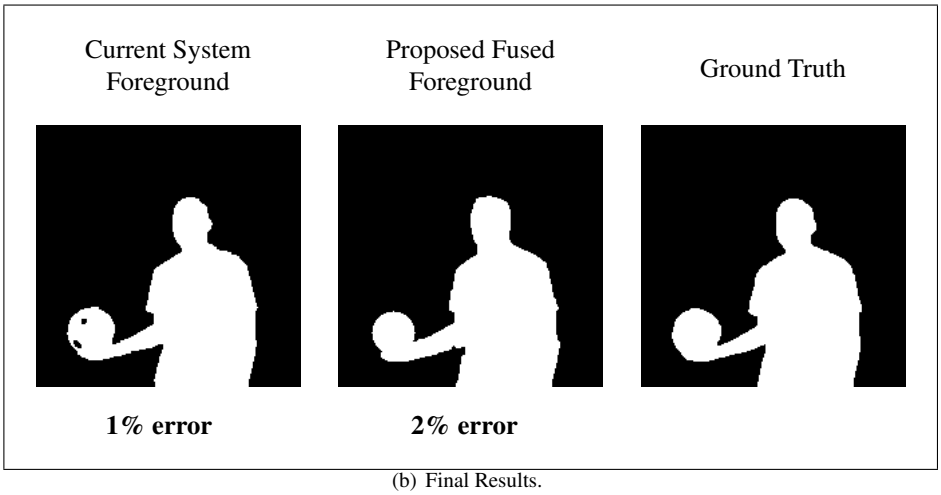
Table 5.1 summarizes the quantitative results of our fusion algorithm in comparison to the existing TEEVE system.

We can see that, for all of the problematic scenarios that we focused on in this thesis, our proposed fusion algorithm greatly improves foreground detection accuracy. This is primarily due to the fact that most of the problem scenarios we focused on are primarily due to factors that cause detrimental effects in the visible wavelengths. We assumed that the temperature distribution of the indoor environment is much more benign, allowing us to emphasize the thermal information observed in the scene.

One interesting observation about these results is there is a low, but consistent, level of error in the fused foreground. From looking at the foreground mask images, we can see that this is caused by errors along the boundary of the person in the scene. We believe that four factors contribute to this. First and most prominent is the fact

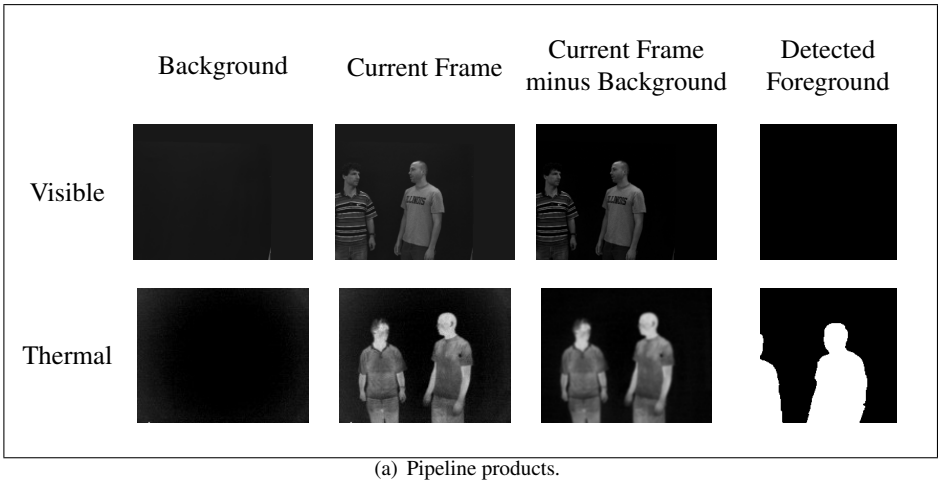


(a) Pipeline products.

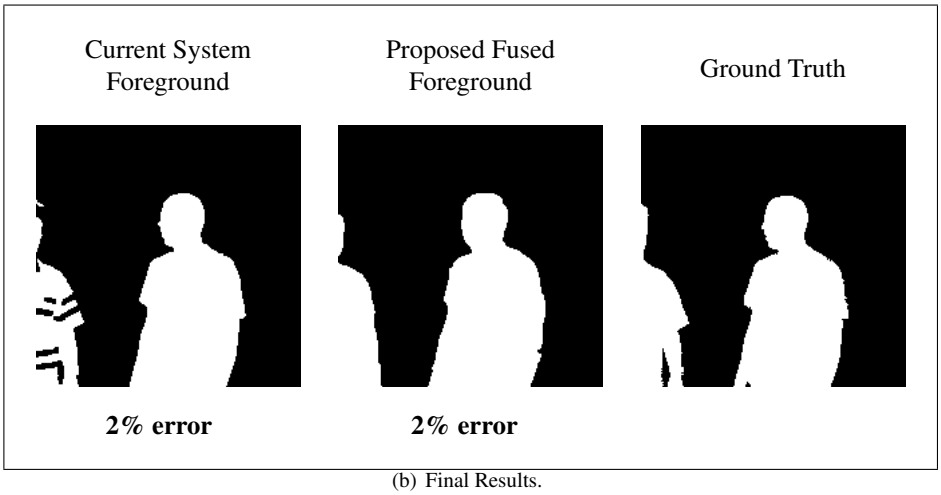


(b) Final Results.

Figure 5.6: Ideal scene experimental results: This experiment demonstrates a scene that approaches the ideal (see Section 3.2. Given such a scene, the current TEEVE system performs quite well.

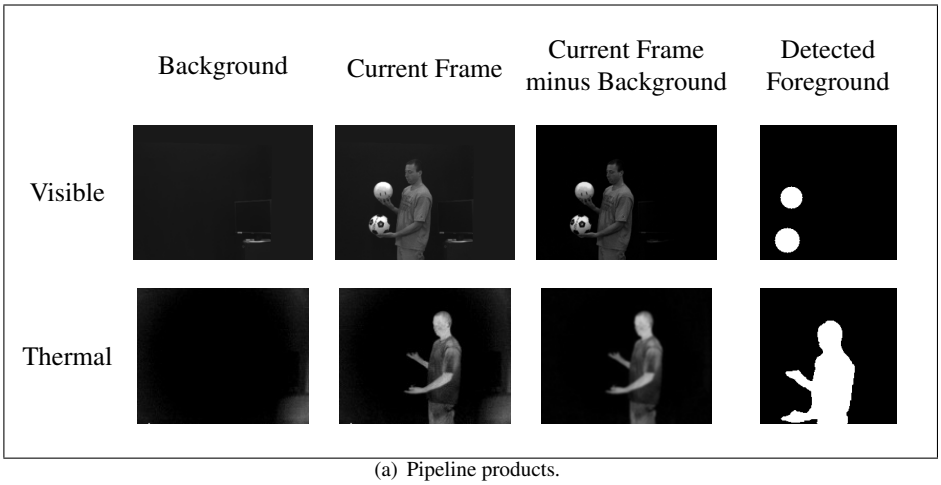


(a) Pipeline products.

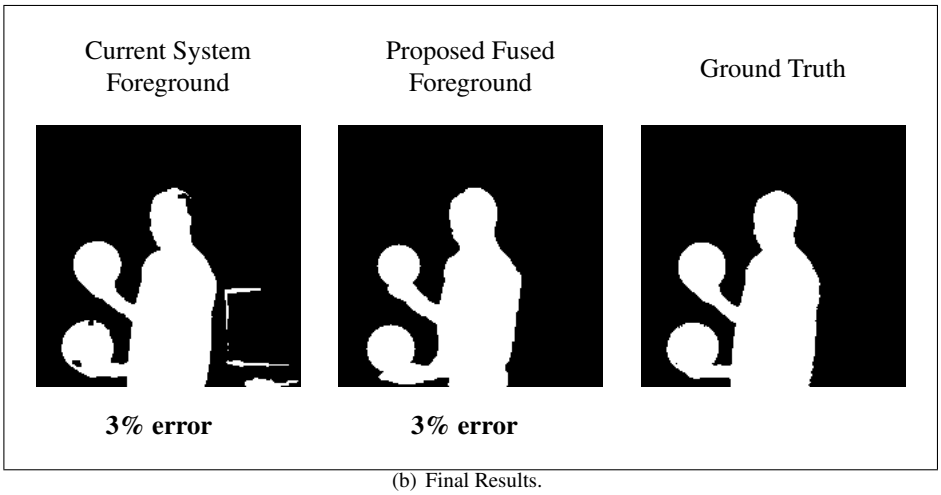


(b) Final Results.

Figure 5.7: Multiple subject scene experimental results: This experiment demonstrates that multiple objects of interest can be detected in the scene. Again, the background is approaching the ideal, so both the existing TEEVE system and the proposed fusion system perform quite well.



(a) Pipeline products.



(b) Final Results.

Figure 5.8: Multiple spheres experimental results: This experiment demonstrates that multiple inanimate objects of interest (in this case, spheres) can be detected in the scene. Again, the background is approaching the ideal, so both the existing TEEVE system and the proposed fusion system perform quite well.

Experiment	Method	F Neg	F Pos	Total	% err
Changing Illumination	TEEVE	141	21939	22080	57
	Fusion	566	420	986	2
Moving foreground casting shadows	TEEVE	0	10674	10674	28
	Fusion	244	876	1120	3
Moving background	TEEVE	122	2205	2327	6
	Fusion	117	424	541	1
Low contrast between foreground and background	TEEVE	6056	209	6265	16
	Fusion	741	647	1388	4
Low contrast between different foreground objects	TEEVE	74	1127	1201	3
	Fusion	206	1010	1216	3
Ideal scene	TEEVE	297	175	472	1
	Fusion	359	344	703	2
Multiple people in scene	TEEVE	567	384	951	2
	Fusion	330	590	920	2
Multiple spheres in scene	TEEVE	359	698	1057	3
	Fusion	677	601	1278	3

Table 5.1: Table of quantitative results, comparing performance of current TEEVE system and our proposed fusion algorithm. "F Neg" represents the number of pixels that were incorrectly classified as background (i.e. the false negative detections). "F Pos" represents the number of pixels that were incorrectly classified as foreground (i.e. the false positive detections). The "Total" is the sum of these two pixel counts, and the percent error represents the percentage of the image that was misclassified.

that we are aligning the thermal region using estimated depth values. This estimation utilizes depth from the previous time step stereo correlation to estimate the depth of the blob in the current image. Because the foreground object is typically moving, there small errors are introduced by using these old depth values. Second, the difference in position and orientation of the thermal and visible image mean that different parts of the foreground object are observed by each camera. Third, the low resolution of the thermal camera means that when we upsample thermal blobs when transforming them into the visible frame, we magnify any small errors in preliminary thermal blobs. Fourth, the thermal and visible images are only loosely synchronized. Synchronization errors cause a moving foreground object to be in slightly different poses between the thermal and visible image.

Despite these small errors along the boundary, we believe that the our fusion method will lead to a much improved experience within the immersive virtual environment.

Chapter 6

Conclusions

This thesis presented a method of extracting objects of interest from 2D images in order to synthesize a tele-immersive virtual environment. Because tele-immersive systems are meant to facilitate interaction between real people in the real world, the central objects of interest are these people, the things they jointly manipulate, and the tools they need to perform this manipulation.

Our method added thermal infrared imagery to existing tele-immersive platforms currently developed by the Department of Computer Science at the University of Illinois at Urbana-Champaign, UC Berkeley, and NCSA. Estimates of the 3D structure of objects were used to combine information from visible and thermal cameras in a common reference frame. Subsets of visible and thermal images were analyzed in this multi-dimensional feature space in order to construct the foreground of the scene. This framework was designed to use motion, depth, color, and temperature (as well as region-based characteristics of these layers such as moments, entropy, and texture) to detect people and other objects of interest.

We implemented and tested our fusion algorithm in a prototype hardware system. Our algorithm performed well across a number of experiments for which foreground detection is typically hard.

This work serves as an introduction to the potential of multi-sensor fusion in the domain of tele-immersive systems. The main contributions of this thesis are:

1. Calibration of visible and infrared cameras: we extended a state of the art automatic multi-camera calibration technique [41] to simultaneously calibrate grayscale, color, and thermal cameras using a flashlight
2. Development of methodology for fusing visible and infrared images based on tele-immersive system scene modeling and estimation of scene 3D structure
3. Building prototype hardware to acquire visible and thermal IR imagery, and the design of off-line processing and analysis algorithms
4. Quantitative analysis of the TEEVE system with and without fusion of visible and infrared information

6.1 Future Work

The development of our prototype system will continue such that robust, real-time operation of the extended TEEVE system is achieved. Hardware and networking challenges will be encountered here, although we believe that the speed of local networks will allow us to integrate the thermal camera into the current TEEVE architecture. This integration will also motivate the development of software based synchronization techniques, which will have two benefits. First, digital video cameras without hardware synchronization capabilities will become available for use in tele-immersive systems. Second, software synchronization will remove the need for complex external circuitry currently required, making tele-immersive systems more portable and reconfigurable.

Further, we have identified a number of avenues for further research:

1. In this thesis we used manual inspection of experimental data to define object models. This process would become more adaptive and robust by using semi-supervised machine learning techniques to learn different classes of object models. The difficulty here is how to allow the user to provide feedback to the learning algorithm in an efficient way.

2. Instead of throwing out information about foreground regions between each time step, object tracking can be used to more accurately initialize the fusion algorithm. Object tracking would also lead to more accurate depth estimations by minimizing the chances that a depth representing background will be assigned to a foreground object.

3. We have implemented a prototype that detects simple (convex) objects. However, we are also interested in detecting more complicated objects (for example, wheelchairs). The simple feature space described in this thesis would not lead to robust detection of such complicated objects. However, it seems possible that more complicated features (such as a motion similarity and proximity to a human region) have interesting potential.

4. We are interested in developing an interface that would allow a tele-immersive user to train the system to perform finely tuned foreground object detection. This will be a useful capability because of the wide range of potential usage patterns in tele-immersive systems. For example, one could train the system to understand that there should only be one person in the immersive environment. Thus, even if an office mate occasionally moves into the camera field of view, the system will not treat that second person as a foreground object.

We believe that further research in these areas can dramatically improve the immersive experience in state-of-the-art tele-immersive systems, and are excited about what lies ahead.

Appendix

Optimizing Region Alignment

The steps above assume that the (dense) depth of blobs in the thermal image is correct. However, it is really only an approximate depth. This will introduce error in the projection of the foreground masks into the color image reference frame.

To refine this region registration, features can be matched between the projected thermal image and the underlying color image. This is a challenging problem because of the different features available in each imaging modality. Three general techniques have potential for effectively addressing this task: active contours, contour matching, and mutual information.

An active contour is an elastic curve that is formed by deforming an initial guess according to an energy function. In our application, active contours can be used to warp the foreground region to better fit underlying gradients in the color image. This is an optimization problem which can be solved using a number of standard techniques such as dynamic programming [2], variational calculus [21], or various gradient descent approaches [46]. The energy functional is usually decomposed into three parts: internal, image, and external energy. The internal energy component determines how much the active contour can stretch and bend. The image component is usually influenced by edge or corner features in the image data. The external energy can represent external constraints, or other information based on external information.

Let a contour be represented by a sequence of points $v = \{v_1, \ldots, v_N\}$, where $s = 1 \ldots N$ indexes points along the contour, and $v_s = (x_s, y_s)$ are the image coordinates of point s. An energy functional which only incorporates internal and image energies can be defined as:

$$E = \sum_{s=1}^{N} \alpha(s)E_{cont} + \beta(s)E_{curv} - \gamma(s)E_{image} \tag{.1}$$

$$E_{cont} = \|v_s - v_{s-1}\| - \bar{d} \tag{.2}$$

$$E_{curv} = \|v_{s-1} - 2v_s + v_{s+1}\|^2 \tag{.3}$$

$$E_{image} = \|\nabla I\,(x(s), y(s))\| \tag{.4}$$

where $\bar{d}$ is the average euclidean distance of the $N-1$ contour edges defined by the N

contour points. E_{cont} and E_{curv} represent the internal energy. E_{cont} helps maintain contour point spacing, while E_{curv} represents a curvature constraint. E_{image} is the gradient magnitude.

Similarly contours, or contour segments can be matched between the projected thermal image and the color image. This could be done using chain code representation of contours.

Contours are often represented by an integer sequence depending on some property of the curve, such as the discretized angle between one point along the curve and the next [27],[13]. This is referred to as the chain-code of the curve. For example, a contour with n points could be represented by the sequence $\{a\} = \{a_1, a_2, \ldots, a_n\}$, with $a_i \in \{1, 2, \ldots, 7\}$. In this case, one unit in the integer sequence represents an angle of 45°. Various modifications can be made to this basic chain code representation to handle integer wrap-around, noise, and variations in rotation and scale between curves (see [27] for details).

In order to match contours (or other open or closed curves), a similarity function is needed. The correlation measure presented in [27] can be used to determine matching contours. In this case, let contour A be represented by a scale and rotation invariant chain code $\{a_i\}$, and let B be represented by a similarly invariant chain code $\{b_k\}$. Let the longer contour (for example, A) be resampled (or otherwise normalized) to be the same length, N, as B. Now the correlation measures is defined as:

$$D_{kl} = \frac{1}{N} \sum_{j=0}^{N-1} \cos \frac{\pi}{4} (a_{k+j} - b_{l+j}) \tag{.5}$$

where k and l represent the starting index along contour A and B, respectively. From this, a similarity function can be devised:

$$S_{AB} = \max_{l \in M} (D_{kl}) \tag{.6}$$

where M specifies the various starting locations for contour B.

Finally, at a coarser scale, mutual information techniques can be used to refine the foreground projection.

Mutual information was pioneered by both Viola and Collignon [44], [11]. For two images A and B, mutual information I can be defined as

$$I(A, B) = H(B) - H(B|A) \tag{.7}$$

where $H(B)$ is the Shannon entropy of image B, and $H(B|A)$ is the conditional entropy, which is based on the probability of gray values b in image B given that the corresponding location in A has gray value a. In other words, the mutual information encodes how much information A contains about B.

The Shannon entropy of an event is defined as [40]:

$$H = \sum_i p_i \log \frac{1}{p_i} = -\sum_i p_i \log p_i \tag{.8}$$

Image entropy can be computed by focusing on the distribution of gray values in the image. A probability distribution over gray values can be computed by computing a histogram of gray value counts and normalizing by the total number of pixels. Note that local spatial structure and information is not represented by the image entropy. Again, if an image contains little information (consists of similar gray values), the entropy will be low, while an image with large quantities of very different intensities results in a high entropy.

Another way to formulate the mutual information is:

$$I(A, B) = H(A) + H(B) - H(A, B) \tag{.9}$$

This formulation, equivalent to Equation .7, shows that maximizing the mutual information can be seen as minimizing the joint entropy of A and B. Mutual information proves to be more robust than using only joint entropy in cases where only small parts of the two images actually overlap [38].

A third form of mutual information is analogous to the Kullback-Leibler measure:

$$I(A, B) = \sum_{a,b} p(a, b) \log \frac{p(a, b)}{p(a)p(b)} \tag{.10}$$

This formulation measures the distance between the joint distribution of the images' gray values $p(a, b)$ and the joint distribution in the case of independence of the images, $p(a)p(b)$ [38].

While all of this discussion has been based around images A and B, it is often the case that some pre-processing is performed before performing mutual information registration. For example, regions of interest can be specified by a user in a semiautomatic registration system [10].

While we did not implement these techniques in our prototype system, we believe that they will lead to more accurate foreground extraction, and will utilize them in future work.

References

[1] Besma Abidi, Shafik Huq, and Mongi Abidi. Fusion of visual, thermal, and range as a solution to illumination and pose restrictions in face recognition. In *International Carnahan Conference on Security Technology*, pages 325–330, 2004.

[2] A. A. Amini, S. Tehrani, and T. E. Weymouth. Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints. In *International Conference on Computer Vision*, pages 95–99, 1988.

[3] Peter Bajcsy. Gridline: automatic grid alignment dna microarray scans. *IEEE Transactions on Image Processing*, 13(1):15–25, 2004.

[4] Peter Bajcsy, Rob Kooper, David Scherba, and Martin Urban. Toward hazard aware spaces: Knowing where, when and what hazards occur. Technical Report isda06-002, National Center for Supercomputing Applications, 2006.

[5] Ruzena Bajcsy, Reyes Enciso, Gerda Kamberova, Lucien Nocera, and Radim Šára. 3d reconstruction of environments for virtual collaboration. In *IEEE Workshop on Applications of Computer Vision*, pages 160–167, 1998.

[6] H. Harlyn Baker, Donald Tanguay, Irwin Sobel, Dan Gelb, Michael E. Goss, W. Bruce Culbertson, and Thomas Malzbender. The coliseum immersive teleconferencing system. Technical Report HPL-2002-351, HP Mobile and Media Systems Laboratory, 2002.

[7] George Bebis, Aglika Gyaourova, Saurabh Singh, and Ioannis Pavlidis. Face recognition by fusing thermal infrared and visible imagery. *Image and Vision Computing*, 24(7):727–742, 2006.

[8] M. Bertozzi, A. Broggi, A. Lasagni, and M. Del Rose. Infrared stereo vision-based pedestrian detection. In *unkown*, 2005.

[9] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.

[10] Andres Carrillo, Jeffrey L. Duerk, Jonathan S. Lewin, and David L. Wilson. Semiautomatic 3-d image registration as applied to interventional mri liver cancer treatment. *IEEE Transactions on Medical Imaging*, 19(3):175–185, 2000.

[11] Andre Collignon. *Multi-modality medical image registration by maximization of mutual information*. PhD thesis, Catholic University of Leuven, 1998.

[12] Kevin Michael Curry. Supporting collaborative awareness in tele-immersion. Master's thesis, Virginia Polytechnic and State University, 1999.

[13] Xiaolong Dai and Siamak Khorram. Feature-based image registration algorithm using improved chain-code representation combined with invariant moments. *IEEE Transactions on Geoscience and Remote Sensing*, 37(5):2351–2362, 1999.

[14] Kostas Daniilidis, Jane Mulligan, R. McKendall, D. Schmid, Gerda Kamberova, and Ruzena Bajcsy. *The Confluence of Computer Graphics and Vision*, chapter Real-time 3D-Tele-immersion. Kluwer Academic Publishers, 2000.

[15] Thomas A. DeFanti, Jason Leigh, Maxine D. Brown, Daniel J. Sandin, Oliver Yu, Chong Zhang, Rajvikram Singh, Eric He, Javid Alimohideen, Naveen K. Krishnaprasad, Larry Smarr, Mark Ellisman, Phil Papadopoulos, Andrew Chien, John Orcutt, Robert Grossman, and Marco Mazzucco. Teleimmersion and visualization with the optiputer. In *International Conference on Artificial Reality and Telexistence*, 2002.

[16] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.

[17] Aglika Gyaourova, George Bebis, and Ioannis Pavlidis. Fusion of infrared and visible images for face recognition. In *European Conference on Computer Vision*, volume IV, pages 456–468, 2004.

[18] David L. Hall and James LLinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.

[19] Ju Han and Bir Bhanu. Detecting moving humans using color and infrared video. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 228–233, 2003.

[20] Lijun Jiang, Feng Tian, Lim Ee Shen, Shiqian Wu, Susu Yao, Zhongkang Lu, and Lijun Xu. Perceptual-based fusion of ir and visual images for human detection. In *International Symposium on Intelligent Multimedia, Video and Speech Processing*, pages 514–517, 2004.

[21] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. In *International Journal of Computer Vision*, volume 1, pages 321–331, 1988.

[22] Nikhil Kelshikar, Xenophon Zabulis, Jane Mulligan, Kostas Daniilidis, Vivek Sawant, Sudipta Sinha, Travis Sparks, Scott Larsen, Herman Towles, Ketan Mayer-Patel, Henry Fuchs, John Urbanic, Kathy Benninger, Raghurama Reddy, and Gwendolyn Huntoon. Real-time terascale implementation of tele-immersion. In *International Conference on Computational Science*, 2003.

[23] Seong G. Kong, Jingu Heo, Faysal Boughorbel, Yue Zheng, Besma R. Abidi, Andreas Koschan, Mingshong Yi, and Mongi A . Abidi. Multiscale fusion of visible and thermal ir images for illumination-invariant face recognition. In *International Journal of Computer Vision*, 2007.

[24] Praveen Kuman, Jay C. Alameda, Peter Bajcsy, Mike Folk, and Momcilo Markus. *Hydroinformatics: Data Integrative Approaches in Computation, Analysis, and Modeling*. CRC Press, 2006.

[25] Sang-Chul Lee and Peter Bajcsy. Three-dimensional volume reconstruction based on trajectory fusion from confocal laser scanning microscope images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2221–2228, 2006.

[26] Jason Leigh, Thomas A. DeFanti, Andrew E. Johnson, Maxine D. Brown, and Daniel J. Sandin. Global tele-immersion: Better than being there. In *7th International Conference on Artificial Reality and Tele-Existence*, 1997.

[27] Hui Li, B. S. Manjunath, and Sanjit K. Mitra. A contour-based approach to multisensor image registration. *IEEE Transactions on Image Processing*, 4(1):320–334, 1995.

[28] Yi Ma, Stefano Soatto, Jana Košecká, and S. Shankar Sastry. *An Invitation to 3-D Vision*. Springer, 2004.

[29] Fernando C. M. Martins, Brian R. Nickerson, Vareck Bostrom, and Rajeeb Hazra. Implementation of a real-time foreground/background segmentation system on the intel architecture. In *IEEE ICCV99 Frame Rate Workshop*, 1999.

[30] J. Michel, N. Nandhakumar, and V. Velten. Thermophysical algebraic invariants from infrared imagery for object recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19, pages 41–51, 1997.

[31] Jane Mulligan and Kostas Daniilidis. Real time trinocular stereo for tele-immersion. In *IEEE International Conference on Image Processing*, 2001.

[32] Jane Mulligan, Volkan Isler, and Kostas Daniilidis. Trinocular stereo: a real-time algorithm and its evaluation. In *IEEE Workshop on Stereo and Multi-Baseline Vision*, 2001.

[33] N. Nandhakumar, V. Velten, and Jonathan Michel. Thermophysical affine invariants from ir imagery for object recognition. In *Workshop on Physics-Based Modeling in Computer Vision*, pages 48–54, 1995.

[34] P.J Narayanan, Peter Rander, and Takeo Kanade. Constructing virtual worlds using dense stereo. In *Proceedings of the Sixth IEEE International Conference on ComputerVision (ICCV'98)*, pages 3 – 10, January 1998.

[35] Kyoung S. Park, Yong J. Cho, Naveen K. Krishnaprasad, Chris Scharver, Michael J. Lewis, Jason Leigh, and Andrew E. Johnson. Cavernsoft g2: A toolkit for high performance tele-immersive collaboration. In *ACM Symposium on Virtual Reality Software and Technology*, pages 8–15, 2000.

[36] James R. Parker. *Algorithms For Image Processing and Computer Vision*. John Wiley & Sons, 1997.

[37] Ioannis Pavlidis, James Levine, and Paulette Baukol. Thermal imaging for anxiety detection. In *IEEE Workshop on Computer Vision Beyond the Visible Spectrum: Methods and Applications*, pages 104–109, 2000.

[38] Josien P. W. Pluim, Antoine Maintz, and Max A. Viergever. Mutual-information-based registration of medical images: A survey. *IEEE Transactions on Medical Imaging*, 22(8):986–1004, 2003.

[39] Surya Prakash, Pei Yean Lee, Terry Caelli, and Tim Raupach. Robust thermal camera calibration and 3d mapping of object surface temperatures. In Jonathan J. Miles, G. Raymond Peacock, and Kathryn M. Knettel, editors, *SPIE Proceedings: ThermoSense XXVIII*, volume 6205, page 62050J, 2006.

[40] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423/623–656, 1948.

[41] Tomáš Svoboda, Daniel Martinec, and Tomáš Pajdla. A convenient multi-camera self-calibration for virtual environments. *PRESENCE: Teleoperators and Virtual Environments*, 14(4):407–422, August 2005.

[42] Roger Y. Tsai. A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, 1987.

[43] Radim Šára, Ruzena Bajcsy, Gerda Kamberova, and Raymond A. McKendall. 3-d data acquisition and interpretation for virtual reality and telepresence. In *IEEE and ATR Workshop on Computer Vision for Reality Based Human Communications*, pages 88–93, 1998.

[44] Paul Viola and William M. Wells III. Alignment by maximization of mutual information. In *International Conference on Computer Vision*, pages 16–23, 1995.

[45] Jian-Gang Wang, Eric Sung, and Ronda Venkateswarlu. Registration of infrared and visible-spectrum imagery for face recognition. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 638–644, 2004.

[46] Donna J. Williams and Mubarak Shah. A fast algorithm for active contours. In *International Conference on Computer Vision*, 1990.

[47] Lawrence B. Wolff, Diego A. Socolinsky, and Christopher K. Eveland. *Computer Vision Beyond the Visible Spectrum*, chapter Face Recognition in the Thermal Infrared. Springer, 2005.

[48] Tao Yang, Stan Z. Li, Quan Pan, and Jing Li. Real-time and accurate segmentation of moving objects in dynamic scene. In *ACM International Workshop on Video Surveillance and Sensor Networks*, 2004.

[49] Zhenyu Yang, Yi Cui, Bin Yu, Jin Liang, Klara Nahrstedt, Sang-Hack Jung, and Ruzena Bajcsy. TEEVE: The next generation architecture for tele-immersive environments. In *International Symposium on Multimedia*, 2005.

[50] H. K. Yuen, J. Princen, J. Illingworth, and J. Kittler. Comparative study of hough transform methods for circle finding. *Image and Vision Computing*, 8(1):71–77, 1990.

[51] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.