A SYSTEM FOR CHANGE DETECTION AND HUMAN RECOGNITION IN VOXEL SPACE USING STEREO VISION

A Dissertation presented to the Faculty of the Graduate School

University of Missouri

In Partial Fulfillment

Of the Requirements for the Degree

Doctor of Philosophy

---

by

ROBERT H. LUKE III

Dr. James M. Keller, Dissertation Supervisor

July 2010

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled

A SYSTEM FOR CHANGE DETECTION AND HUMAN RECOGNITION IN VOXEL SPACE USING STEREO VISION

Presented by Robert Luke

A candidate for the degree of Doctor of Philosophy

And hereby certify that in their opinion it is worthy of acceptance.

Professor James M. Keller

Professor Marjorie Skubic

Professor Curt Davis

Professor Mihail Popescu

Professor Guilherme DeSouza

Thank you to my sister, my father and my mother.

Merrily, merrily, merrily, merrily,…

ACKNOWLEDGEMENTS

To my advisor Jim Keller I am eternally grateful. Not only has he been a constant source of information and guidance over the better part of my collegiate years, but he also gave me the interest and confidence to continue my education to a post graduate degree. Through him I have been introduced to a worldwide network of researchers that will be lifelong colleagues throughout my career. Jim's personality has aided in making my college education the greatest experience of my life.

My deepest gratitude goes to Derek Anderson for his help and friendship in my graduate years. Countless research topics have come from our conversations over coffee, a beer or a cigar. Derek and his family have become an extension of my own and I thank them for allowing me in their lives.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

Table Page

ABSTRACT

Image space change detection algorithms do not adequately address the complexities in real-world dynamic environments. With few exceptions, these algorithms rely on pixel-level information to detect proper foreground change. They have difficulty adapting the background model when objects are moved or when lighting changes abruptly. This dissertation proposes a world space system to detect change in the living quarters of a single elderly person in an assisted living community. Using stereo vision, this method discretizes the living space into volume elements (voxels) and determines the configuration of the scene using stereo vision. Voxel representations of the scene built over time are used to determine change. Further processing of this voxel change space using segmentation, shape and color determines the presence and location of humans. Experiments demonstrate the success of this change detection procedure in a range of real-world, dynamic test situations.

1. INTRODUCTION

1.1. Motivation

Video surveillance is a vital activity for security in a number of locations including airports, banks, military installations and government buildings. The desire for greater security has become even more prevalent since the September 11th terrorist attack on the United States of America. Surveillance generally consists of face recognition in crowded settings, so as to protect against a specific person; or abnormal behavior detection, such as individuals acting in known strange fashions, fighting, etc.

In contrast, the research described in this dissertation focuses on the surveillance of a single person in a home setting for their own well-being. Elderly people living alone have a set of vulnerabilities that can be ameliorated by a surveillance system. The most common fear for elders is falling and being incapacitated until being found by someone. A surveillance system could recognize a fall and notify a caregiver. A less obvious change in wellbeing is related to mental health. A system that recognizes and tracks human activity could also notify a caregiver when behavioral changes occur.

High-level computer vision systems performing human activity analysis must be provided stable and reliable information regarding the whereabouts of people. No present human segmentation system can perform robustly for long periods of time in a loosely constrained indoor environment. The system defined in this dissertation is

designed to reliably segment humans and other large objects that the human interacts with in an unconstrained indoor environment based on stereovision. It then outputs the three-dimensional segmentation of the person.

1.2. Overview

Three-dimensional space can be represented with discrete non-overlapping cubes known as volume elements (voxels). This system takes two sets of stereo pair images as input and builds a voxel representation of the human in the scene as output. Each stereo pair initially builds a separate three-dimensional voxel representation of the scene. By placing the stereo cameras at opposite sides of the room, the amount of the scene hidden from the cameras' representations is minimized. Therefore, a more accurate model of the scene is created when their data are fused.

A voxel representation of the scene is built at each time step. Voxels representing static objects will be occupied most of the time. We define the background model as the voxels occupied by nonhuman objects. Therefore, a rolling window of the past several time steps are held in memory. If a voxel is frequently occupied in the previous time steps, it is likely occupied by a static object, and is therefore added to the background. This function is performed on all voxels in the scene to create a background model.

Given a new voxel scene representation, voxels that are currently occupied, but are vacant in the background model, are classified as foreground voxels. These foreground

voxels represent the change from the background model. These change voxels are part of a moving human, a moving nonhuman object, or error due to incorrect stereo matching and voxel representation. Foreground false alarms due to error are usually small, erratic volumes scattered throughout the scene.

It is possible that multiple volumes are associated with change at a given moment, such as when a person moves a chair and continues walking. It is therefore required that all changed volumes are represented as unique segments. Because foreground false alarms are generally small, segments smaller than a given volume are rejected. Remaining segments are either humans or nonhuman moving objects.

Of the foreground segments, we assume that at most only one is a human. Each segment is therefore classified as human or nonhuman. A segment is classified as a human if a head can be found in the segment. The human head is a salient object because it is unique when compared to most objects in a common living area. To find the head, a three-dimensional kernel is convolved over each segment to find volumes that have a shape like the top of a head. Then, using the three-dimensional segments and the original two-dimensional images, voxels that represent skin color are labeled. Fusing the volumes that represent skin and head shape returns mostly true positives, but can also return some false alarms such as lamp shades. Hence, only head returns above 1.25 meters are kept as true positives.

The segment intersecting with the head is classified as human. All voxels contained in this segment are highlighted and recorded. As well, a color histogram is built for this

human segment for tracking when the head cannot be found. Finally, the human segment can be removed from the current time frame's representation and the current scene model can be used to update the background model.

1.3. Novel Research Contributions

- Real-time voxel environment modeling based on multiple stereo pairs

Using two stereo cameras, this system creates a three dimensional voxel model for an entire scene. The scene is over six meters on a side and the model can be created and updated in real time. Nonhuman static objects in the scene are stored in a background model, but the the voxel location of the human subject is highlighted.

This is a paradigm shift from the popular 2D models. Image space algorithms manipulate relatively low level information such as pixels, pixel areas and image space illumination. It is my belief that the current accepted lines of research in the change detection community will never approach the capabilities of a human. I believe this is because the information gathered from images as well as the algorithms used to detect change assume non-realistic models of scenes and cannot accurately model the complexities of this problem.

- Three-dimensional voxel-based video change detection using stereo vision, which is highly robust to significant image-space weaknesses, e.g. illumination changes and shadows.

Image space models do not intrinsically handle drastic changes in lighting. All of these models instead have a component that tries to appropriately handle these lighting changes. Some adapt the background model while others build several models over a wide range of lighting possibilities and choose the one most appropriate to the current setting. Both of these approaches have significant shortcomings.

Adaptive models such as [**1**] will absorb a stationary human into the background unless another component is used to track the human. If the human is tracked, a drastic change in lighting will likely drastically change his or her appearance and will likely be lost for a short time. Adaptive models also have the extra variable related to adaptation speed. This relates how significant new frames are to the update of the background model. A quick adaptation will successfully update the background model during significant changes in lighting, but will also adapt a stationary person into the background faster. A slow adaptation will allow people to temporarily stop moving, but significant changes in lighting will take too long to absorb into the background model. Therefore, the adaptation rate must be chosen for the specific setting.

As well, algorithms that try to model many unique lighting possibilities are limited. The number of lighting combinations grows exponentially with the number of light sources. Hence, this requires many models to accurately perform change detection. Some

algorithms such as Eigen backgrounds [**2**] use this lighting model successfully, but do not have a mechanism to add new lighting scenarios. Given that light sources can move or be added, this is a significant shortcoming. Only recently have researchers tried to add adaptation to Eigen Backgrounds [**3**,**4**,**5**].

In contrast to these image space modeling techniques, depth values created using stereo vision are robust to changes in lighting. Even image space algorithms using depth have shown reliable results with respect to lighting change [**6**]. Extending the use of depth to a full three-dimensional model further resolves the accuracy of scene modeling by allowing the fusion of multiple stereo pairs.

- Novel classifier based on skin color, head shape and height that operates on voxel segmented islands for human identification and tracking.

There is an entire field of research related to human identification from video. Face detection is the most prevalent of these techniques, [**7**,**8**,**9**]. Some of these approaches are accurate enough to be implemented in consumer digital cameras to automatically set the exposure levels. Unfortunately, the majority of these techniques require the face to be aimed at the camera and have a large number of pixels over the face area. Neither of these attributes will be guaranteed in our setting.

Human face/head detection herein uses skin color on the face retrieved from image space, and head shape and height retrieved from voxel space. All three of these

features are readily available and the fusion of their information reliably finds the head of any standing person in any orientation.

- Natural way to address tracking and updating, i.e. not adapting stationary humans into the background while still absorbing moved objects even under significant lighting variations.

Due to the height constraint, faces will not be found when the person is sitting. As well, the face detector will periodically fail while the person is standing. Therefore, a mechanism is developed to describe and track the color associated with the human in the scene. This tracking technique is computationally inexpensive to build and run, making it a good candidate for a real-time solution. Assuming that there will only be a single person in the scene at a time and that there are very few nonhuman objects moving through the scene at a given time, this method reliably tracks a previously detected person.

Knowing the location of the human at all times simplifies the modeling of the background scene. The person can be removed from the model of the scene at a given time step, resulting in only nonhuman objects in the model. The background model is then updated at each time step using the current model of the nonhuman scene. In this manner, we are guaranteed to never adapt the person into the background model. This is especially helpful when the person is sitting in a chair or on a couch.

- False alarm reduction for change detection regarding movement of nonhuman objects (e.g. chairs) and objects manipulated by a person.

All image space change detection algorithms output any change from the background model whether it is human or nonhuman. Many higher level systems, including human activity recognition, require only human change detection and segmentation. Without a reliable human detection mechanism, there is no way to reject nonhuman change.

Because our system is able to track the person at all times, nonhuman objects are never classified as change if they are moved. An example of this is when a person moves a chair. Both the human and the chair have moved, but because the chair can be differentiated from the person, it can be removed from the change detection output.

2. BACKGROUND

Previous research in automated video surveillance has produced a plethora of background modeling algorithms. This chapter will include a brief description of the most common background modeling algorithms in 2D and 3D. These systems operate using a range of input data including grayscale, color and pixel depth. As well, this chapter will describe a set of algorithms to perform three-dimensional segmentation of objects.

2.1. Image Space Algorithms

A large number of background adaptation and foreground segmentation algorithms have been proposed. A short list of the most valuable includes Temporal Difference, Mean and Threshold, Gaussian Mixture Model [**1**], Eigen Backgrounds [**2**], and Wallflower [**10**]. This section describes each as well as their strengths and weaknesses.

2.1.1. Temporal Difference

Arguably the simplest background-subtraction procedure is temporal difference. The difference between two consecutive images at each pixel is computed. Any pixel differences larger than a pre-defined threshold are considered foreground,

$$f_t(x,y) = |I_t(x,y) - I_{t-1}(x,y)| > \tau,$$

where $I_t(x, y)$ and $I_{t-1}(x, y)$ are the pixel intensities at pixel (x, y) in image I at times t and t-1 respectively, $f_t(x, y) \in [0,1]$, and τ is the scalar threshold value.

This algorithm works very poorly for most situations, due to factors such as:

- Only objects moving at a sufficient speed with adequate feature, color or intensity, values are detected.
- A moving object is labeled as foreground twice according to the current frame and the previous as in figure 2.1.
- Decisions are at a pixel level and no region or spatial coherency is incorporated.
- Shadows and specular highlights are detected, and all moving objects are discovered, i.e. there is nothing specific to the tracking of humans.

Figure 2.1: An example of the double foreground phenomenon produced by temporal differencing. The person is in the current frame as well as the previous. His locations are then classified as change between the frames.

2.1.2. Mean and Threshold

Another simple background-subtraction algorithm, a logical extension to temporal differencing, is the mean and threshold method. During a training phase, in which the human is assumed to not be in the scene, the mean value is found for each pixel over a set of images. In future images, pixels with a difference greater than a given threshold from the mean are considered foreground. The algorithm can be made adaptive by "alpha updating" the mean and threshold with values from new images at runtime.

First, the mean value is found for each pixel using a training set of images,

$$b(x,y) = \frac{1}{N}\sum_{t=1}^{N} I_t(x,y).$$

Given a new image at time t, the foreground is computed as the pixels having difference from the mean values greater than a predefined threshold,

$$f_t(x,y) = \begin{cases} 1 & if\ |I_t(x,y) - b_t(x,y)| > \tau \\ 0 & else \end{cases}$$

The background can then be "alpha updated" with the pixel values that are not labeled foreground,

$$b_{t+1}(x,y) = \begin{cases} b_t(x,y) & if\ \ f_t(x,y) = 1 \\ (1-\alpha)b_t(x,y) + \alpha I_t(x,y) & else \end{cases}$$

The algorithm also leads to poor results in real-world settings, for reasons such as:

- While the mean and threshold allow one to model subtle variability, most of which is pixel jitter, this approach is insufficient for modeling abrupt illumination changes.

- All moving objects are detected, not just humans.
- For the static version, initialization with the human not in the scene is required and there is no graceful way to recover from many forms of incorrect adaptation. For the updating version, objects such as the human are adapted into the background if they remain still.
- Shadows and specular highlights still exist, and again, only subtle illumination changes are gracefully dealt with.

Also, it is naïve to believe that a pixel will have only a single value for all lighting situations, artificial or natural. If the values are adapted over time, the algorithm can accommodate changes in lighting, but this takes considerable time and will classify a great number of pixels incorrectly.

2.1.3. Gaussian Mixture Model

The Gaussian Mixture Model (GMM) [**1**], models each pixel as a set of Gaussian distributions. A given pixel can have a range of values representing its background state due to pixel jitter or periodic movements of objects. A mixture of Gaussians robustly models a pixel's intensity or color properties by taking into account its wider range of possible values. Foreground pixels are then defined as being greater than a fixed number of standard deviations from the mean of all Gaussian models.

Updating a GMM is vital for a robust system, but it can be difficult to find the set of parameters that properly update the properties of each pixel's models. A fixed number of Gaussian distributions, K, are used to model each pixel, usually 3-5. Each of these models has a probability related to the number of occurrences in the data thus far.

$$P\left(\vec{I}_t(x,y)\right) = \sum_{j=1}^{K} \omega_{j,(x,y)} * \eta(\vec{I}_t(x,y), \vec{\mu}_{j,(x,y)}, \Sigma_{j,(x,y)})$$

$\vec{\mu}_{j,(x,y)}$ and $\Sigma_{j,(x,y)}$ are the mean and covariance matrix of a normal distribution η,

$$\eta\left(\vec{I}_t(x,y), \vec{\mu}_{j,(x,y)}, \Sigma_{j,(x,y)}\right) = \left(\frac{1}{(2\pi)^{D/2}\left|\Sigma_{j,(x,y)}\right|^{1/2}}\right) e^{-\frac{1}{2}\left(\vec{I}_t(x,y)-\vec{\mu}_{j,(x,y)}\right)^T \Sigma_{j,(x,y)}{}^{-1}\left(\vec{I}_t(x,y)-\vec{\mu}_{j,(x,y)}\right)},$$

Where D is the feature dimensionality, $\omega_{j,(x,y)}$ is the jth mixture weight (the portion of the data accounted for by this Gaussian), and the weights are subject to the constraint that

$$\sum_{j=1}^{K} \omega_{j,(x,y)} = 1.$$

A user specified variable T, $0 \leq T \leq 1$, defines the percentage of input data that should belong to a background model. Therefore, B of the K mixtures, $1 \leq B \leq K$, model the background at any given moment. For computational simplicity, the authors of [**1**] assume that the distributions are represented as $\Sigma_{j,(x,y)} = \sigma^2_{j,(x,y)} I$, where I is the identity matrix.

The value of a pixel in a new image is tested against each Gaussian distribution for that pixel. Before selecting the set of mixtures that make up the background, the models are

sorted according to $\omega_{j,(x,y)}/\sigma_{j,(x,y)}$, or in the case of a multidimensional $\vec{\sigma}^2_{j,(x,y)}$, the models are sorted according to $\omega_{j,(x,y)}/\left\|\vec{\sigma}_{j,(x,y)}\right\|$. This value increases both as the variance decreases and the mixtures weight increases. The number of background mixtures, $B(x,y)$, is therefore

$$B(x,y) = argmin_b\left(\sum_{k=1}^{b}\omega_{k,(x,y)} > T\right).$$

If the value is within a user specified number of standard deviations, usually 2-3, from the mean of a distribution, that pixel is considered background. The parameters and probability associated with the winning distribution are "alpha and rho updated" with the current pixel's value.

$$\omega_{j,(x,y),t} = (1-\alpha)\omega_{j,(x,y),t-1} + \alpha\left(M_{j,(x,y),t}\right),$$

$$\vec{\mu}_{j,(x,y),t} = (1-\rho_{(x,y)})\vec{\mu}_{j,(x,y),t-1} + \rho_{(x,y)}\vec{I}_t(x,y),$$

$$\vec{\sigma}^2_{j,t} = \left(1-\rho_{(x,y)}\right)\vec{\sigma}^2_{j,t-1} + \rho_{(x,y)}\left(\vec{I}_t(x,y) - \vec{\mu}_{j,(x,y),t-1}\right)^T\left(\vec{I}_t(x,y) - \vec{\mu}_{j,(x,y),t-1}\right),$$

$$\rho_{(x,y)} = \alpha\eta\left(\vec{I}_t(x,y), \vec{\mu}_{j,(x,y),t-1}, \Sigma_{j,(x,y),t-1}\right).$$

The variable α, $0 \leq \alpha \leq 1$, is a user specified learning rate, $\omega_{j,(x,y),t}$ is the jth mixture weight at time t, $\omega_{j,(x,y),t-1}$ is the jth mixture weight at the previous time step, $M_{j,(x,y),t}$ is a function that is 1 for the matched mixture, and 0 otherwise.

If the pixel does not match any model, a new model is created. The distribution with the smallest probability is chosen for replacement by the new pixel. The mean of the

new distribution is set to the pixel's value, $\vec{\mu}_{j,(x,y),t} = \vec{I}_t(x,y)$, and the standard deviation is set to a predefined value.

The GMM has become one of the most widely used algorithms for background modeling. This is probably because of its simplicity and general accuracy in real-world problems. The algorithm has a modest computational expense and can therefore be implemented on computers using limited memory and processing capabilities. The algorithm also performs well in outdoor environments where objects such as trees carry out cyclical motion due to wind.

The shortcomings of this algorithm include:

- It only performs pixel level change detection.
- The segmentation procedure is not specific to the tracking of humans.
- The initialization and recovery from a corrupt background model are the same as in the single mean and threshold procedure.
- Shadows and specular highlights cause problems.
- Only gradual illumination changes are gracefully handled.

2.1.4. Gaussian Mixture Model Extensions

Several researchers have continued the work of Stauffer and Grimson [**1**] and extended the GMM. This section describes two notable extensions proposed by Li [**11**] and

Zivkovic [**12**]. These researchers have made their implementations public, and these programs are used in the experiments of section 7.1.

Li [**11**] proposes the use of Bayesian statistics to classify a pixel as foreground or background. He begins by building a large feature vector at each pixel representing a wide range of possible feature values. His conjecture is that the background pixels will undergo minimal change due to pixel jitter, cyclical motion, or lighting change. It is assumed that a background pixel will undergo minimal or at most cyclical change; therefore, only a small subsection of the feature vector will show noticeable change. In contrast, change due to foreground objects can take on many different forms and therefore will have a greater affect on a larger subsection of the feature vector. In this fashion, Li performs feature selection to determine which features are useful for stationary backgrounds, moving background, and foreground for each pixel.

Li's algorithm begins by performing both background difference and temporal difference. Both operations require little computation, and classify a pixel as either stationary or changing. Foreground pixels can then be found using Bayesian classification over the feature vectors learned previously. The foreground pixels are then further processed using opening and closing operators to remove scattered error points and to connect foreground pixels. Background learning can then be applied using the current frame and foreground segmentation.

Zivkovic [**12**] makes a more direct extension to the original GMM. First, he simplifies the update equations by removing the ρ variable entirely,

$$\omega_{j,(x,y),t} = \omega_{j,(x,y),t-1} + \alpha(M_{j,t} - \omega_{j,(x,y),t-1}) \text{ - } \alpha C_T,$$

where C_T is a user defined constant such as .01.

$$\vec{\mu}_{j,(x,y),t} = \vec{\mu}_{j,(x,y),t-1} + M_{j,(x,y),t}\left(\frac{\alpha}{\omega_{j,(x,y),t-1}}\right)\left(\vec{I}_t(x,y) - \vec{\mu}_{j,(x,y),t-1}\right),$$

$$\vec{\sigma}^2_{j,(x,y),t} = \vec{\sigma}^2_{j,(x,y),t-1} + M_{j,(x,y),t}\left(\frac{\alpha}{\omega_{j,t-1}}\right)\left(\vec{I}_t(x,y) - \vec{\mu}_{j,(x,y),t-1}\right)^T\left(\vec{I}_t(x,y) - \vec{\mu}_{j,(x,y),t-1}\right).$$

The variable $M_{j,(x,y),t}$ is set to 1 for the "close" mixture with the largest $\omega_{j,(x,y),t}$ and all others are set to 0. They define a sample is "close" to a mixture if the Mahalanobis distance from the mixture's mean is less than three times the variance.

$$D^2_{j,(x,y),t}\left(\vec{I}_t(x,y)\right) = \frac{\left(\vec{I}_t(x,y) - \vec{\mu}_{j,(x,y),t}\right)^T\left(\vec{I}_t(x,y) - \vec{\mu}_{j,(x,y),t}\right)}{\vec{\sigma}^2_{j,(x,y),t}}.$$

If there are no "close" components, a new mixture is made.

Zivkovic also changes the mechanism for foreground/background classification. The author uses a non-parametric model based on a color-temporal kernel. A data set X_T represents the previous T time steps of collected data. This data can be grayscale, color or and other image space information. If the function

$$P_{non-parametric}\left(\vec{I}_t(x,y) \mid X_T, BG\right) \approx \frac{1}{TV}\sum_{m=t-T}^{t} b^m K\left(\frac{\left\|\vec{I}_m(x,y)\right\|}{D_{(x,y),t}}\right),$$

is greater than a threshold c_{thr}, the pixel is classified as background. The kernel function $K(u) = 1$ is $u < 1/2$ and 0 otherwise. The variable V is the volume of a hyper sphere of the kernel with diameter D. The median med is calculated for the absolute differences

$\left\|\vec{I}_t(x,y) - \vec{I}_{t-1}(x,y)\right\|$ of the samples from X_T, and a simple estimate of the standard deviation is used

$$D_{(x,y),t} = \frac{med_{(x,y),t}}{.68\sqrt{2}}.$$

Classifying the current pixel as background should only be done using information from other background pixel information in the set X_T. Therefore, the background model considers only samples with $b^m = 1$ representing previous samples that were classified to belong to the background.

2.1.5. Eigen Backgrounds

A more recent algorithm for background modeling uses Eigen analysis [**2**]. A set of background images is taken with a wide range of conditions such as lighting changes and cyclic motion of objects such as trees. The background data set can contain people, but better modeling is performed when no humans are present. N images are used as training data, each is unwrapped into a column vector,

$$\vec{v}_i = \begin{pmatrix} I_i(0,0) \\ \dots \\ I_i(R-1,0) \\ \dots \\ I_i(R-1,C-1) \end{pmatrix}.$$

R is the number of rows and C the number of columns. The mean of the N samples is computed

$$\vec{\mu} = \frac{1}{N}\sum_{i=1}^{N} \vec{v}_i.$$

and a matrix is formed from the concatenation of mean subtracted data points,

$$A = [\ (\vec{v}_1 - \vec{\mu}) \ldots (\vec{v}_N - \vec{\mu})\ ].$$

Oliver [**2**] then proposes building a matrix of the outer product of the sample points

$$\mathrm{Cov} = AA^T$$

and performing Principle Component Analysis (PCA) on this matrix. For images of even moderate resolution, the dimensionality of the sample points is huge, (307200 for 640x480 images). This results in the C matrix being very large, (307200x307200 for 640x480 images), and PCA require a significant amount of computation.

Therefore, to reduce the complexity of PCA on the input data, a clever mathematical trick is applied. A matrix is created using an inner product

$$L = A^T A,$$

Instead of the covariance matrix. The size of which is based on the number of samples, NxN, instead of the dimensionality of the data. The matrix L is subjected to Eigen analysis, where Φ is the matrix of eigenvectors, whose vectors are arranged in column format, and Λ is the matrix of eigenvalues, where the eigenvalues are aligned along the diagonal and the matrix is zero at off diagonal elements. The covariance of the data set can be decomposed as follows

$$Cov = \Phi^{\mathrm{T}} \Lambda \Phi.$$

PCA can then be applied to an NxN matrix because

$$A^T A\vec{\Phi}_i = \Lambda_{i,i}\vec{\Phi}_i,$$

where $\vec{\Phi}_i$ is the ith column vector. However, multiplying both sides by A we have

$$AA^T A\vec{\Phi}_i = \Lambda_{i,i}A\vec{\Phi}_i,$$

That is, eigenvectors of L are also eigenvectors of AA^T. This trick reduces the computation required for PCA to the number of sample points instead of the number of pixels in each image.

The matrix U represents the Eigen background images.

$$U = [\ \vec{\Phi}_1\ \ldots\ \vec{\Phi}_{N'}\ ]$$

A matrix V is then created

$$V = AU,$$

which projects the original data onto the space used for Eigen analysis.

For each new image I, the mean is subtracted and projected into the Eigen space.

$$\vec{\Omega} = V^t(\vec{v} - \vec{\mu})$$

The image is then mapped back into the original space.

$$\vec{s} = V\vec{\Omega}$$

The absolute difference between the mapped image and the original image is then computed.

$$\vec{\zeta} = |(\vec{v} - \vec{\mu}) - \vec{s}|$$

Any pixels having $\vec{\zeta} > \tau$ are considered foreground.

A major downfall of the original Eigen Background procedure is that it cannot update over time. Combinations of a finite discrete number of lighting possibilities are used in training with the hope that future lighting conditions are similar. If a lighting condition occurs that is not in the training set, the algorithm will perform poorly. Research has been performed to extend the Eigen Background algorithm to update over time [**3**,**4**,**5**], however the same problems as with Gaussian mixture exist, namely shadows, specular reflections, moving objects, a procedure not-specific to human segmentation, failure to properly recover from a corrupt background model .

2.1.6. Wallflower

A newer form of background modeling is the Wallflower algorithm. This algorithm was developed at Microsoft [**10**] to address common problems associated with background modeling, such as moving objects, sudden and gradual lighting changes, and camouflage. Wallflower addresses several of these situations with moderate success and continues to be one of the most commonly used change detection algorithms.

Wallflower is a unique background subtraction algorithm in that it models multiple scales consisting of pixel, region and frame levels. Models are maintained at the pixel level and processing is performed at runtime to classify the foreground and model the background. At the region level, information across multiple pixels is used to refine the foreground classification. Frame level modeling handles large changes in lighting due to synthetic light sources such as overhead fixtures.

The pixel level modeling is performed using a Wiener prediction filter [**13**]. This prediction modeling treats each pixel as a signal source through time, filters the signal, and produces an estimated next signal.

$$S_t = -\sum_{k=1}^{p} a_k S_{t-k}$$

S_t is the predicted value at frame t, S_{t-k} and the past values of the pixel, and the a_k are the prediction coefficients. The filter uses p past values to make its prediction. The expected squared error $E[e_t^2]$, is

$$E[e_t^2] = E[S_t^2] + \sum_{k=1}^{p} a_k \, E[S_t S_{t-k}]$$

The values of a_k are computed from the sample covariance values of the S_n. The author in [**10**] uses the past 50 values of S_n to compute $p = 30$ prediction coefficients. The difference between the predicted value and the true value is computed as the amount of change at that pixel for a new image. Pixels with a difference greater than a predefined threshold, $4\sqrt{E[e_t^2]}$, are considered foreground.

The algorithm has a region level processing mechanism which fills in areas of change having a flat color. In these situations, sequential frames register change at the leading and trailing edge of the area. The red areas in figure 2.2 (a) displays the change detected between frames t-2 and t-1 for a homogeneously valued disk translating to the right. The blue region of figure 2.2 (b) displays the change detected between frames t-1 and t for the same homogeneously valued region. The intersection of these two change detections results in the purple area of figure 2.2 (c). For each 4-connected region of change (such as the purple regions in figure 2.2 (c)), a color histogram is created using the original image at time t-1. These change regions are then used as seeds to grow back the areas of similar color in the original image at time t-1.

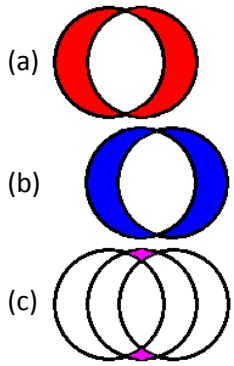


Figure 2.2: A shortcoming of sequential difference of frames for areas of homologous color or intensity. (a) A disk of homologous color moves to the right. Sequential difference of the disk at times t-2 and t-1 only register change on the leading and trailing edge of the disk at time. (b) The same occurs between time t-1 and t. (c) The only area certain to be change at time t-1 is the intersection of (a) and (b) resulting in the purple areas.

A high level approach is also taken with this algorithm. Two separate models are maintained to handle drastically conditions of lights being on or off. Both models are used for background subtraction when a new image is present. The model that produces the least amount of foreground is chosen as the winner and is updated accordingly. While this algorithm addresses more problems than most, it still suffers from shadows, specular highlights, the specific heuristics used to detect major scene illumination changes and correct for them, and as in all background-modeling procedures, correction from a corrupt current knowledge source is very difficult.

2.1.7. Disparity for Background Modeling

All of the algorithms described up to this point are based on simple image space features such as color, intensity and texture from a single camera. None of these proposed background modeling approaches can handle the large number of complex and dynamic factors in a scene, given such simple image space features. The most notable shortcomings of these models are changes in synthetic lighting from lamps or overhead fixtures. None of these approaches can handle these dramatic light changes, while accurately tracking proper foreground change. They can update the model over several frames, but that requires some time and can be inaccurate when a person is present in the scene during a dramatic lighting change. The heuristics used to develop these algorithms makes them less robust in real world situations.

As proposed in [**6**], pixel depth is a more robust feature for background modeling and foreground segmentation. Though the lighting may change over time, at any given moment, the lighting is approximately the same between the two images of the stereo pair. Pixel matching between these images returns a disparity, which is converted to the distance of objects to each pixel from the camera. Using these distance values, a background model can be built. Any significant change in the background, now in terms of depth, represents a change in the scene.

Another strength of the depth feature is its robustness to shadows. Because shadows are a product of lighting change, i.e. the object is still in the same physical position, they are handled implicitly by this feature. Comparatively, algorithms relying on simple color, intensity, and texture image space features must include a separate system for shadow removal. To date, no shadow removal algorithm has performed sufficiently in a large range of complex and dynamic situations. They either remove too much or too little of the foreground region.

Several researchers have continued the use of depth for change detection [**14**,**15**,**16**,**17**]. This work has furthered the progress of change detection research, but contains many of the shortcomings of previous models. These approaches mainly apply depth information to previously created algorithms and make subtle extensions. None of them work well enough in real-world situations. The papers associated with these algorithms only demonstrate success with benign scenes having little or no background objects.

It should be noted that there are problems with the depth-based approaches. Most notable are reflections from mirrors and specular reflections of light. These types of problems are viewpoint dependent and therefore return depth values greater than the proper distance. Also, depending on the feature used in computing image correspondence, there can be problems with a large area of a single flat color or repeated textured regions. However, when compared to the progress made in the area of background modeling, greater progress has been made in the area of stereo image correspondence [**18**]. As this work empirically demonstrates, even in light of these problems, this general approach is far superior as it relates to the modeling and segmenting humans from a scene.

2.2. World Space Algorithms

Image space refers to a two-dimensional plane in which raster images and their intensities or color information reside. World space refers to a three-dimensional space in which quantities are now represented by volume elements, voxels. The pixel level intensity and color information do not portray the complex information in a three dimensional scene. The modeling of world space values has a larger resource requirement with relation to computation and memory footprint versus image space modeling, but contains a greater degree of rich object information, and leads to more robust modeling of scenes. Multiple three-space systems are described in the section to familiarize the reader with current work in this research area.

It should be noted that the two systems described in the following sections are only similar to the system proposed in this dissertation and not directly related. Neither system segments the entire human from the scene as is done in our system. Therefore, there is no direct way to compare the outputs of these two systems to the one proposed in this dissertation. The description of these systems is provided simply to familiarize the reader with the state-of-the-art in the field and show alternate paths taken by other researchers.

2.2.1. People Detection and Tracking Using Stereo Vision and Color

For three-dimensional identification and tracking of humans, [**19**] begins by using stereovision to create a disparity map of the scene. The authors then determine the three-dimensional position corresponding to each pixel in the scene and those points are projected onto a horizontal ground plane. The horizontal plane is quantized and each cell is assigned the maximum height from the set of points projected onto the plane at that cell. To remove noise, the median height value over a series of frames is used at each time step.

More specifically, each pixel in the disparity map is first transformed into three-dimensional camera space using the disparity, d, the baseline between the cameras, b, and the pixel location (u, v) in the image.

$$Z_{cam} = \frac{fb}{d}$$

$$X_{cam} = \frac{uZ_{cam}}{f}$$

$$Y_{cam} = \frac{vZ_{cam}}{f}$$

$$P_{cam} = \left\{p_{cam}^{0}, \dots, p_{cam}^{np-1} | p_{cam}^{i} = \left(X_{cam}^{i}, Y_{cam}^{i}, Z_{cam}^{i}, 1\right)^{T}\right\}$$

Knowing the position and orientation of the camera, the camera space vectors are then transformed into world space using a 4x4 linear transformation matrix T.

$$p_w = Tp_{cam}$$

$$p_w^i = \left(X_w^i, Y_w^i, Z_w^i, 1\right)$$

Each point in world space is then projected to the horizontal plane of x and y,

$$x^i = \left(X_w^i/\delta\right),\ y^i = \left(Y_w^i/\delta\right),$$

where δ is the width of each cell, 1 cm, on the horizontal plane. The sets of all points on the horizontal plane are made up as

$$P_{(x,y)} = \left\{i | x^i = x \wedge y^i = y \wedge Z_w^i \in [h_{min},\ h_{max}]\right\},$$

where $[h_{min},\ h_{max}]$ is a height range in which all points must reside. This range removes points that are either too high or too low to be of importance to the segmentation problem.

An instantaneous height map is then built for the scene for each input image at time t,

$$H_{(x,y)}^{t} = \begin{cases} max\left(Z_w^j | j \in P_{(x,y)}\right) & if\ P_{(x,y)} \neq \emptyset \\ h_{min} & if\ P_{(x,y)} = \emptyset \end{cases}.$$

A more robust height map is then built using the median of the previous 13 instantaneous height maps,

$$\widehat{H}_{(x,y)} = median(H_{(x,y)}^{t=t_0}, \dots, H_{(x,y)}^{t=t_0+12\Delta t}),$$

where Δt is 400 ms.

Given a height map $\widehat{H}$, a map signifying change in the horizontal plane can be determined. Denote

$$F_{(x,y)} = \left\{ i | i \in P_{(x,y)} \wedge Z_w^i > \widehat{H}_{(x,y)} \right\}$$

as the density map.

This density map must be normalized by the amount of surface that it represents in the real scene.

$$O_{(x,y)} = \sum_{j \in F_{(x,y)}} \frac{\left(Z_{cam}^j\right)^2}{f^2}$$

Human detection is now the procedure of finding an object, the visible point cloud, in the density map and verifying that it is a human. The paper goes on to describe a face detection algorithm and a Kalman filter used for tracking. Neither is relevant to this dissertation.

The problems with this approach include:

- The overly simple fashion in which a height map is assumed, built, updated, and used for segmentation. However, it does represent a more significant approach to the detection of humans over any of the prior mentioned background model approaches.
- No sufficient general object segmentation procedure is proposed, only point clouds are used. When real-world objects are merged, such as the human sitting on the couch, there is no attempt to identify which set of points belongs to which objects.
- No adequate tracking procedures are proposed.
- Experiments used benign scenes with unchanging background.

2.2.2. Human Head Tracking in Three Dimensional Voxel Space

In [**20**], multiple stereovision camera pairs are used and the visible volume of the object is identified. For each camera, every voxel is assigned a label as "inside", "outside", or "surface". In addition, "surface" voxels are assigned an RGB color value. The type of voxel is determined by

$$L^{C^n}(V_i, j, k) = \begin{cases} Outside & D_1 < D_2 + th \\ Inside & D_1 < D_2 + th \\ Surface & |D_1 - D_2| < th \end{cases},$$

where th is a constant defined by the user, D_1 is the distance of the voxel to the camera, along its respective pixel ray, and D_2 is the respective computed depth value for that

ray. The voxel assignments of four unique cameras, thus two stereovision pairs, are merged using

$$L(V_i, j, k)\text{=LookUpTable}(L^{C^A}, L^{C^B})$$

The look up table is shown in table 3.1.

Table 2.1: The lookup table used to combine multiple camera descriptions of a single voxel.

		$L^{C^A}_{i,j,k}$		
		Outside	Surface	Inside
$L^{C^B}_{i,j,k}$	Outside	Outside	Outside	Outside
	Surface	Outside	Surface	Surface
	Inside	Outside	Surface	Inside

After the stereo representations are merged, surface voxels are assigned an RGB value. Because multiple cameras may view the same surface voxel, the camera representing the shortest distance to the voxel is used for color assignment. The voxel is projected into screen space of the winning camera, and the median RGB value in that image is assigned to the voxel.

The paper then describes the tracking of a human head as a 6x6x6 set of voxels input to a particle filtering algorithm. The experiment section showed that the system was able to track the head of a person for over 1000 frames at 5 frames per second.

The problems with this approach include:

- Head detection is not described.
- Only the head of the person is tracked. Whole body segmentation would be more useful for higher level processes.
- Experiments used benign vacant scenes with unchanging background.

2.3. 3-D Segmentation

Researchers in several fields have developed algorithms to perform segmentation in three-dimensions. Most notable are those that come from computation intelligence, pattern recognition and computer graphics. Though the algorithms developed in these fields cannot be directly applied to the problems in this dissertation, aspects of each were implemented in parts of the system defined herein.

2.3.1. Computational Intelligence and Pattern Recognition

Within the fields of Computational Intelligence and Pattern Recognition, the three-dimensional segmentation problem can be cast as an instance of clustering. The main goal of clustering algorithms is to organize unlabeled input data into groups whose members are similar in some way. A large number of clustering algorithms have been developed to date [**21**,**22**,**23**]. Unfortunately, most of these algorithms require the number of clusters to be known ahead of time, something not practical to assume for the reliable segmentation of a voxel world. Cluster validity involves varying the number

of clusters and measuring a specific type of quality, generally based on how compact and well separated clusters are. However interpretation of the cluster validity results is not something easily automated and can be subject to interpretation, [**24**,**25**,**26**]. Additionally, cluster tendency algorithms exist, where tendency is the process of determining whether clusters are present in a data set [**27**]. Algorithms, such as VAT, depend on the cluster profile, primarily compact and well separated clusters, require a human to interpret, and are also subject to various interpretations. CLODD is an algorithm for visual interpretation and clustering from of VAT image [**28**].

Neural Gas (NG) is a more recent algorithm of note that differs greatly from classical clustering algorithms. Data is clustered in the original space using a small number of fixed nodes. Nodes in NG are similar to codebook nodes in a Self Organizing Feature Map (SOFM), [**29**]. However, unlike an SOFM, the topological structure of the nodes is a general graph not a rigid two dimensional lattice. Nodes are incrementally updated towards denser regions and the topological graph connections are updated based on nearest neighbors to data points and hence on “mutual excitation”, a Hebb’s rule approach [**30**]. This algorithm has the advantage of not requiring the number of clusters as input.

Both CLODD and NG represent unsupervised ways to acquire the number of clusters in a data set. However, given the structure of a three-dimensional world created using back-projection, problems exist. Specifically, back-projection error, such as the extended volume of objects in the non-visible areas, tend to connect real world objects, e.g. a

couch and an end table. Additionally, correspondence errors and back-projection result in noise throughout the voxel space.

Also, clustering algorithms are also usually susceptible to outlier data points. Outliers that are equidistant to two clusters often create an unwanted connection between the clusters.

Lastly, objects take on many different shapes that cannot be modeled. Therefore, the compactness, separatability, and structure, and hence, what models and measures are needed for measuring similarity to an object, are not the standard set used for clustering. More knowledge and specific procedures for segmentation via clustering for this domain are required than exhibited in current existing techniques.

Another field of research, Model-Based Vision, uses computer generated 3D models to recognize segments in the scene, [**31**,**32**,**33**]. These recognition models are exceptionally expensive for recognition and have become less popular in recent years. More important to this dissertation, this type of recognition is not useful for segmenting background objects. The main reason for this is because a model is needed for each type of object in the scene. Because this dissertation is to be used in an assisted living community, the number of possible couches and chairs is too large to be used.

2.3.2. Computer Graphics

The field of Computer Graphics provides the most directly relevant algorithms for the task of segmentation required for this dissertation. Research involving the use of multiple cameras [**34**], range finders [**35**] and disparity from stereo [**36**] have returned highly resolved three-dimensional models, both solid and visual hull representations. Applications of these systems range from the educational display of artifacts in museums [**37**] to model and character design in movies and video games. The main shortcoming of the majority of these algorithms is that they were developed to create a model of a single object, while the process for this dissertation multiple objects must be identified and modeled concurrently.

3. THREE-SPACE OPERATIONS

To familiarize the reader with the work proposed here, a set of procedures must be made clear. First, the problems related to stereovision are discussed. This includes the problems of correspondence and occlusion.

This chapter also describes the quantization of three-dimensional space into voxel space. Quantizing the space simplifies the projection of two-dimensional screen space coordinates into three dimensions and the segmentation of objects in that space. The projection of two-dimensional silhouettes into voxel space will also be described with and without the use of depth information.

Lastly, this chapter describes the current state of the system. This consists of the camera settings and placements, three-dimensional registration of camera coordinates as well as the modeling in voxel space.

3.1. Stereo Vision

Generally speaking, low level stereo vision correspondence approaches are categorized as local or global. Local algorithms take a single referent pixel in one image and match it against pixels in the corresponding row of the other image [**38**,**39**]. The pixel that most resembles the referent pixel is considered the matching pixel. These algorithms are rather inexpensive and can therefore be run in real-time. A drawback to local

algorithms is that they often return non-unique, incorrect matches due to the use of only local information.

In contrast, global algorithms begin with local matching information for all pixels across one or more rows to determine a unique and more accurate overall correspondence [**40**,**41**]. The downside is a dramatic increase in computational complexity compared to local algorithms. For this reason, the system described in this dissertation uses a local algorithm, though a global algorithm could certainly be used in the future given significantly increased computational power or a more efficient algorithm. The reason for desiring speed is the need for real-time systems for abnormal event detection such as falls.

3.1.1. Image Rectification

Disparity, the amount of parallax for a given point in image space between stereo images, is computed for a given pixel in an image by matching that pixel to its corresponding pixel in the other image. Matching a given pixel in one image to that in the other image is performed by finding the most similar pixel in the other image within the same row. Though the stereo images are taken side-by-side, due to lensing effects and small inaccuracies in the stereo rigging, the rows of pixels in each image do not correspond perfectly. The images are therefore rectified using Epipolar rectification [**42**]. This rectification ensures that the rows of each image correspond, as shown in figure 3.1. For this system the development kit designed for the Bumblebee2 cameras

developed by Point Grey is used to rectify the incoming images [**43**], but there is also a free Matlab package developed by the California Technical University [**44**] to perform the tasks of lensing calibration and image rectification.

Figure 3.1: Left and right images in a stereo pair is shown in the top row. The bottom images show the epipolar rectified images.

3.1.2. Correspondence

After image rectification, matching can be performed between the images in the rectified space. There has been a considerable amount of research performed on producing disparity maps from local features [**18**]. Most algorithms perform

convolution using a Laplacian of Gaussian kernel [**45**,**46**] over the input image and then match patches of pixels between the two images. The Laplacian of Gaussian convolution removes noise while enhancing changes in intensity such as edges. Because the output of this convolution is based on relative difference of intensity values, as opposed to absolute intensity values, matching is resistant to lighting differences between cameras. This property is helpful for a stereo setup that uses automatic color adjustment, because intensities might differ slightly between the stereo camera images.

First, the Laplacian of Gaussian convolution filter is pre-computed over the entire image. A descriptor is then built for each pixel location using a surrounding window of values, which is length W on each side. Therefore, a descriptor of size W^2 is created for each pixel in each image. The variable W is set to 11 here and therefore results in a descriptor of length 121.

Each pixel is compared to pixels in the corresponding row of the other image. But only a small range of these pixels, R, needs to be tested for matching. The size of this range of pixels is related to the range of depth desired for recognition. We experimentally picked R=40. As well, a number of pixels, S, are ignored before the first pixel match (S = 5 here). For our stereo setup, this results in a recognition depth range of roughly 1 to 10 meters from the cameras. A change in R would affect the near end of the depth range, while a change in S would affect the far depth range. Figure 3.2 graphically demonstrates these variables.

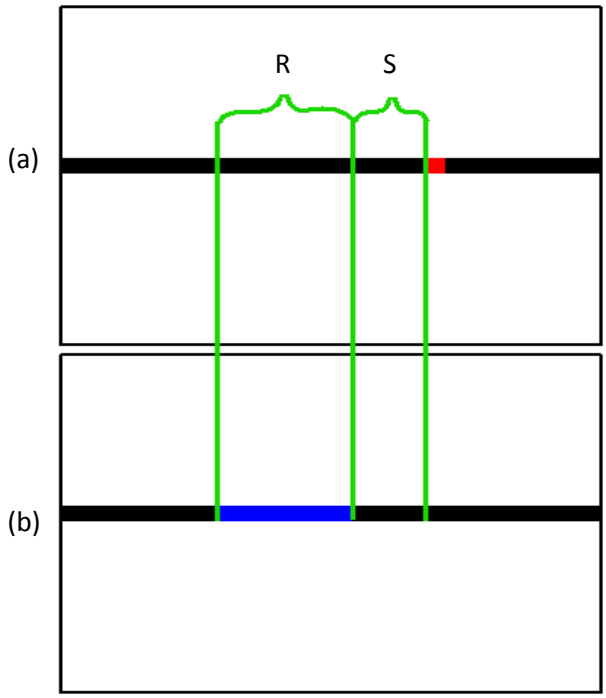
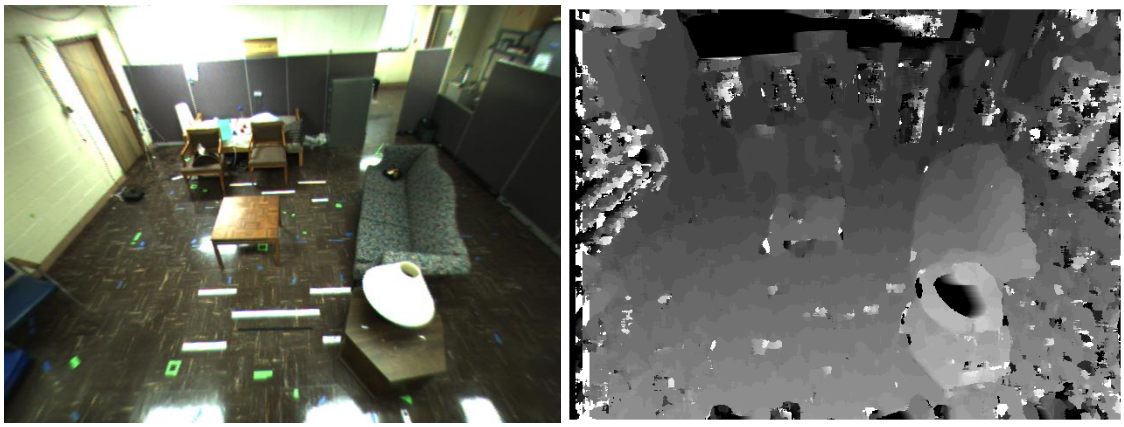


Figure 3.2: Only a fraction of pixels along a scan line need to be tested for stereo matching between two images. The value of S represents the number of pixels to skip before performing matching. The value of R signifies the number of pixels along the scan line to be matched. Smaller values of S extend the far end of the viewing range with a theoretical maximum of infinity at S=0. Larger values of R bring the near end of the viewing range closer to the cameras. (a) The right image in a stereo pair. The red area is the pixel to be matched. (b) The left image in a stereo pair. Using the values of S and R, the descriptors for a set of pixels in blue are matched against those of the red pixel in (a).

Each pixel is then compared to its R related pixels in the other image. Matching similarity is computed as the sum of the absolute difference of the pixels' feature vectors. This results in a vector of R values for the matching pixel. The proper disparity match occurs at the index with the smallest difference. Figure 3.3 displays an output disparity map.



(a) (b)

Figure 3.3: Disparity map output. (a) The right image in a stereo pair. (b) The output disparity map. The black area at the top is due to incorrect matches from the flat white area in the original image. The far left area has significant mismatches due to repeating patterns of blocks on the wall.

The large majority of pixels in the resulting disparity image are derived from proper matching between the images. However, some pixels will not match properly due to occlusion, repeating patterns, or areas of no texture and uniform color. If two separate disparity maps are built, one representing the matching of pixels in the left image to

those in the right and the other being the matches of right to left, pixels can be tested for correspondence from left-to-right and right-to-left. Pixels that do not have corresponding matches between the images are marked as improper matches. Figure 3.4 shows the output of this correspondence check, where improper matches are displayed as white. Figure 3.5 shows the output after further checks such as texture and surface similarity [**43**]. Note that many of the pixels marked this way correspond to areas on the walls and floor of the room.

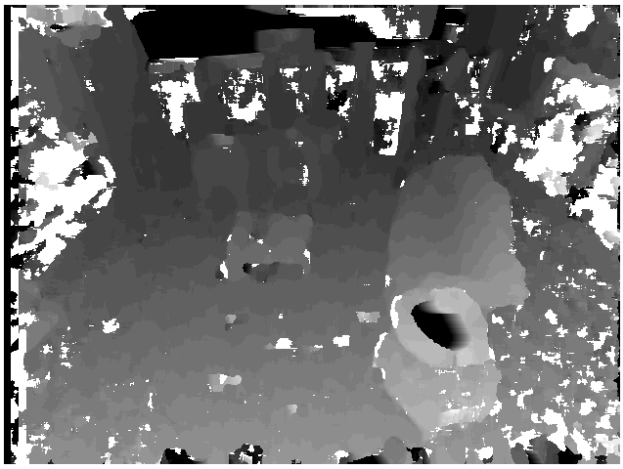


Figure 3.4: Left-to-right and right-to-left image matching is performed. Pixels having conflicting disparity are marked as white in the disparity image.

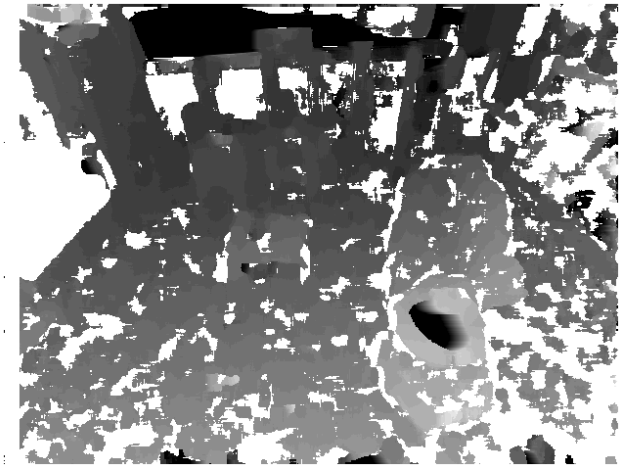


Figure 3.5: Further matching checks, performed by Point Grey [**43**], using texture and surface values can be performed to identify more pixels in white that are unreliable matches.

The approach defined to this point would result in only R levels of disparity over a range a range of 1 to 10 meters from the camera. This level of resolution is not adequate for the applications of scene modeling. The Point Grey stereo system therefore has the optional capability of producing a sub-pixel resolution in disparity through post processing. A Point Grey white paper [**47**] claims "stereo matches within $1/200^{th}$ of a pixel." Therefore, sub-pixel disparity applies to matching a specific pixel in the right image to a specific location in the left image up to a resolution of $1/200^{th}$ of a pixel. Unfortunately, Point Grey does not allow access to their code or describe the method used for sub-pixel disparity.

Before acquiring the Point Grey stereo vision package, I developed a stereo vision program which implemented a sub-pixel procedure. As described earlier, the matching of a given pixel results in a vector of length R, where each value represents the distance to a pixel in the opposing image. A parabola is constructed from the minimum index value and its two neighboring index values using a Taylor series expansion. The minimum of this parabola represents the true disparity of this pixel.

The finite differences at x are computed as

$$f'_{forward}(x) = f(x+1) - f(x),$$

$$f'_{backward}(x) = f(x) - f(x-1).$$

Both are used together to calculate the central difference,

$$f'(x) = \frac{f'_{forward}(x) + f'_{backward}(x)}{2},$$

$$f'(x) = \frac{f(x+1) - f(x-1)}{2}.$$

The second derivative of f is also computed using finite differencing,

$$f''(x) = f'_{forward}(x) - f'_{backward}(x),$$

$$f''(x) = f(x+1) - 2f(x) + f(x-1).$$

The parabola is approximated using a second order Taylor series expansion,

$$g(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2}(x-a)^2.$$

The derivate of $g(x)$ is then taken with respect to x,

$$g'(x) = f'(a) + f''(a)(x - a).$$

The parabola is minimum when the derivative is equal to zero,

$$g'(x) = 0,$$

$$0 = f'(a) + f''(a)(x - a).$$

The exact location of the minimum is

$$x = \frac{-f'(a)}{f''(a)} + a,$$

$$x = \frac{f(a-1)-f(a+1)}{2(f(a+1)-2f(a)+f(a-1))} + a.$$

The Point Grey software returns disparity values as a 16 bit number. It is likely that the software performs a sub-pixel function similar to that above and then stores the resulting disparity as 8 bits and the sub-pixel disparity as 8 bits. This would roughly correspond to their claim of a resolution of 1/200th of a pixel disparity.

3.1.3. Disparity to Depth Transformation

The disparity values are not directly helpful for the tasks in this system. Disparities must be transformed into three dimensional coordinates with relation to the camera. The three values are related to the screen space coordinates and disparity values, δ, as well

as intrinsic parameters of the camera such as focal length, f, and baseline between cameras, B.

$$(c_x, c_y) = \text{Center pixel location.}$$

$$(n, m) = \text{Image space pixel location.}$$

$$u = c_x - n$$

$$v = c_y - m$$

$$Z = \frac{fB}{\delta}$$

$$X = \frac{uZ}{f}$$

$$Y = \frac{vZ}{f}$$

The distance, d, from the camera is therefore,

$$d = \sqrt{X^2 + Y^2 + Z^2}$$

Figure 3.6 displays the conversion of the disparity map in figure 3.4 to a depth map with sub-pixel disparity.

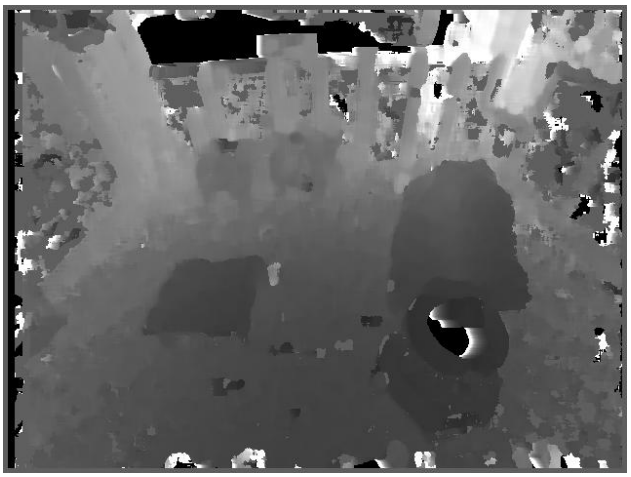


Figure 3.6: The depth map built using the disparity map in figure 3.4. Greater intensity represents greater distance from the camera. Sub-pixel disparity is used to create more accurate resolution in depth.

3.2. Voxel Space Camera Calibration

Before we can process any data, calibration must be performed to determine the world space coordinates and orientations for all cameras. Camera c, $1 \leq c \leq C$, at time t, $0 \leq t \leq T$, is located at position $\vec{p}_c$. An image, $I_{c,t}$, is a two dimensional collection, $N \times M$, of picture elements (pixels). A pixel's value is generally a discrete number indicating some characteristic of a volume of the scene, generally collected using a CCD-based sensor and light in the visible part of the electromagnetic spectrum. Images can also represent other phenomenon, such as the already discussed disparity or depth, which may or may not be intended for human visualization. More generally, pixels are grayscale (monochromatic) values, $\vec{I}_{c,t}(n,m) \in [0,1]$ or $\vec{I}_{c,t}(n,m) \in \{0,255\}$, where

$0 \leq n < N$ and $0 \leq m < M$, or any n-tuple, e.g. a multiple channel color space such as red, green, and blue (RGB), $\vec{I}_{c,t}(n,m) =< r, g, b >^T$, YCbCr, HSV, CIELab, or any other space in which luma and chroma are separated [**48**].

In addition to camera/image space, there is world space; a three-dimensional space in which members are volume elements (voxels). Voxels are non-overlapping cubes that discretize a volume, similar to how pixels discretize an image. Voxels are specified according to their center location, $\vec{v}_{(i,j,k)}$, and axis *widths*, $\{\omega_1, \omega_2, \omega_3\}$. Generally, the *widths* are the same for all dimensions, i.e. $\omega_1 = \omega_2 = \omega_3$. The voxel $\vec{v}_{(i,j,k)}$ is associated with a $\{0,1\}$ number indicating inclusion/exclusion of the element in a particular set (e.g. a three-dimensional object representing the human), a $[0,1]$ value indicating its probability (i.e. frequency of occurrence) or fuzzy membership (i.e. membership degree), or any n-tuple (e.g. its color according to its back-projection to a camera image plane). An environment, E, is first converted to a voxel representation,

$$E = \left\{\vec{v}_{(i,j,k)} \middle| i \in \{x_{min}, x_{min} + \omega_1, x_{min} + 2\omega_1, \dots, x_{max}\}, j \in \{y_{min}, y_{min} + \omega_2, y_{min} + 2\omega 2, \dots, ymax, k \in zmin, zmin + \omega 3, zmin + 2\omega 3, \dots, zmax, \text{ and } i,j,k \in \mathbb{Z}\}\right.,$$

where the minimum (min) and maximum (max) variables indicate the extreme bounds of the axes of E.

The goal of calibration is to have a list of voxels for each pixel representing the voxels visible to that pixel. These lists can only be built after each stereo pair's location and orientation is known within the scene, and the view volume of each pixel in those cameras in determined.

For each camera pair in the scene, a set of size NM, of unit length vectors is calculated,

$$\vec{R}_c = \{< x, y, z >^T_{(0,0)}, \dots, < x, y, z >^T_{(N-1,M-1)}\},$$

where $\vec{R}_{c,(n,m)}$ is the view ray vector for pixel (n, m). For stereo vision, at each location there are two cameras, the *left* and *right* cameras. For each stereo vision pair, both cameras are used to calculate disparity and depth, however only the *right* camera is subsequently used in voxel operations. In this work, at least two stereo pairs are required and are labeled c_1 and c_2 (i.e. the two *right cameras*). Initially, $\vec{R}_{c,(n,m)}$ is specified in a coordinate system local to camera c. To use the camera view vectors with voxels, all calculations must be performed in world space coordinates. Therefore, the majority of work related to calibration is in transforming the cameras' spaces to world space.

In [**49**] a method is described using singular value decomposition (SVD) to calculate a transformation matrix M through minimization of the resulting error in the approximated three space locations between the cameras. Suppose you have two 3D sets $\{\vec{p}_i\}: i = 1,2, \dots, N$ (here $\vec{p}_{1i}$ and $\vec{p}_{2i}$ are considered 3 x 1 column matrices). The two data sets $\vec{p}_1$ and $\vec{p}_2$, collected from two unique coordinate systems, are related by

$$\vec{p}_{1i} = R\vec{p}_{2i} + \vec{\Gamma} + \vec{\aleph}_i$$

where R is a 3 x 3 rotation matrix, $\vec{\Gamma}$ is a translation vector (3 x1 column matrix), and $\vec{\aleph}_i$ is a noise vector. We want to find the R and $\vec{\Gamma}$ that minimize the sum of squared error

$$\Sigma^2 = \sum_{\mathrm{i}=1}^{\mathrm{N}} \left\| \vec{\aleph}_i \right\|^2,$$

$$\Sigma^2 = \sum_{\mathrm{i}=1}^{\mathrm{N}} \left\| \vec{p}_{1i} - \left(R\vec{p}_{2i} + \vec{\Gamma} \right) \right\|^2.$$

The two datasets must first be translated so that their means are at the origin.

$$\vec{p}_1' = \frac{1}{N} \sum_{i=1}^{N} \vec{p}_{1i}$$

$$\vec{p}_2' = \frac{1}{N} \sum_{i=1}^{N} \vec{p}_{2i}$$

$$\vec{q}_{1i} = \vec{p}_{1i} - \vec{p}_1'$$

$$\vec{q}_{2i} = \vec{p}_{2i} - \vec{p}_2'$$

Error minimization then becomes

$$\Sigma^2 = \sum_{\mathrm{i}=1}^{\mathrm{N}} \| \vec{q}_{1i} - R\vec{q}_{2i} \|^2$$

$$= \sum_{\mathrm{i}=1}^{\mathrm{N}} (\vec{q}_{1i} - R\vec{q}_{2i})^t (\vec{q}_{1i} - R\vec{q}_{2i})$$

$$= \sum_{\mathrm{i}=1}^{\mathrm{N}} \vec{q}_{1i}{}^t \vec{q}_{1i} + \vec{q}_{2i}{}^t R^t R \vec{q}_{2i} - \vec{q}_{1i}{}^t R \vec{q}_{2i} - \vec{q}_{2i}{}^t R^t \vec{q}_{1i}$$

$$= \sum_{\mathrm{i}=1}^{\mathrm{N}} \vec{q}_{1i}{}^t \vec{q}_{1i} + \vec{q}_{2i}{}^t \vec{q}_{2i} - 2\vec{q}_{1i}{}^t R \vec{q}_{2i}$$

Therefore, minimizing Σ^2 is equivalent to maximizing

$$F = \sum_{\mathrm{i=1}}^{\mathrm{N}} \vec{q}_{1i}{}^{t} R \vec{q}_{2i}$$

$$= \mathrm{Trace}\left(\sum_{\mathrm{i=1}}^{\mathrm{N}} \mathrm{R}\vec{q}_{2i}\vec{q}_{1i}{}^{t} \right) = \mathrm{Trace(RH)}$$

where the 3 x 3 matrix H is

$$H = \sum_{i=1}^{N} \vec{q}_{2i}\, \vec{q}_{1i}{}^{t}.$$

Lemma: For any positive definite matrix $\mathcal{AB}$, and any orthonormal matrix $\mathcal{C}$,

$$Trace(\mathcal{AB}) \geq Trace(\mathcal{CAB})$$

Proof of Lemma: Let a_i be the ith column of $\mathcal{A}$ and b_i be the ith column of $\mathcal{B}$. Then

$$Trace(\mathcal{CAB}) = Trace(\mathcal{BCA})$$

$$= \sum_i b_i^t(\mathcal{C}a_i).$$

By the Schwartz inequality,

$$b_i^t(\mathcal{C}a_i) \leq \sqrt{(b_i^t a_i)(b_i^t \mathcal{C}^t \mathcal{C} a_i)} = b_i^t a_i.$$

Hence,

$$Trace(\mathcal{CAB}) \leq \sum_i b_i^t a_i = Trace(\mathcal{AB}).$$

Let the singular value decomposition of H be:

$$H = U\Lambda V^t$$

where U and V are 3 x 3 orthonormal matrices, and Λ is a 3 x 3 diagonal matrix with nonnegative elements. Now let

$$X = VU^t \text{ (which is orthonormal).}$$

We have

$$XH = VU^t\, U\Lambda V^t$$

$$= V\Lambda V^t$$

which is symmetrical and positive definite. Therefore, from Lemma, for any 3 x 3 orthonormal matrix $\mathcal{C}$,

$$Trace(XH) \geq Trace(\mathcal{C}XH)$$

Thus, among all 3 x 3 orthonormal matrices, X maximizes F. If $det(X) = 1$, X is a rotation matrix.

As previously stated, singular value decomposition of H returns

$$H = U\Lambda V^t.$$

The diagonal values of matrix Λ are the square roots of the eigenvalues of $\mathrm{HH^t}$ and $\mathrm{H^tH}$. The columns of V are the Eigen Vectors of $\mathrm{H^tH}$, while the columns of U are the eigenvectors of $\mathrm{HH^t}$. Alternatively, the matrix U can be calculated much simpler

algebraically after V has been calculated. Because the V matrix represents an orthonormal basis,

$$\mathrm{V}^{-1} = \mathrm{V}^{\mathrm{t}}.$$

Therefore,

$$\mathrm{U} = \mathrm{HV}\Lambda^{-1}.$$

Also, because Λ is a diagonal matrix, computing the inverse is trivial. Each value along the diagonal is the reciprocal of the value at that index.

$$\Lambda^{-1} = \begin{bmatrix} \frac{1}{\Lambda_{(1,1)}} & 0 & 0 \\ 0 & \frac{1}{\Lambda_{(2,2)}} & 0 \\ 0 & 0 & \frac{1}{\Lambda_{(3,3)}} \end{bmatrix}.$$

The rotation matrix is then computed as

$$R = VU^t.$$

Now the translation is found by

$$\vec{\Gamma} = \vec{p}_1' - R\vec{p}_2'.$$

It was later noted in [**50**] that if the data sets p_1 and p_2 are severely corrupted, the rotation matrix gives a reflection, $\det(R) = -1$, instead of a rigid rotation. Umeyama, [**50**], proposes the rotation matrix to be computed as

$$R = VSU^t,$$

where S is must be chosen as

$$S = \begin{cases} I & if\ \det(U)det(V) = 1 \\ diag(1,1,-1) & if\ \det(U)det(V) = -1 \end{cases}.$$

The matrix R is then always guaranteed to be a rigid rotation.

The rotation matrix and translation vector can be combined to create a 4 x 4 transformation matrix M. The upper 3 x 3 of the matrix is set to the rotation matrix R while the upper three values in the last column are set to the translation vector $\vec{\Gamma}$. The lower row is set to zero except for the last index which is set to 1. The matrix looks like

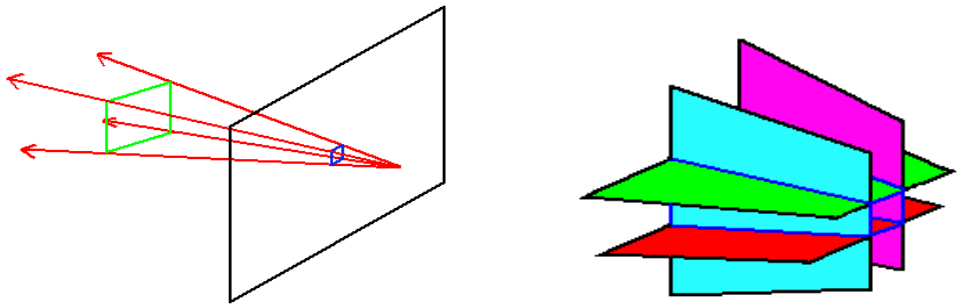
$$M = \begin{bmatrix} R_{(1,1)} & R_{(1,2)} & R_{(1,3)} & \Gamma_{(1)} \\ R_{(2,1)} & R_{(2,2)} & R_{(2,3)} & \Gamma_{(2)} \\ R_{(3,1)} & R_{(3,2)} & R_{(3,3)} & \Gamma_{(3)} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

To use this matrix, three dimensional view vectors must be homogenized by adding a one in the fourth index of the vector.

A 4x4 orthonormal transformation matrix, M_1, is constructed to transform c_2 to the coordinate system consistent with camera c_1. To achieve this, a unique calibration object is moved throughout the scene, where the exact location of the object over time is not known. Next, the object is located in image space and its camera space location is determined at each frame using the resulting depth from stereo vision and the view vector from each camera. This results in the two datasets p_1 and p_2 used for the SVD method previously described.

Finally, another 4x4 transformation matrix, M_2, is calculated by using known locations in world space, (it's easiest to use 9 locations on the floor), and the same SVD procedure as in computing M_1. Camera view rays can thus be transformed into world space for c_1 using M_2 and camera c_2 using $M_2 M_1$.

Now that the camera view ray vectors are given in terms of a global coordinate system, per-camera per-pixel voxel intersect lists are built, $L_{c,(n,m)}$. For a stereo vision pair, $L_{c,(n,m)}$ is only built for the *right* camera. For each pixel, a view volume is constructed and used to build the lists. Whether or not a voxel belongs to a given pixel's subset is determined by a series of inside-outside tests. The view volume of a single pixel has the shape of a pyramid, figure 3.7 (a). The four sides of this pyramid make up the decision planes for the inside-outside test, also shown in figure 3.7 (b).



(a) (b)

Figure 3.7: (a) A single pixel's viewing volume as a pyramid in three-dimensional space. (b) The four decision planes that make up the sides of the viewing volume of a pixel.

If the simplification of representing each voxel as a sphere instead of a cube is made, the inside-outside test is greatly simplified, while losing minimal accuracy. This is because testing for sphere intersection requires much less computation than box intersection. The center of each voxel, now represented as a sphere of radius r=0.0433 meters for a voxel of size 5x5x5 cm, is tested against each decision plane of a single pixel. If the point resides inside all decision planes, then the voxel is put into the voxel list $L_{c,(n,m)}$. But, the center of a sphere can reside outside a decision boundary and still intersect the view volume figure 3.8. Therefore, the sphere center can reside up to the radius of the sphere outside a decision boundary and still be added to the voxel list.

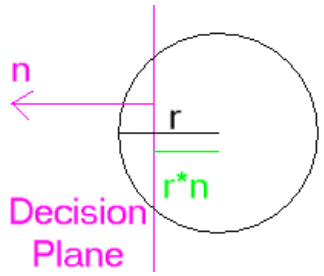


Figure 3.8: An edge on view of a decision plane with an intersecting sphere. The sphere is centered outside the plane, but part of its volume is inside the decision boundary.

The normals for all decision planes in the view volume pyramid for each pixel, $\vec{n}_{c,(n,m),q}$, are initially calculated in camera space and then transformed to world space using M_2

for c_1 and M_2M_1 for c_2, where $q \in \{1,2,3,4\}$ is the decision plane index. Voxel list inclusion is determined by

$$E_c = \{\vec{v}_{(i,j,k)} \in E | (\vec{v}_{(i,j,k)} - \vec{p}_c) \cdot \vec{n}_{c,(n,m),q} > -r \ \ \forall q \in \{1,2,3,4\}\}.$$

Building the voxel lists for a single camera for a voxel resolution of 5x5x5 cm for a world of size 128x128x64 voxels takes approximately 1 to 2 minutes on current common hardware when an octree spatial partitioning of the environment is used [**51**]. This procedure only has to be computed once offline. Next, each voxel list is sorted according to the voxel distances from the camera.

Because the camera calibration is a cumbersome procedure, I have written a program to streamline the process. This program integrates the Point Grey stereo vision package to simplify stereo calibration and frame grabbing. The cameras must first be placed in their desired locations. The user then moves a target through the scene. The target is a blue square with a red dot in the center. This target is easy to find in a color image and returns reliable stereo depth values. After data collection the transformation matrix between the two camera coordinate systems is computed. Another data set is collected placing the target at unique locations in world space. The transformation matrix between camera 1 and the world is then computed. The cameras are queried for their pixel view vectors, they are transformed into world space and the voxel pixel lists are created.

3.3. Voxel Operations

Given the sorted voxel lists and a depth map, the three-dimensional representation of a scene can be built quickly online. There are two general approaches to modeling a scene using voxels, additive and subtractive. Additive modeling starts with an empty set (scene) and adds elements (voxels) to that set as they are found. Respectively, subtractive modeling starts initially with the full voxel space, and elements (voxels) not visible are removed. At time t, for each pixel in each *right* camera, the depth value is used to identify its corresponding voxel in list $L_{c,(n,m)}$. In the additive approach, the scene according to camera c, $E_{A,c}$, is the union of voxel subsets from $L_{c,(n,m)}$, particularly all voxels located behind (i.e. greater) and including the current depth,

$$E_{A,c} = \bigcup_{(n,m)} \{\vec{v}_{(i,j,k)} \in L_{c,(n,m)} | \|\vec{v}_{(i,j,k)} - \vec{p}_c\| \geq d_{(n,m)} \},$$

where $d_{(n,m)}$ is the current depth value for pixel (n, m) at time t obtained from stereo vision. In contrast, $E_{S,c}$ is calculated in a subtractive way by first computing the set $E_{F,c}$, which is the union of all voxel subsets from $L_{c,(n,m)}$ that have a depth value less than the current depth at pixel (n, m),

$$E_{F,c} = \bigcup_{(n,m)} \{\vec{v}_{(i,j,k)} \in L_{c,(n,m)} | \|\vec{v}_{(i,j,k)} - \vec{p}_c\| < d_{(n,m)} \}.$$

Next, $E_{S,c}$ is calculated as $E - E_{F,c}$, where, $-$, is defined as,

$$\Phi - \Omega = \{\vec{v}_{(i,j,k)} \in \Phi | \vec{v}_{(i,j,k)} \notin \Omega\},$$

where Φ and Ω are two voxel sets.

Because the subtractive method does not remove voxels outside of camera viewing frustums, large volumes can be incorrectly labeled as occupied. Therefore, we prefer to use the additive method, $E_{A,c}$, for this system.

Another frequently used set is the visible shell $E_{V,c}$. This set is the union of current depth intersected surface voxels from $L_{c,(n,m)}$,

$$E_{V,c} = \bigcup_{(n,m)} \{\vec{v}_{(i,j,k)} \in L_{c,(n,m)} \big| \|\vec{v}_{(i,j,k)} - \vec{p}_c\| = d_{(n,m)} \pm r \}.$$

In addition, the closest surface voxel for each pixel in camera c according to a depth map is

$$E_{D,c,(n,m)} = \underset{\vec{v}_{(i,j,k)} \in L_{c,(n,m)}}{\operatorname{argmin}} \left(\big| \|\vec{v}_{(i,j,k)} - \vec{p}_c\| - d_{(n,m)} \big| \right).$$

The visible shell for all objects is

$$E_V = \bigcup_{c=1}^{C} E_{V,c}.$$

This process of back-projecting pixels based on depth results in at least two types of error. The first type of error, category 1, i.e. *visible* error, is located between the camera and the nearest object for each pixel. This error generally occurs due to errors in stereo correspondence. The next error type, category 2, is *non-visible* error, i.e. the volume incorrectly identified behind the actual objects. Category 2 error is the most

predominant error for a single stereo camera pair because it represents unknown volumes that are occluded by objects. The use of multiple stereo camera pairs allows for the refinement of the environment and minimization of these errors. Information fusion and voxel space refinement is performed by intersecting the representations of voxel environments from multiple cameras.

Because category 2 errors occur when voxels cannot be seen by either camera pair, the best chance of seeing all voxels in a scene is to set all cameras' viewing directions as far apart as possible. With two stereo pairs, this means setting them 180 degrees apart, looking at each other, figure 3.8 (a). For three cameras it is best to set them 120 degree apart aiming at the center of the room, figure 3.8 (b). The angle between cameras should therefore be,

$$\Upsilon = \frac{360}{C}.$$

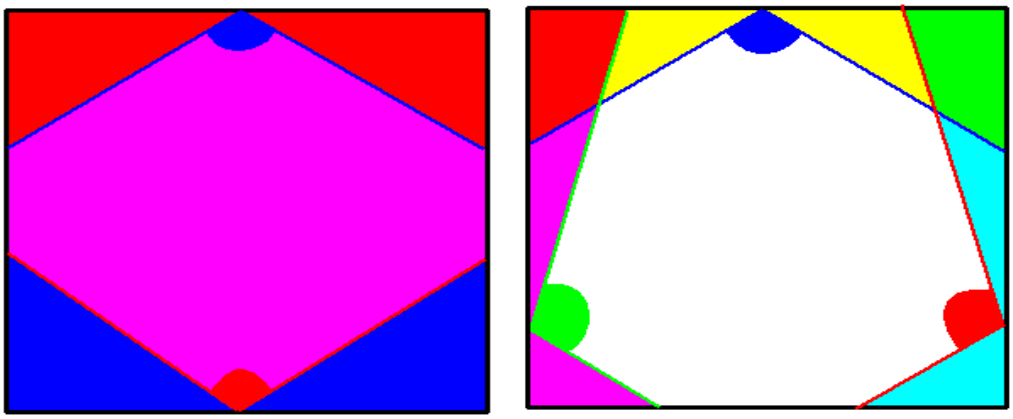


Figure 3.9: Camera placement affects category 2 error for best results. (a) Two stereo pairs should be placed at opposite ends of a room facing each other. (b) Three stereo pairs are placed 120 degrees apart from each other, oriented toward the center of the room.

Other set operations used in this dissertation include binary morphology, e.g. erosion, $\ominus$, dilation, $\oplus$, reconstruction, $\otimes$, union, $\cup$, and umbra, $\odot$ [**52**]. For example, the translation of a set of voxels Φ by $\vec{x}$ is

$$M_{\Phi,\vec{x}} = \left\{\vec{v}_{(i,j,k)} + \vec{x} | \vec{v}_{(i,j,k)} \in \Phi\right\}.$$

The erosion, of Φ by a structuring kernel K, is

$$\Phi \ominus K = \left\{\vec{v}_{(i,j,k)} \in \Phi | M_{\vec{v}_{(i,j,k)},\vec{x}} \in \Phi, \forall \vec{x} \in K\right\}.$$

Erosion helps reduce noise and counterbalance some of the effect of category 2 error. The dilation of Φ by K is

$$\Phi \oplus K = \bigcup_{\vec{x} \in K} M_{\Phi,\vec{x}}.$$

Reconstruction of Φ by K_1 and K_2 is defined by the algorithm

$\Gamma = \Phi \ominus K_1$

$while\ TRUE$

$\Omega = \Gamma \oplus K_2$

$\Gamma = \{\vec{v}_{(i,j,k)} \in \Phi | \vec{v}_{(i,j,k)} \in \Omega\}$

$If\ \Omega == \Gamma\ Break$

$DONE$

The reconstruction operation is used to remove random errors voxels that are due to incorrect stereo correspondence while keeping the proper segments representing real objects in the world. In our system, we have empirically selected a one step reconstruction (i.e. morphological opening), which is

$$\Phi \otimes K = \{\vec{v}_{(i,j,k)} \in \Phi | \vec{v}_{(i,j,k)} \in (\Phi \ominus K) \oplus K\}.$$

Lastly, the covering set, known as the Umbra, for a set Φ within voxel space E is defined as

$$\circledcirc \Phi = \{\vec{v} \in E | \exists \vec{g} \in \Phi\ where\ g_k \geq v_k\},$$

where the k[th] index is the third vector component of $\vec{v} \in E$ and $\vec{g} \in \Phi$, i.e. the world up direction. To put it another way, all vertical columns in the space are treated

individually and all voxels below or equal to the highest occupied voxel in each column are turned on, all above the highest voxel are turned off.

4. SYSTEM COMPONENTS

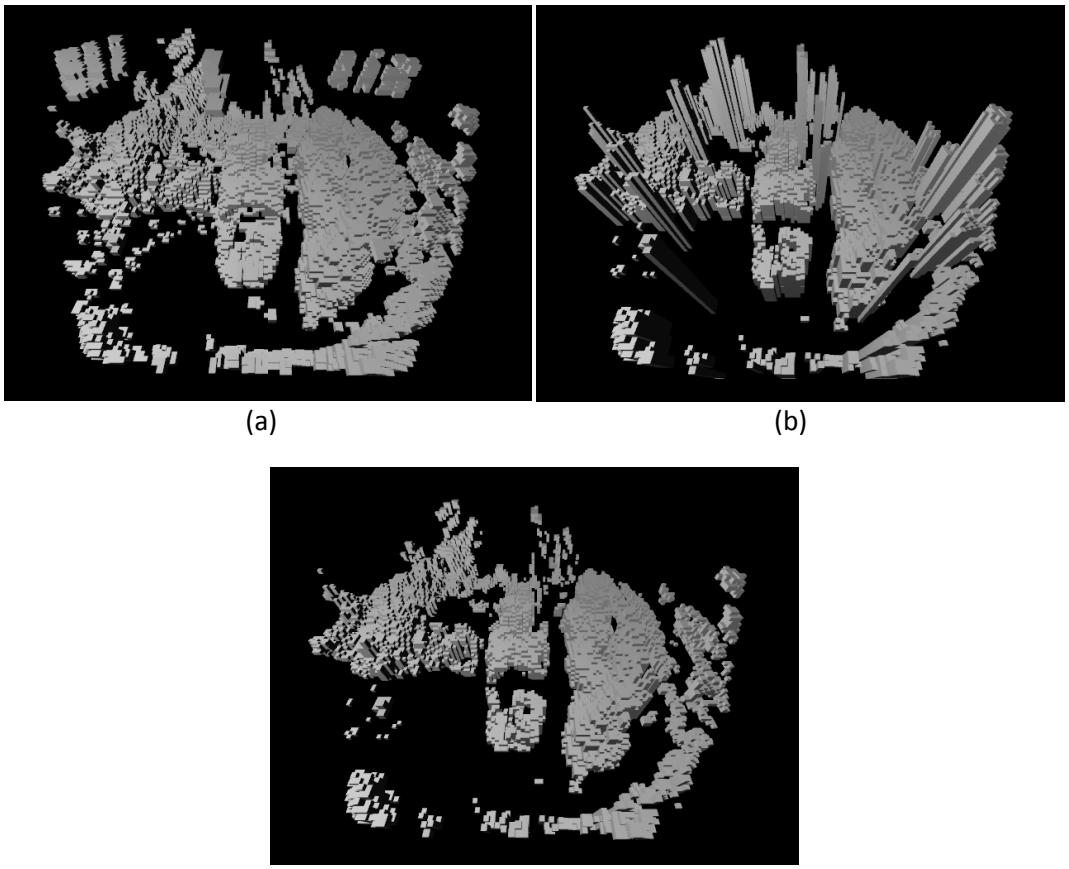
This system is a collection of many smaller subsystems. Stereo vision and voxel scene representation are two methods that have been defined in previous sections. This section defines the three novel methods designed for this system. These subsystems include segmentation, human detection, and color description of the human used for tracking.

4.1. Segmentation

A scene is comprised of many large and salient objects such as chairs, couches, tables and people. An approach is needed to segment an ungrouped scene ($E_A = \bigcap_{c=1}^{C} E_{A,c}$) into individual objects. There are many algorithms for segmenting 2D images [**53**,**54**,**55**] and 3D voxel spaces [**56**,**57**,**58**]. The majority of voxel space segmentation research has gone into algorithms assisting the segmentation of medical MRI data. All of these algorithms work on pixels or voxels that have a wide range of values and are therefore not directly applicable to this system. The voxel space in this system is binary and therefore does not require such complex segmentation algorithms.

To aid in object segmentation, we define the blanketed set, E_{Ξ}, for a set of voxels E_A using the umbra of the visible shell E_V

$$E_{\Xi} = \left\{ \vec{v}_{(i,j,k)} \in E_A \middle| \vec{v}_{(i,j,k)} \in \circledcirc E_V \right\}.$$



(a) (b)

(c)

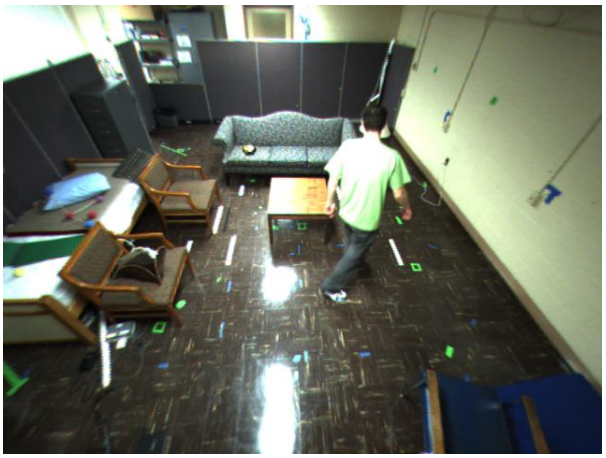
Figure 4.1**:** A visual representation of the covering set $\circledcirc E_A$ and blanketed set E_Ξ extracted from the intersected set E_A. (a) The intersected set E_A. (b) The covering set $\circledcirc E_A$. (c) The blanketed set E_Ξ.

This set is of particular value in downward camera viewing situations, such as monitoring elders in their homes. A visual representation of the blanketed set as well as the covering set and intersected set is shown in figure 4.1. Because category 2 (non-visible) error can have a drastic impact on later segmentation stages and the overall shape of an object, it is preferred to eliminate as much category 2 error as possible at

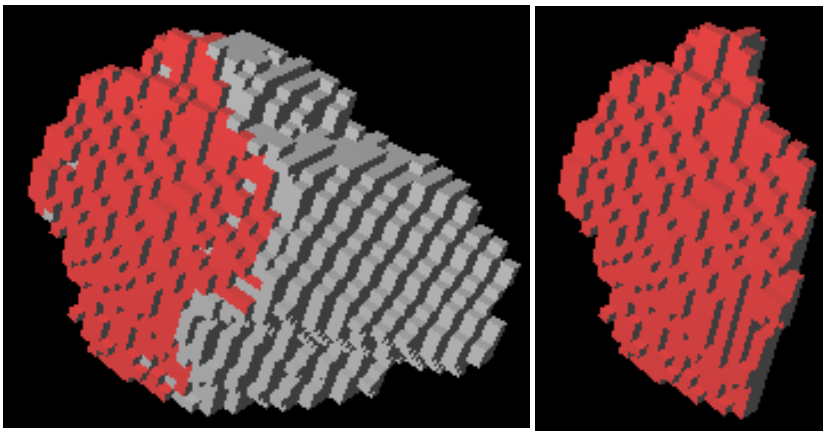
the risk of losing some of the actual object. Additionally, the visible shell of the blanketed set is

$$E_{\Theta} = E_{\Xi} \cap E_{V}.$$

Figure 4.2 illustrates E_A, E_{Θ}, and E_{Ξ} for a given stereo pair c.



(a)



(b) (c)

Figure 4.2: (a) Raw image from the *left* camera of stereo pair one. (b) The visible shell of the human, $\boldsymbol{E_{\Theta}}$, is shown in red as well as the intersection of all camera back-projections, $\boldsymbol{E_{A}}$, in gray. (c) The blanketed set of the human is $\boldsymbol{E_{\Xi}}$.

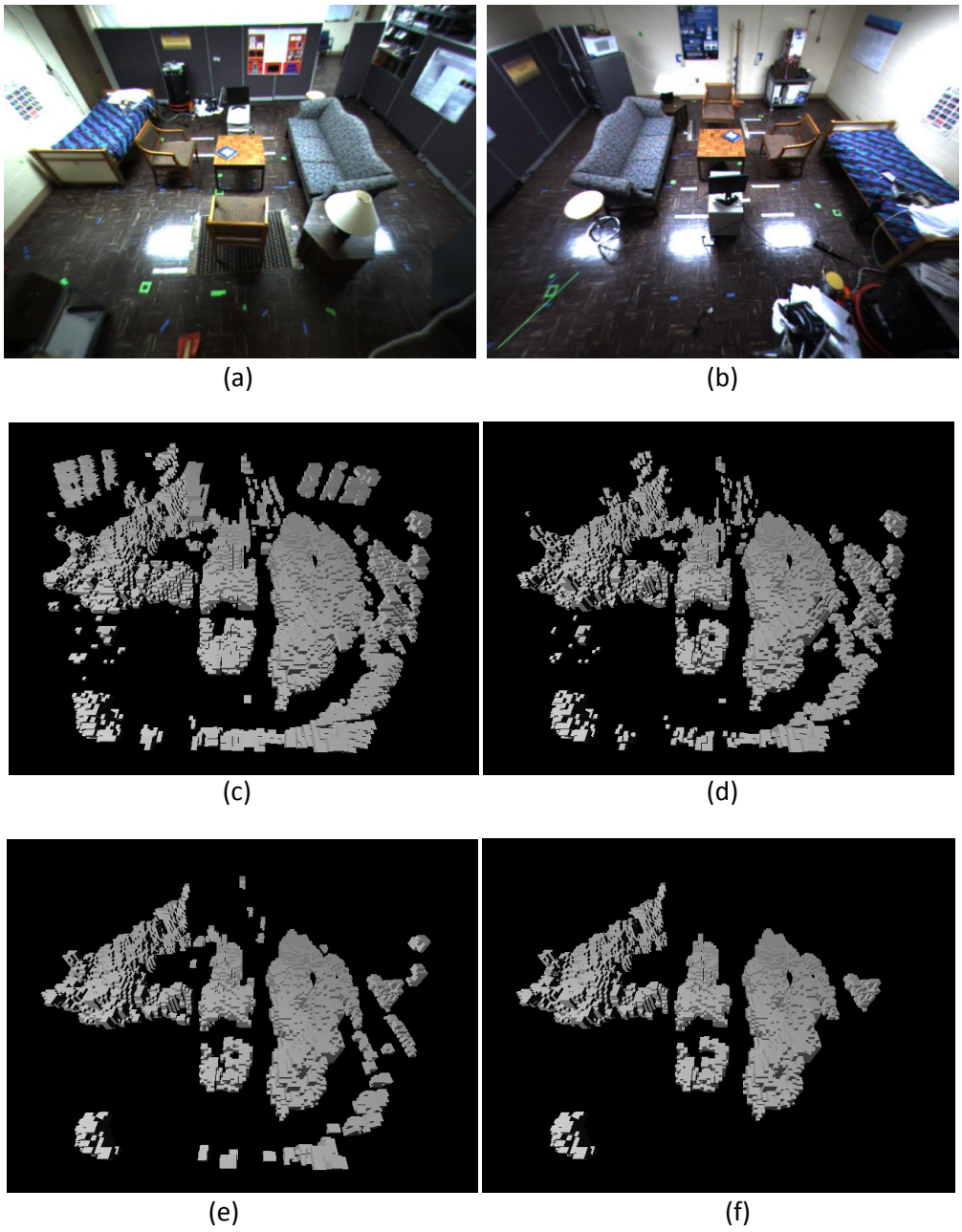
A one step reconstruction is then performed on E_{Ξ} using a 3x3x3 kernel of ones, K_M,

$$E_R = E_{\Xi} \otimes K_M.$$

In particular, this helps eliminate small islands of erroneous voxels. It also separates regions that are joined by only a thin connection of voxels.

Figure 4.3 shows the entire process. The intersection of the two stereo pairs is shown in figure 4.3 (c). Some erroneous voxels are visible at the top of the image, which result from incorrect stereo registration. Figure 4.3 (d) shows the blanketed set built using the umbra. Performing reconstruction on the blanketed set removes some thin volumes of erroneous voxels, figure 4.3 (e). The effects of reconstruction is most obvious on the far right side of image figures 4.3 (d) and (e). These coincide with small stereo registration errors behind the couch. Figure 4.3 (f) shows the objects that have volumes large enough to be considered salient objects for background modeling.

The three dimensional voxel space E_R can then be segmented using a connected components algorithm with 6-connectivity. Each component represents a segmented object in the scene. Components having a volume less than X voxels, (we empirically picked $X = 250$), are insignificant for purposes of this human tracking and are therefore rejected, resulting in L voxel objects, $\{O_1, \ldots, O_l, \ldots, O_L\}$, figure 4.3 (f).



(a) (b)

(c) (d)

(e) (f)

Figure 4.3: Segmentation of a scene using 2 stereo pairs into objects. Voxel representations are with respect to stereo pair one. (a) Original image from stereo pair one, right camera. (b) Original image from stereo pair two, right camera. c) The original voxel scene $\boldsymbol{E_A}$. d) The blanketed set of the voxel scene, $\boldsymbol{E_{\Xi}}$. e) The reconstruction $\boldsymbol{E_R}$ of $\boldsymbol{E_{\Xi}}$. f) The objects having less than 250 voxels are removed from $\boldsymbol{E_R}$.

Camera placement is specific to each unique environment. In our case, using two stereo pairs, the best results occur when the cameras are placed at opposite extremes of the room, near the ceiling, angled downward toward the center of the room as seen in figures 4.3 (a) and (b). This results in the greatest view volume and best coverage of subjects in the scene. This also returns better data for building the blanketed set and the shape of objects and people in the scene.

4.2. Human Detection

Since our goal is to analyze human activity, an algorithm is required to classify human and nonhuman segmented chunks in voxel space. One of the most salient features of the human body is the face. There are countless algorithms to find faces, [**7**,**8**,**9**], but the vast majority use image space features. For these algorithms to function properly, it is assumed that faces occupy a reasonably sized region of pixels in the image and most frequently are facing the camera. Neither of these properties is practical for the system developed in this dissertation. A human head detection algorithm is designed herein for voxel space using color, shape and height. None of these features are reliable enough to use alone, but the aggregation of these features is very robust.

Not all voxels can be assigned a color, only those in the visible shell of segmented objects. The set

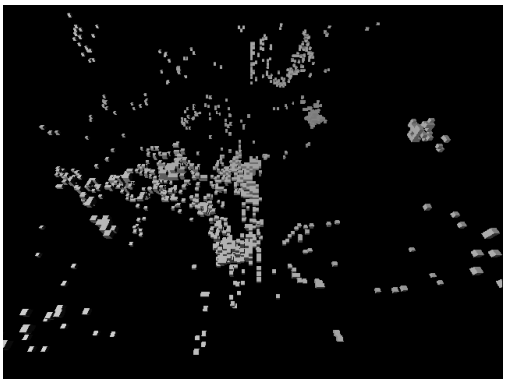
$$N_{(i,j,k)} = \left\{ \vec{I}_{c,t}(x,y) \in \{I_{1,t}, \dots, I_{C,t}\} | \vec{v}_{(i,j,k)} \in \bigcup_{l=1}^{L} \left(E_\Theta \cap E_{O_l}\right), \vec{v}_{(i,j,k)} = E_{D,c,(x,y)} \right\},$$

contains the color tuples for all pixels for all cameras for which $\vec{v}_{(i,j,k)}$ is the first visible voxel. The color of $\vec{v}_{(i,j,k)}$, $\vec{\omega}_{(i,j,k)}$, according to the color set $N_{(i,j,k)}$, is calculated as the median of the individual color components. Skin detection is then performed using the red and blue components of, $\vec{\omega}_{(i,j,k)}$. Many articles exist for skin detection [**59**,**60**,**61**], but none are robust enough for such unconstrained environments. Skin appearance can vary greatly in an unconstrained environment, for example with respect to florescent, incandescent, or other typical household illumination sources such as a television which emits many different colors of light at a rapid rate. Skin voxels are therefore recognized as,

$$E_\Omega = \left\{ \vec{v}_{(i,j,k)} \in \bigcup_{l=1}^{L} \left(E_\Theta \cap E_{O_l}\right) | \vec{\omega}_{(i,j,k),R} > \left(1.3\vec{\omega}_{(i,j,k),B} + .07\right) and\ \vec{\omega}_{(i,j,k),R} < \left(1.7\vec{\omega}_{(i,j,k),B} + .07\right) \right\},$$

where $\vec{\omega}_{(i,j,k),R}$ and $\vec{\omega}_{(i,j,k),B}$ are the red and blue channels for voxel at index (i,j,k) and the constants were chosen experimentally. Because these constants were chosen for data including a wide range of illumination possibilities, their use allows this system to accurately classify skin for future test possibilities. Figure 4.4 displays the result of skin detection in a typical scene.

(a) (b)

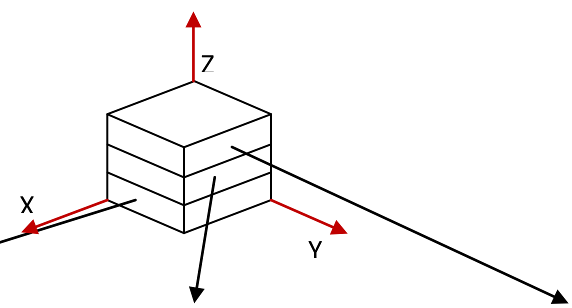


c)

Figure 4.4: Skin detection in voxel space. (a) The right image of the stereo pair. (b) Pixels labeled as skin. (c) The corresponding skin voxels.

Labeling skin over such a wide range of values leads to a large amount of false positives, shown in figures 4.4 (b) and (c). The false positives from this weak classifier are generally eliminated when used in combination with human head shape detection. The shape of the head is quite unique with relation to all objects in a living space, especially the top of the head. Therefore, the second feature is head shape, or more specifically the top of the head, in voxel space. A three-dimensional kernel, K, is built where values

reside in $\{-1,0,1\}$. Each kernel element represents the requirement that a particular voxel is part of a head, (i.e. 1 is the head, 0 is don't care, and -1 is not the head). Building this kernel is not a trivial task. This required trial and error to determine a kernel that is both rotationally insensitive around the vertical z-axis and also takes into account the possibility of the head being tilted, assuming the only joint is at the neck and the head is a rigid body. This kernel could be learned from training data, but was determined experimentally in this dissertation, figure 4.5.



-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	0	0	0	0	0	0	0	-1
-1	0	1	1	1	1	1	0	-1
-1	0	1	1	1	1	1	0	-1
-1	0	1	1	1	1	1	0	-1
-1	0	1	1	1	1	1	0	-1
-1	0	1	1	1	1	1	0	-1
-1	0	0	0	0	0	0	0	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	0	0	0	0	0	0	0	-1
-1	0	1	1	1	1	1	0	-1
-1	0	1	1	1	1	1	0	-1
-1	0	1	1	1	1	1	0	-1
-1	0	1	1	1	1	1	0	-1
-1	0	1	1	1	1	1	0	-1
-1	0	0	0	0	0	0	0	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

Figure 4.5: Three dimensional kernel intended to be shaped like the top of a voxel human head.

Correlation of E with the kernel K is performed. Voxels with a value greater than γ_1, (γ_1 chosen to be 23), represent voxels at the top of a head,

$$E_H = \left\{ \vec{v}_{(i,j,k)} \in E \middle| \sum_{a=-3}^{3} \sum_{b=-3}^{3} \sum_{d=-1}^{1} K_{a+3,b+3,d+1} \emptyset_1\left(\vec{v}_{(i+a,j+b,k+d)}\right) > \gamma_1 \right\},$$

where $\emptyset_1\left(\vec{v}_{(i,j,k)}\right)$ is a function that returns 0 if $\vec{v}_{(i,j,k)} \notin \bigcup_{l=1}^{L} E_{O_l}$ and 1 if $\vec{v}_{(i,j,k)} \in \bigcup_{l=1}^{L} E_{O_l}$. These voxels are then morphologically dilated using another kernel K_H, a 7x7x5 kernel of ones, to represent all head voxels.

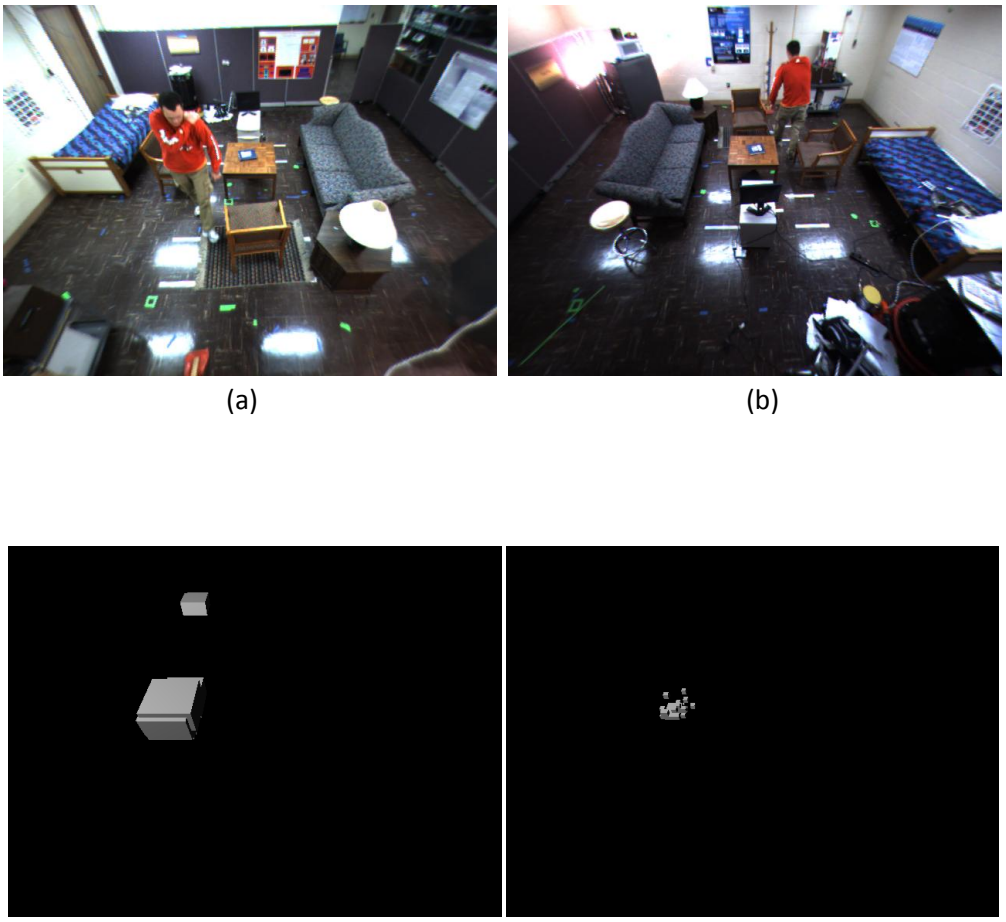
$$E'_H = E_H \oplus K_H.$$

This is shown in figure 4.6 (c). This set is then intersected with skin voxels to return face voxel regions, E_F,

$$E_F = E'_H \wedge E_\Omega.$$

Most face regions are properly detected in E_F, but there are also some false positives. Nearly all misclassifications encountered here are from skin colored chairs, or objects on tables. Only voxels with height greater than γ_2, (γ_2 chosen as 1.25 meters), above the floor are considered correctly classified faces, figure 4.6 (d), resulting in

$$E'_F = \left\{ \vec{v}_{(i,j,k)} \in E_F | v_k > \gamma_2 \right\}.$$



(a) (b)

(c) (d)

Figure 4.6: Voxel space detection of human faces. (a) Original image from stereo pair one, right camera. (b) Original image from stereo pair two, right camera. (c) Voxels whose neighborhood is shaped like a head. (d) The intersection of head regions, and skin voxels from figure 4.4 (c) above 1.25 meters (i.e., $\boldsymbol{E}_F'$).

Finally, for each object, O_l, if

$$|E_F' \cap O_l| > 0,$$

then the object is classified a human.

4.3. Color Descriptor

The previous section describes the process of recognizing a person standing or walking (i.e. upright) in a scene. Unfortunately, that process is unable to recognize a person who is sitting, kneeling, etc., because the subject's head is below 1.25 meters. Because the person object might not always be upright and the face might not always be visible, it is therefore necessary to have a secondary mechanism to recognize previously detected people. Color histogram matching is used to recognize and track a previously discovered human when the prior system does not detect a person.

An RGB color can be transformed [**48**] into a two-dimensional Euclidean space based on hue, $H \in [0, \pi)$, and saturation, $S \in [0,1]$, according to

$$x = \mathrm{S}\cos(H),$$

$$y = \mathrm{S}\sin(H),$$

where $x, y \in [-1,1]$. A color histogram is built for each object found using the algorithm in section 4.1. The hue-saturation space is quantized into Q^2 bins, Q in each dimension (empirically a Q=5 is used here). Each visible shell voxel for a specific object, $\vec{v}_{(i,j,k)} \in E_\Theta \cap E_{O_l}$, has its color added to its respective histogram bin. The bin indices are calculated as,

$$x' = \left\lfloor \frac{Q \cdot (x+1)}{2} \right\rfloor,$$

$$y' = \left\lfloor \frac{Q \cdot (y+1)}{2} \right\rfloor.$$

For each visible shell voxel for object O_l, a histogram bin $B_l(x', y')$ is incremented by one. The two dimensional structure can be represented as a one dimensional histogram using

$$\vec{\Psi}_l = \begin{pmatrix} B_l(0,0) \\ \dots \\ B_l(4,0) \\ \dots \\ B_l(4,4) \end{pmatrix}.$$

The color histogram of an object represents the distribution of colors on that object's surface. These descriptors can be used to match objects from segmented objects in future time steps.

5. COMPONENT AGGREGATION

In the Wallflower paper [**10**], Toyama states, “No perfect system exists.” A decade later, this statement is still true, but technological advances have allowed the use of significantly more complex information and tools, such as higher resolution images and stereo vision. In this chapter, a system is described using the techniques presented in previous sections. This method creates high level information using computer vision techniques to accurately detect and track a human in a scene. While the process is proposed for environments containing a single human, with minimal modification, the system could be adapted to multiple person environments. A high level algorithmic description is shown in the following figures.

A1: ALGORITHM 1

Multiple stereo camera calibration (Preprocessing)

(1) Determine lensing parameters and perform epipolar rectification
(2) Determine transformation matrix of camera 2 to camera 1 space
(3) Determine transformation matrix from camera 1 to world space
(4) Create voxel-pixel list for each pixel of right camera in each stereo pair

A2: ALGORITHM 2

Algorithmic System Overview (Runtime)

WHILE NOT DONE
// Stereo vision and voxel reconstruction
(1) Collect images from all stereo cameras
(2) Build individual account of voxel space for each camera pair
(3) Build intersected (global) voxel space
(4) Build blanketed set
// Change detection and object segmentation
(5) Remove background voxels from current blanketed voxel space
(6) Segment objects
// Human detection
(7) Build color histogram for all objects
(8) Find human from head shape and skin color or color histogram
(9) Update human color histogram
// Background update
(10) Remove human voxels from intersected voxel space
(11) Update background model using nonhuman blanketed voxel space

The desired output of this system is the current volumetric account of a human in the scene at each time step. In order to achieve this, an approximated model of the current nonhuman, three-dimensional scene is required at each time step. At runtime, voxels representing nonhuman locations of the previous T frames (T of 20 used here) are stored in $\{E_{B,t-T}, \ldots, E_{B,t-1}\}$. These T voxel spaces are volumetric snapshots of the

background scene over the previous seconds. The human detection and tracking algorithms defined in the previous sections perform well enough that these background snapshots can be accurately created with or without the human in the scene. This allows the system to be bootstrapped even with a human in the scene. Our assumption is that that voxels occupied by nonhuman objects in τ of the past T frames are part of the background (τ of 5 used in this system). Therefore, at time step t, the background, $E_{B,t}$, is defined as,

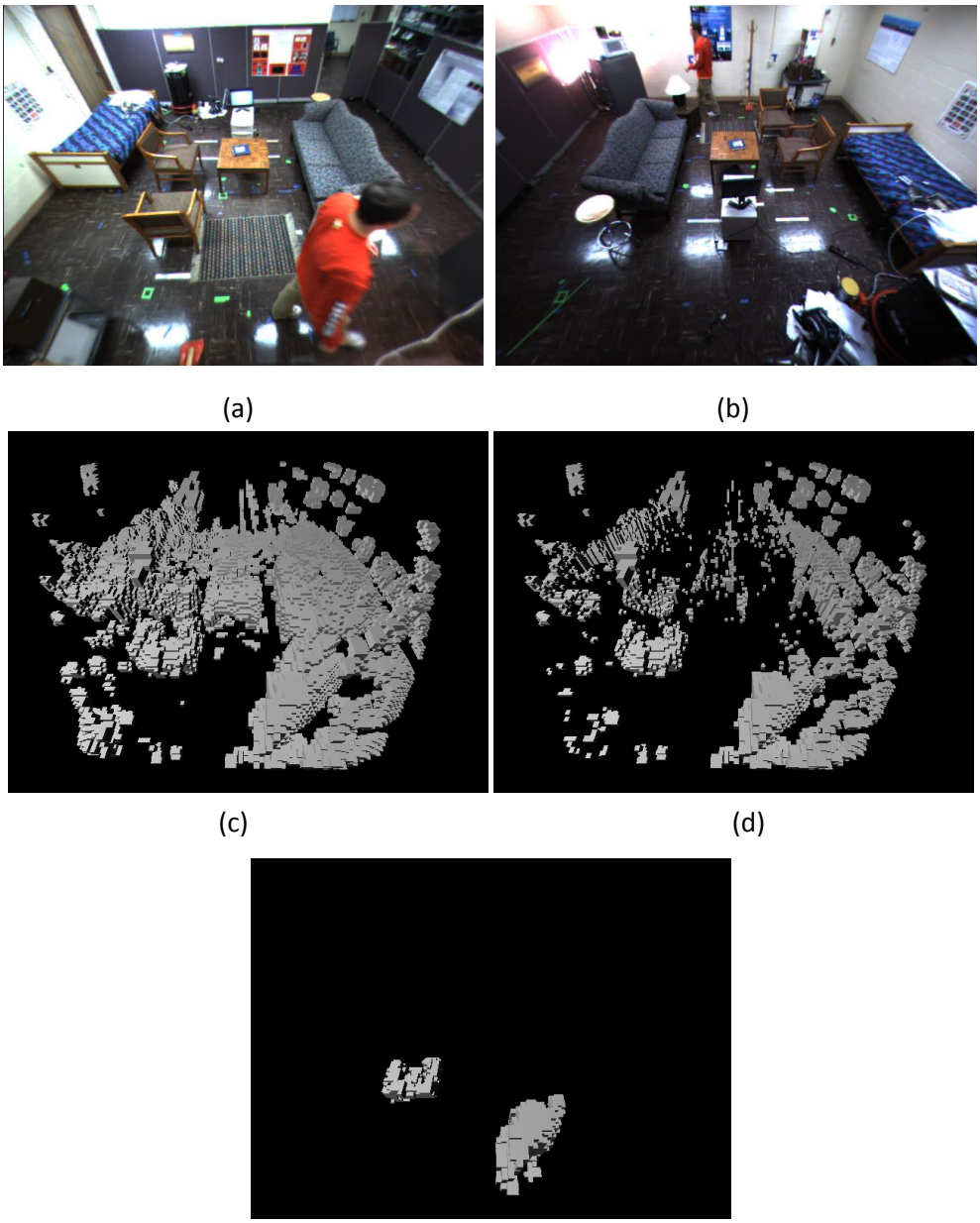
$$E_{B,t} = \left\{ \vec{v}_{(i,j,k)} \in E \middle| \left(\sum_{q=t-T}^{t-1} \emptyset_2\left(\vec{v}_{(i,j,k)}, q\right) \right) > \tau \right\},$$

where $\emptyset_2\left(\vec{v}_{(i,j,k)}, q\right)$ is a function that returns 0 if $\vec{v}_{(i,j,k)} \notin E_\mu$ and 1 if $\vec{v}_{(i,j,k)} \in E_\mu$ at time step q. The voxel space E_μ is the set of nonhuman occupied voxels, i.e. $E_\mu = E_A - O_H$, where O_H is a human object (one of the O_l objects classified as human). Hence, a voxel is part of the background if it is occupied by a nonhuman in 5 of the past 20 back-projected and intersected voxel spaces. Once a background model is created, change detection is trivial. Removing background voxels at time t results in voxels that represent change, E_{Δ_t}, Figure 5.1 (d),

$$E_{\Delta_t} = E_{A,t} - E_{B,t}.$$

Most frequently, voxels representing change are related to a person walking through the scene, but this is not guaranteed. It is possible that a person has moved an object in the scene such as a chair. It is therefore important to segment the change voxel space

into multiple objects and remove small objects as was described in section 4.1, (see figure 5.1 (e)).



(a) (b)

(c) (d)

(e)

Figure 5.1: Change detection and segmentation of voxel space. (a) Original image from stereo pair one, right camera, showing a person who is moving through the scene. (b) Original image from stereo pair two, right camera. (c) Intersected voxel space from two stereo pairs. (d) Removal of background voxels from the scene. (e) The remaining voxels corresponding to a human and a moved chair are segmented after small segments are removed.

If a human is not found by the proposed face detection algorithm, then color histogram matching is performed. Thus, before tracking can be employed, a face must be found at least once. The color histogram of each new object is $\overrightarrow{\Psi}_l$ $(1 \leq l \leq \mathrm{L})$ and the previously detected human is $\overrightarrow{\Psi}_{\mathrm{P}}$. The similarity between $\overrightarrow{\Psi}_l$ and $\overrightarrow{\Psi}_{\mathrm{P}}$ is

$$\psi_l = \frac{\sum_{\mathrm{i}=1}^{\mathrm{Q}^2} \mathrm{minimum}\left(\overrightarrow{\Psi}_{\mathrm{P}}(\mathrm{i}), \overrightarrow{\Psi}_l(\mathrm{i})\right)}{\sum_{\mathrm{i}=1}^{\mathrm{Q}^2} \mathrm{maximum}\left(\overrightarrow{\Psi}_{\mathrm{P}}(\mathrm{i}), \overrightarrow{\Psi}_l(\mathrm{i})\right)}.$$

The maximum match is $\psi_{max} = \max_{l\in\{1,\ldots,L\}} \psi_l$ and the maximum index is $\varphi_{max} = argmax_{l\in\{1,\ldots,L\}}\psi_l$. If ψ_{max} is greater than 0.5, it is a proper match, figure 6.2. If a proper match is found, the color histogram is updated as

$$\overrightarrow{\Psi}_{\mathrm{P,t+1}} = \alpha\overrightarrow{\Psi}_{\mathrm{P,t}} + (1-\alpha)\overrightarrow{\Psi}_{\varphi_{max},t},$$

where $\alpha \in [0,1]$. We picked an α of 0.65.

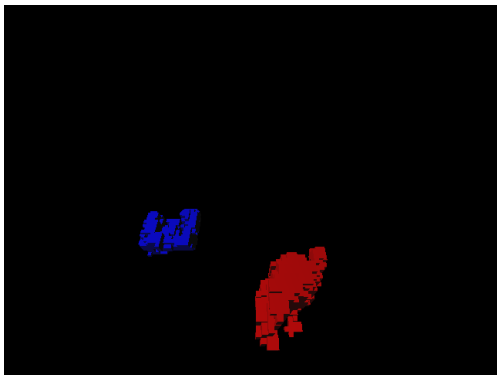


Figure 5.2: Proper labeling of changed objects from figure 5.1 (e). The red object on the right is the human; the blue object on the left is a moved chair.

Voxels representing the human in the current scene have now been determined, O_H. The voxel space $E_\mu = E_A - O_H$ can then be substituted back into the background update equation. In this fashion, the system is able to update the representation of the background model.

6. EXPERIMENTS

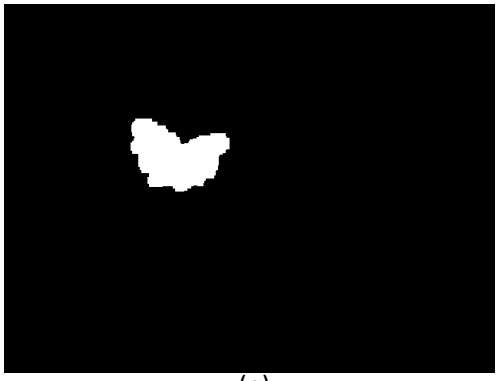
Few parts of the system defined in this dissertation can be validated using mathematical proofs. Therefore, the success of this work is based on the system's abilities relative to previously defined algorithms. Accuracy in this situation is based on the ability to properly identify humans in a scene while correctly classifying the background. Experiments consist of multiple users performing common day to day tasks in a workspace simulating a living area. Measures are presented in order to compute the accuracy of the proposed system. Humans segmented from the three-dimensional space are back-projected into two-dimensional image space and are tested against ground truth and previous image-space background modeling algorithms. Three-dimensional accuracy is also tested by comparing the human segment output of this system compared to hand segmented three-dimensional ground truth.

6.1. 2D Experiments

Image space ground truth can be created relatively easily by hand. The majority of comparative studies are also done only in image space. The three dimensional output of our system can be projected back into the original image space, and then directly compared to image space algorithms. Figure 6.1 shows one such projection and the associated hand segmented ground truth.

(a) (b)

(c) (d)



(e)

Figure 6.1: Experimental setup for a frame moments after a lighting change. (a) The original image. (b) The hand segmented ground truth. (c) The output of GMM **[**1**]**. (d) The output of **[**11**]**. (e) The output of the system defined in this dissertation.

Six sequences, each with 1200 frames, were collected at a rate of four frames per second to demonstrate a range of possible activities in a single person scene. Each sequence is five minutes long. Two subjects are used throughout the experiments. Subject one is used in sets 1, 2, 4 and 6, while subject two is used in sets 3 and 5. All data sets, ground truth and output from the system defined in this dissertation can be found at http://cirl.missouri.edu/vision/.

Experiment one is arguably the easiest set to process. The subject enters the room, sits on a couch, stands then sits on another couch and leaves. No furniture is moved, and the lighting is constant. Our method returns over 90% true positive with 99% true negative. Li's algorithm returns a moderately accurate 74% true positive accuracy with a nearly perfect true negative. The GMM performs poorly because the subject sits still several times and is adapted into the background. This problem with adaptation rate is present in any low level background update algorithm.

Table 6.2. Confusion matrix results of experiment one. Subject walks into the room, sits on a couch, stands, sits on another couch then leaves the scene.

		Ground Truth	
		Foreground	Background
Our Method	Foreground	90.5%	1%
	Background	9.5%	99%
GMM [**1**]	Foreground	57.1%	2.3%
	Background	42.9%	97.7%
Li [**11**]	Foreground	74%	0.1%
	Background	26%	99.9%
Zivkovic [**12**]	Foreground	61.0%	.2%
	Background	39.0%	99.8%

The second sequence is the same as the first, but the subject has a book that he alternates between reading and placing on the table. The accuracies of the outputs are very similar to the first sequence, but slightly lower for the GMM and Li's algorithm. This is because the book is incorrectly classified as foreground when it is placed on the table.

Table 6.2. Confusion matrix results of experiment two. The subject repeats the same tasks as the first experiment, but this time brings in a book. The subject alternates between reading the book and laying it on a table.

		Ground Truth	
		Foreground	Background
Our Method	Foreground	92.3%	1%
	Background	7.7%	99%
GMM [**1**]	Foreground	54.9%	1.5%
	Background	45.1%	98.5%
Li [**11**]	Foreground	72.2%	0.2%
	Background	27.8%	99.8%
Zivkovic [**12**]	Foreground	59.5%	.2%
	Background	40.5%	99.8%

The third sequence tests a common situation where a subject moves furniture, then sits down. Li's algorithm performs similar to the first two sequences, but the GMM output is quite different. At first glance, the GMM appears to perform much better. Because the subject pauses for shorter amounts of time while sitting, he does not adapt into the background and therefore the GMM achieves an 82% true positive accuracy. But, the GMM achieves only an 89% true negative accuracy because it incorrectly labels pixels in the moved chairs as change. Our higher level system is able to recognize that moved

furniture is unrelated to the human and therefore does not mark it as a significant change.

Table 6.3. Confusion matrix results of experiment three. The subject moves from chair to couch to chair in a fashion similar to the previous sequences, but moves furniture while walking through the room.

		Ground Truth	
		Foreground	Background
Our Method	Foreground	82.8%	1.6%
	Background	17.2%	98.4%
GMM [**1**]	Foreground	82.7%	10.6%
	Background	17.3%	89.4%
Li [**11**]	Foreground	73.2%	0.8%
	Background	26.8%	99.2%
Zivkovic [**12**]	Foreground	85.3%	2.0%
	Background	14.7%	98.0%

Experiment four displays one of the greatest advantages of our system over the previous algorithms. This sequence demonstrates the affect of lighting change on the classification. The sequence is identical to the first sequence, but with several lighting changes. The GMM and Li's algorithm return significantly lower true negative

accuracies, because large portions of the space are incorrectly classified as foreground due to the lighting change. This also artificially inflates the true positive results for both algorithms. In contrast, our system is nearly unaffected by the lighting changes.

Table 6.4. Confusion matrix results of experiment four. In this sequence, the subject performs a similar set of actions, moving from seat to seat, but the lighting changes drastically throughout the sequence.

		Ground Truth	
		Foreground	Background
Our Method	Foreground	88.8%	1.3%
	Background	11.2%	98.7%
GMM [**1**]	Foreground	87.5%	13.9%
	Background	12.5%	86.1%
Li [**11**]	Foreground	78%	7.3%
	Background	22%	92.7%
Zivkovic [**12**]	Foreground	90.4%	1.4%
	Background	9.6%	98.6%

Television screens often lead to problems for image space algorithms, but are trivial for the system defined in this dissertation. In this sequence the subject changes the lighting, sits on the couch, turns on the television, and begins watching. The true

negative is lower for the GMM and Li's algorithms, because they have trouble adapting to the change in lighting and continually classify the television screen as foreground. Our algorithm continues to have an accurate true negative rate, but the true positive is underperforming. Upon further review, it was determined that poor true positive accuracy was due to improper stereo correspondence on the subject's legs as is visible in Figure 6.1 (e).

Table 6.5. Confusion matrix results of experiment five. This sequence simulates the subject watching television

		Ground Truth	
		Foreground	Background
Our Method	Foreground	75.7%	1.5%
	Background	24.3%	98.5%
GMM [**1**]	Foreground	66.2%	12.5%
	Background	32.8%	87.5%
Li [**11**]	Foreground	56.4%	0.9%
	Background	43.6%	99.1%
Zivkovic [**12**]	Foreground	61.2%	3.4%
	Background	38.8%	96.6%

The final experiment combines all previous tests into one sequence. This includes, sitting for periods watching television, changes in lighting, manipulating a book and moving furniture. Again, the GMM and Li's algorithm underperform with lower accuracies in all categories. Our system achieves a 83% true positive and a 98% true negative.

Table 6.6. Confusion matrix results of experiment six. A sequence that combines all possibilities. A book is brought into the scene, the lighting changes several times, the furniture is moved and a television is used.

		Ground Truth	
		Foreground	Background
Our Method	Foreground	83.8%	1.5%
	Background	16.2%	98.5%
GMM [**1**]	Foreground	66.3%	10.8%
	Background	33.7%	89.2%
Li [**11**]	Foreground	74.9%	4.2%
	Background	25.1%	95.8%
Zivkovic [**12**]	Foreground	79.9%	6.2%
	Background	20.1%	93.8%

The combined statistics of all sequences displays the significant advantage of our system over the previous systems. The true positive rate is over ten percent higher than either algorithm, while true negative is also higher than each. It should also be noted that the ground truth data had an average of 6576 foreground pixels and 300624 background pixels per test image. So, a 1% change in foreground classification accuracy results in a change of roughly 66 pixels, while a 1% change in background classification accuracy results in a change of 3006 pixels.

Table 6.7. The combined confusion matrix statistics of all six experiments.

		Ground Truth	
		Foreground	Background
Our Method	Foreground	84.9%	1.3%
	Background	15.1%	98.7%
GMM [**1**]	Foreground	70.4%	8.6%
	Background	29.6%	91.4%
Li [**11**]	Foreground	71.0%	2.3%
	Background	29.0%	97.7%
Zivkovic [**12**]	Foreground	74.0%	2.2%
	Background	26.0%	97.8%

It should also be noted that the result of our system is a three-dimensional model of the human and moved objects. This provides a much richer world-space representation for subsequent higher level processing.

6.2. 3D Experiments

Though the previous section displayed the performance of this system over common image space algorithms, the greatest advantage of this system is the output of three-dimensional information. Accuracy in this case is determined as the correct labeling of the human in voxel space. Ground truth data was labeled by hand for the sixth sequence of data described in the previous section. This sequence contains all real-world possibilities that image space algorithms have difficulty accurately labeling such as lighting changes, a television screen and moving background objects.

In contrast to the hand segmentation of the 2D images in the previous section, creating three-dimensional ground truth is difficult and cumbersome. For a given time step, the scene is rendered to voxel space with the voxels colored according to the stereo images. Figure 6.2 shows this color voxel representation of a single time step. As previously described, the scene size is 128x128x64 voxels. This space can conversely be thought of as 64 individual 128x128 pixel images at unique heights between the floor and 3.2 meters at every 5 cm. Figure 6.3 displays some of the images related to the scene in figure 6.2. The human volume is then segmented by hand from these images, figure 6.4.

The 64 segmented images are then packaged back into a 128x128x64 voxel representation of the 3D human voxel volume, figure 6.5.

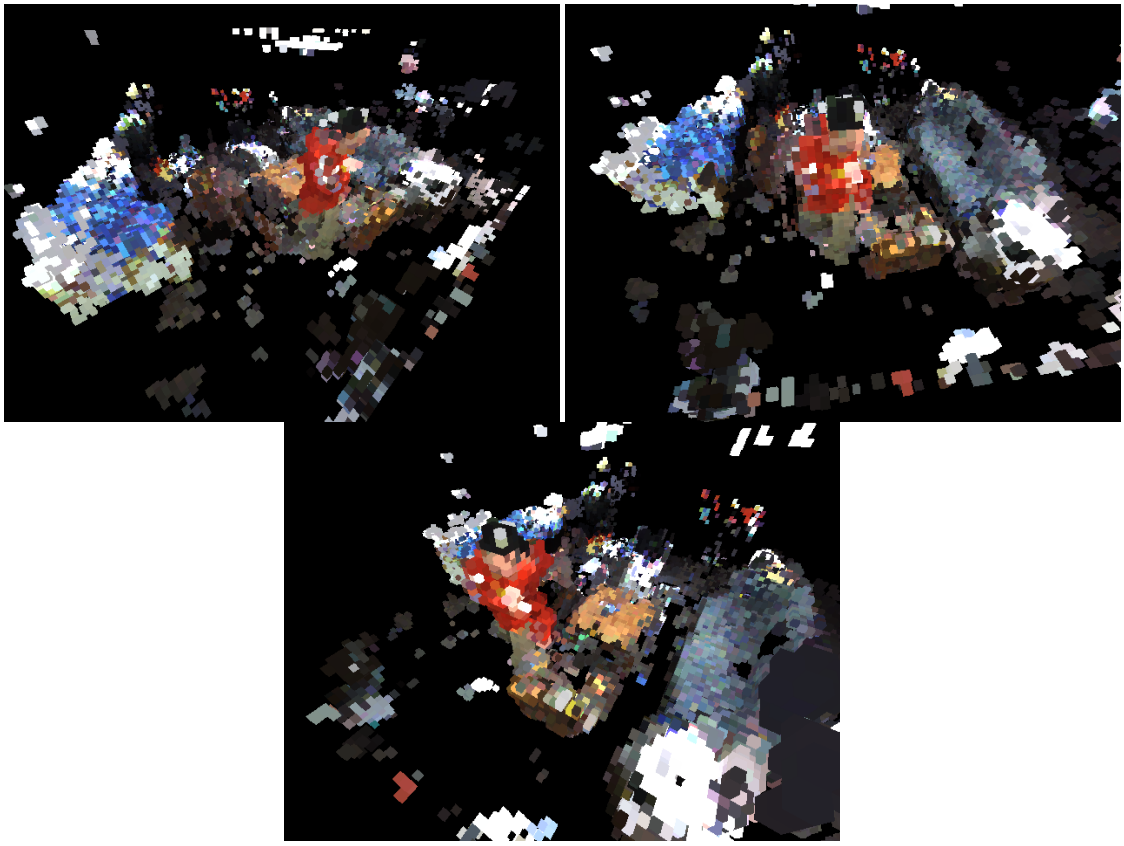
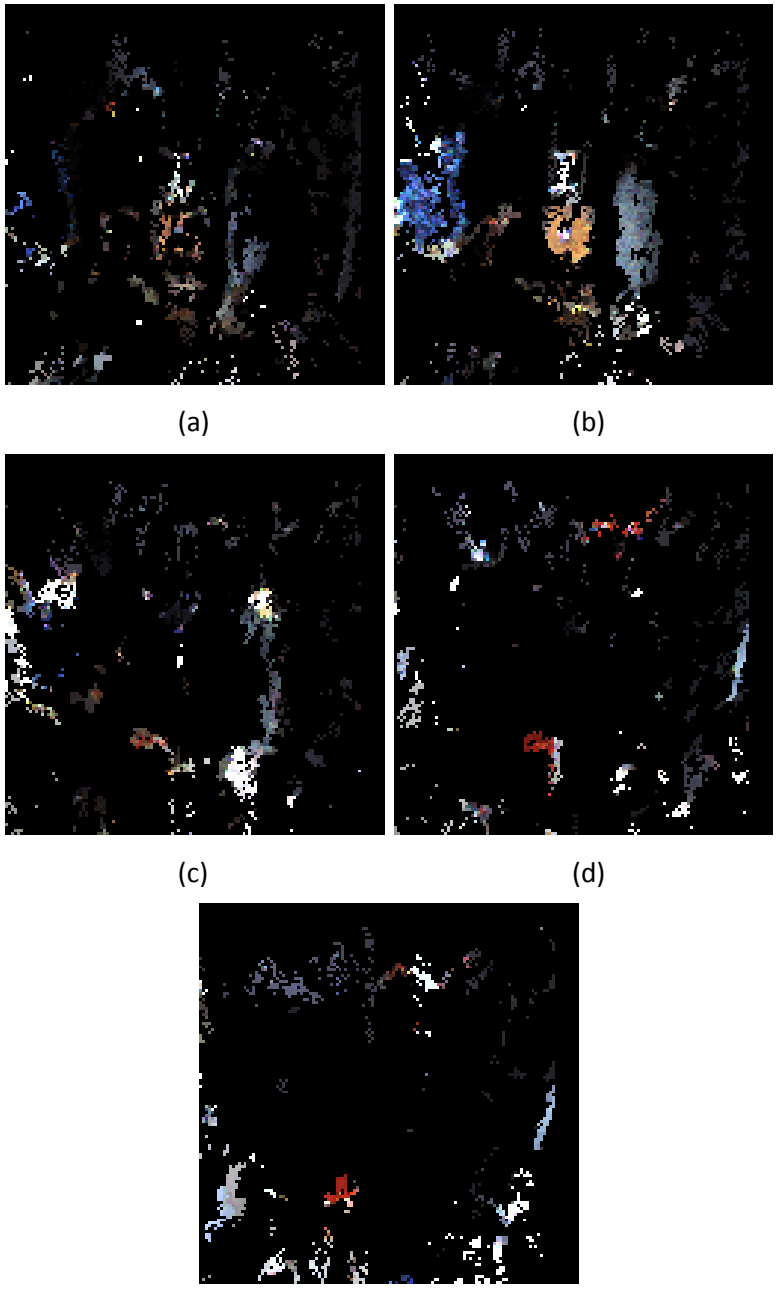


Figure 6.2: Color voxel representation of the scene at a single time step. The human is wearing a red shirt. The couch is a dark blue while the bed is a brighter blue. The brown table is in the center of the room.

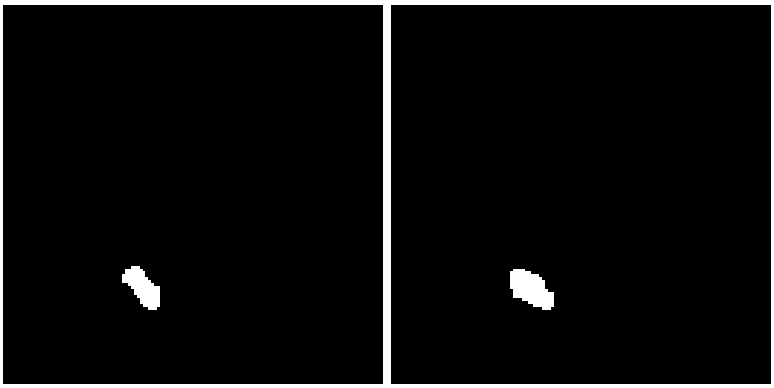


(a) (b)

(c) (d)

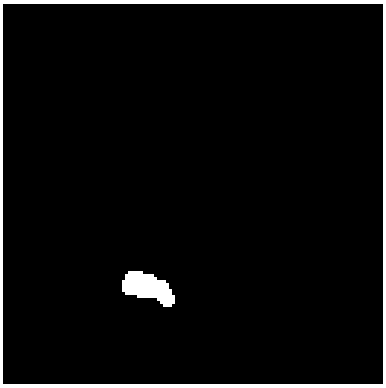
(e)

Figure 6.3: Five image slices of the voxelized room at different heights. (a) At 0.35 meters on the bottom of objects are visible. (b) At 0.7 meters the couch can be seen on the right, the table and chairs in the middle and the bed on the left. (c) At 1.05 meters the person's waist are visible. (d) At 1.4 meters the human's red shirt shows up. (e) At 1.75 meters the shoulders of the person are visible.

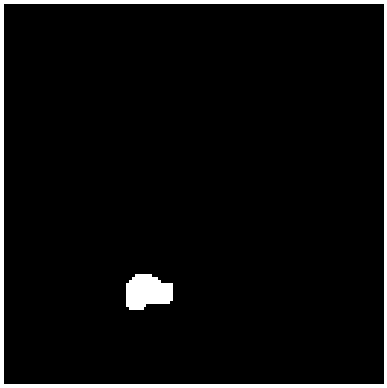


(a)

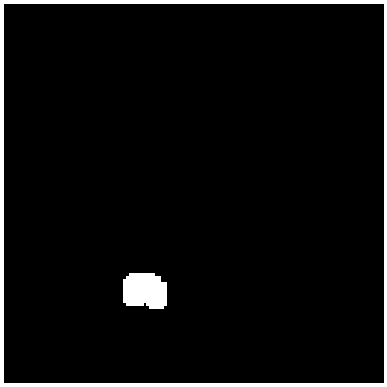
(b)



(c)



(d)



(e)

Figure 6.4: Hand segmentation of the human from the images in figure 6.3. The stride of the person is noticeable in (a) and (b). In (c) the shape of the waist becomes obvious. In (d) and (e) the chest and shoulders of the subject become visible.

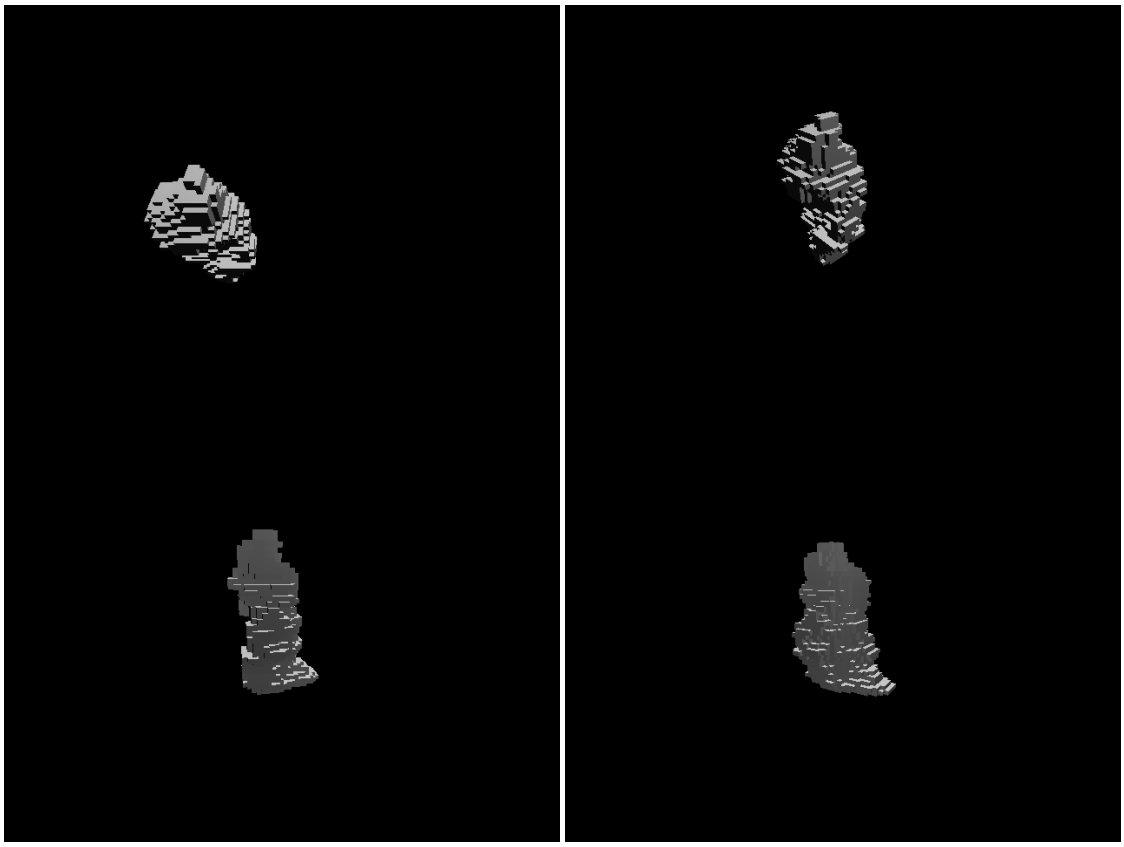


Figure 6.5: Three-dimensional voxel representation of the human after hand segmentation. The four unique viewpoints display the three dimensional shape of the hand segmented human.

Because the three-dimensional hand segmentation is so time consuming and cumbersome, only seven frames of the 1200 in the sequence have been hand segmented as ground truth. Though only seven frames are used, each frame was chosen specifically to test real-world problems such as lighting changes and moved

background objects. Therefore, though this test set is small, it is a good representational cross-section of challenges encountered in real-world operations.

A second set of metrics are tested based on features extracted from the data. The height and centroid of the human volume is computed from the output of the system and then compared to the ground truth height and centroid. These metrics display how the system's output can be used to accurately build usable features for higher level processes.

The experiments are broken into two sequences. The first test uses the entire sequence of data starting with the person outside the scene. Starting with the human outside the scene is a common assumption made by change detection algorithms. Of course, in real-world settings, it will never be guaranteed that the scene is free from a foreground subject. Therefore, the second test uses the same sequence, but starts several hundred frames in while the human is inside the scene. This tests the bootstrapping ability of the system.

6.2.1. Experiments Using the Entire Data Sequence

6.2.1.1. Volumetric Experiment

In the first experiment the entire sequence is processed using all 1200 frames of data beginning with the person outside the scene. The system outputs the volumetric location of the human at each time step of the sequence. The true positives, true

negatives, false positives and false negatives are described for each of the seven segmented ground truth time steps.

The first test frame is of the person walking through the scene. Figure 6.6 displays the original image and the segmented human output from the system. As can be observed, the human is accurately segmented from the scene.

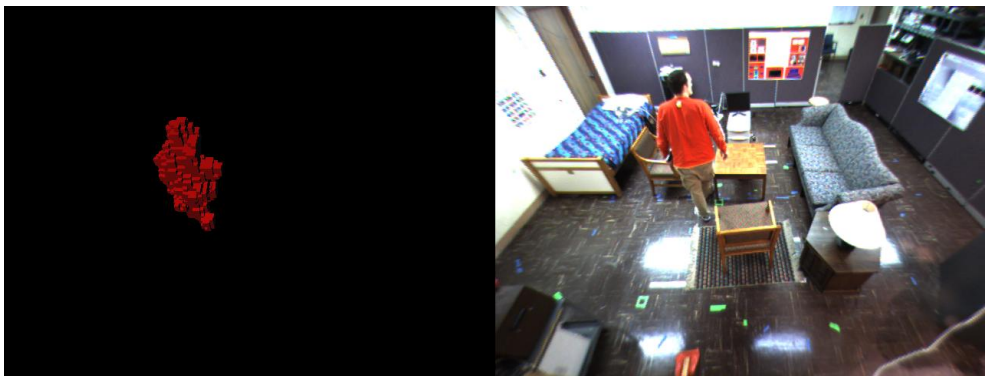


Figure 6.6: The first test frame. The output of the system is shown in the left image.

The confusion matrix for this frame, table 8, tells an interesting story that is consistent through all test frames. The system accurately segments only around half of the true positive voxels of the person and around half of the person is missed as false negative locations. The true negative rate is nearly perfect with very few false positives.

Table 6.8. Test frame one confusion matrix statistics.

Confusion Matrix Frame 1		Ground Truth	
		Foreground	Background
Our System	Foreground	53.76%	0.02%
	Background	46.24%	99.98%

The initial inspection of this confusion matrix was very disconcerting, but upon further review, the reason for the low true positive accuracy became clear. The three dimensional model output by the system is slightly smaller than the ground truth model. Figure 6.7 graphically displays the size difference between the output and the ground truth. The true positive blue voxels output by the system are surrounded by red false negative voxels. So that the volume of the true positive blue voxels are not completely obscured by the thin layer of the false negative volume, the red false negative voxels are drawn at $1/8^{th}$ scale.

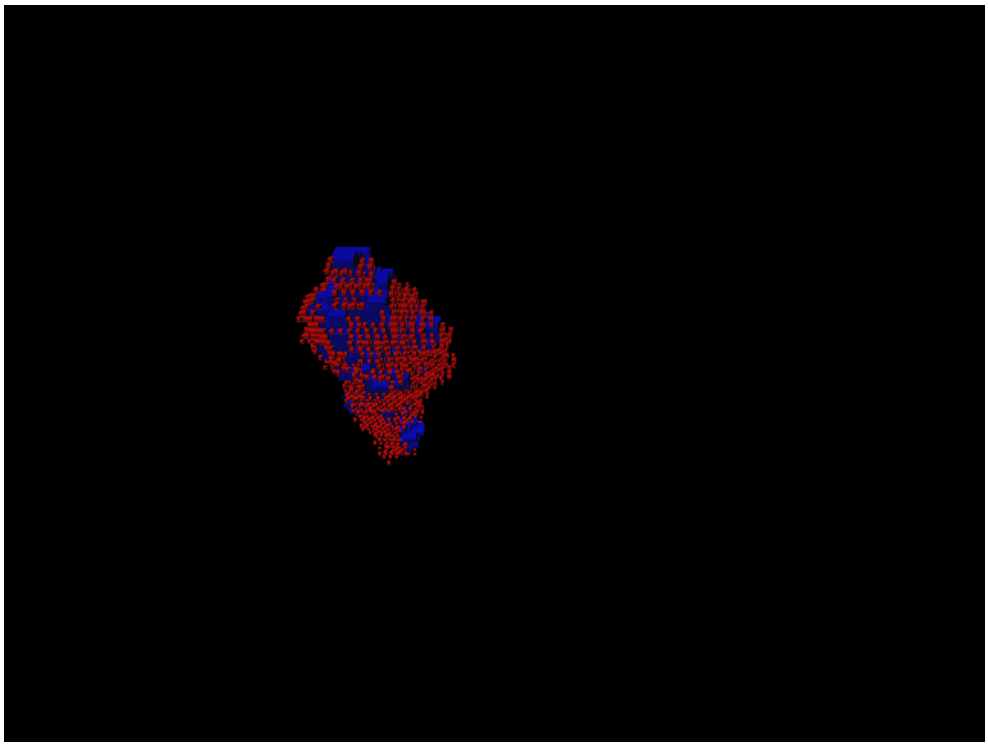


Figure 6.7: A graphical description of the error related to the output of the system for the first test frame. True positive are represented as blue voxels, while false negatives are represented as smaller red voxels.

The true positive accuracy could have been increased by morphologically dilating the human voxels output from the system. It was decided that morphological dilation would not be used because it would be done strictly to increase the true positive rate while not adding any more useful information. The accuracies of two features associated with this useful information, height and centroid, are described in the next section.

It should also be noted that the hand segmentation of the ground truth required significant human judgment in determining the true foreground boundary. These human volumes appear to be slightly oversized when viewed in voxel space.

The second frame has the subject sitting in a chair with his legs up on the table figure 6.8. The confusion matrix is shown in table 6.9. As figure 6.9 shows, the true positive rate is low because a large portion of the feet and legs are missed. The reason the legs are missed is likely because of error in the registration between the two stereo pair cameras. If the error between the two stereo pairs is too great, some voxels will be incorrectly carved away after the two voxel representations are intersected. In the thicker regions of the body such as the torso, many voxels still remain. But, thinner volumes such as legs and feet can be completely removed due to some of the system's morphological operations.

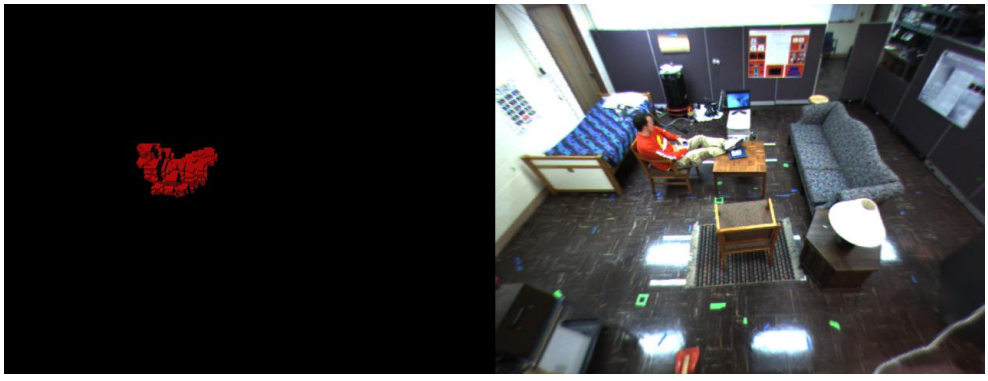


Figure 6.8: The second test frame. The output of the system is shown in the left image.

Table 6.9. Test frame two confusion matrix statistics.

Confusion Matrix Frame 2		Ground Truth	
		Foreground	Background
Our System	Foreground	36.88%	0.04%
	Background	63.12%	99.96%

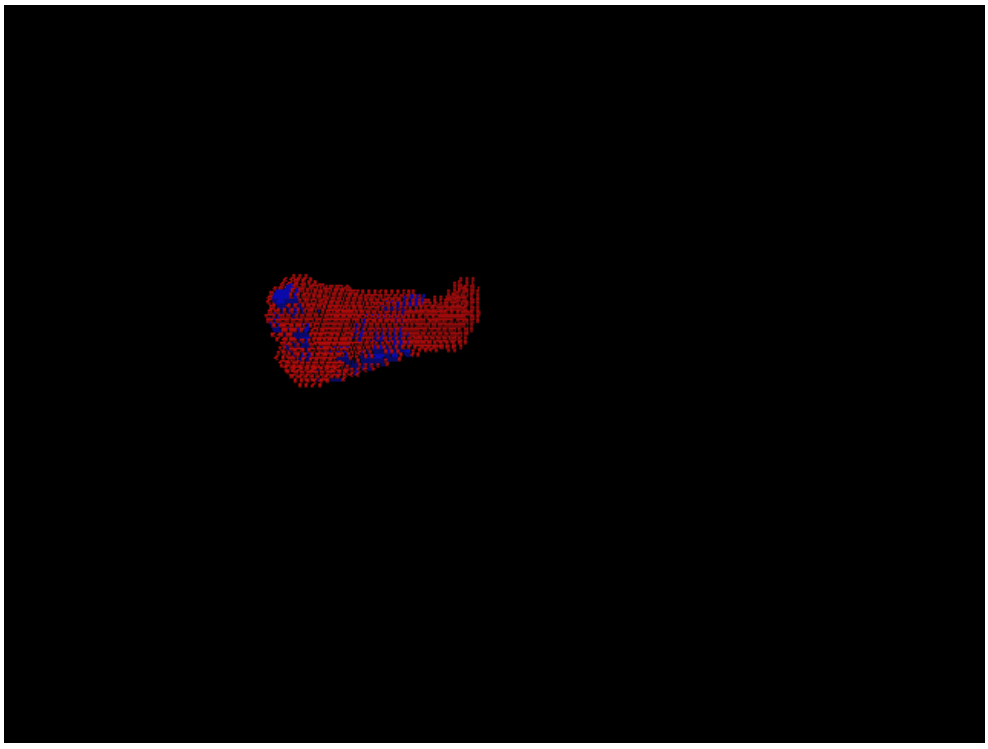


Figure 6.9: A graphical description of the error related to the output of the system for the second test frame. True positive are represented as blue voxels, while false negatives are represented as smaller red voxels.

The third frame is taken when the person is again walking. The output of the human voxels in figure 6.10 is again significantly undersized. In this case, the specular highlight on the floor may have had something to do with improper stereo registration.

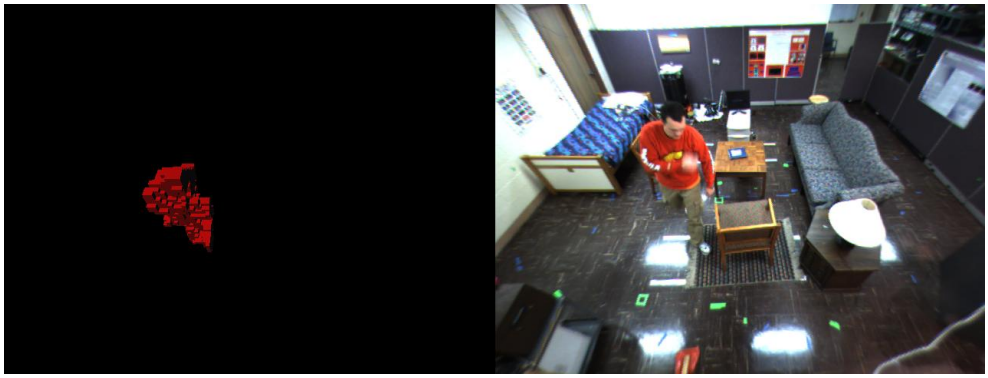


Figure 6.10: The third test frame. The output of the system is shown in the left image.

Table 6.10. Test frame three confusion matrix statistics.

Confusion Matrix Frame 3		Ground Truth	
		Foreground	Background
This System	Foreground	37.30%	0.0%
	Background	63.70%	100.00%

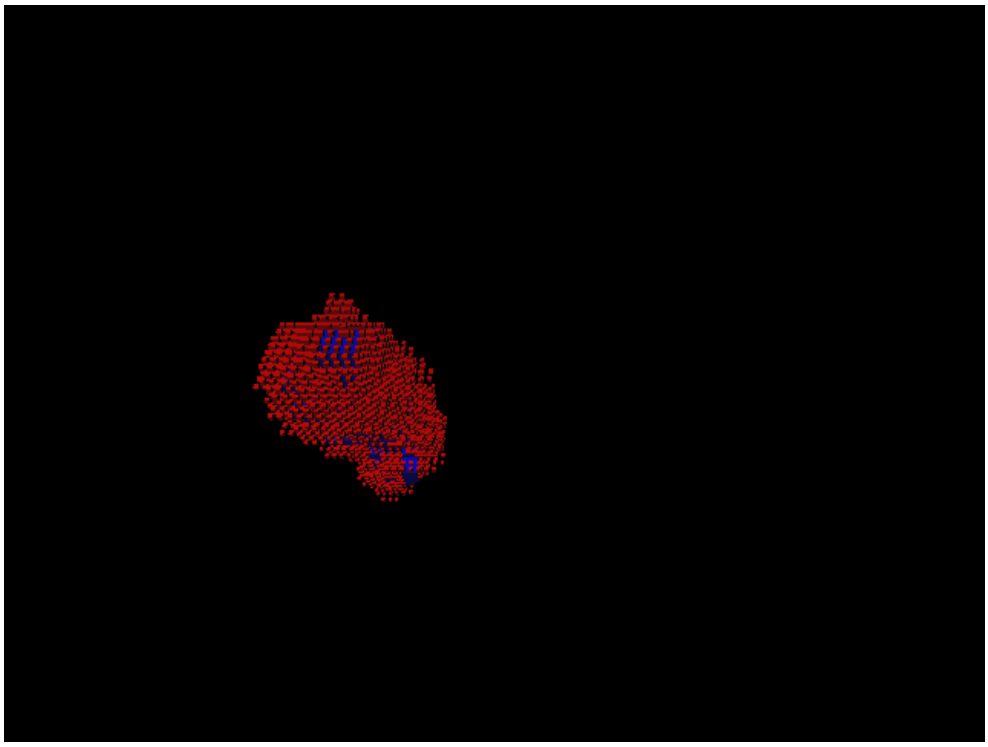


Figure 6.11: A graphical description of the error related to the output of the system for the test frame. True positive are represented as blue voxels, while false negatives are represented as smaller red voxels.

The fourth test frame is arguably the greatest example of the abilities of this system, figure 6.12. A chair has been moved by the person and is therefore recognized as changed, but is classified as a nonhuman object and quickly adapted into the background model. Again, the volume of the human is undershot.

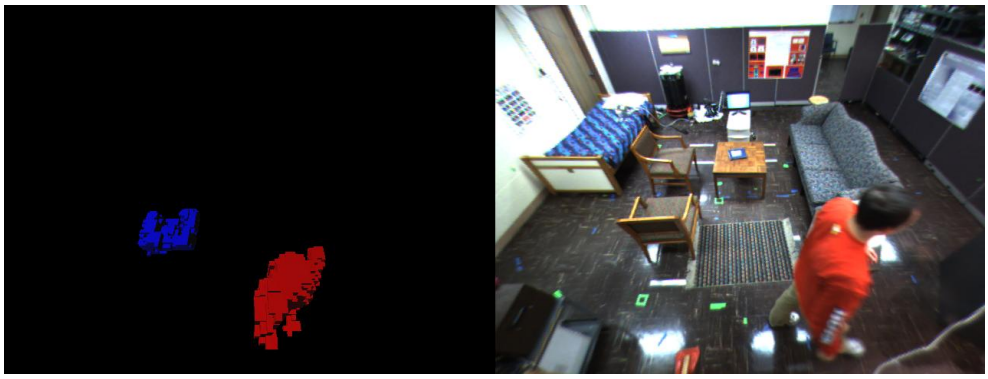


Figure 6.12: The fourth test frame. The output of the system is shown in the left image.

Table 6.11. Test frame four confusion matrix statistics.

Confusion Matrix Frame 4		Ground Truth	
		Foreground	Background
Our System	Foreground	41.62%	0.03%
	Background	59.38%	99.97%

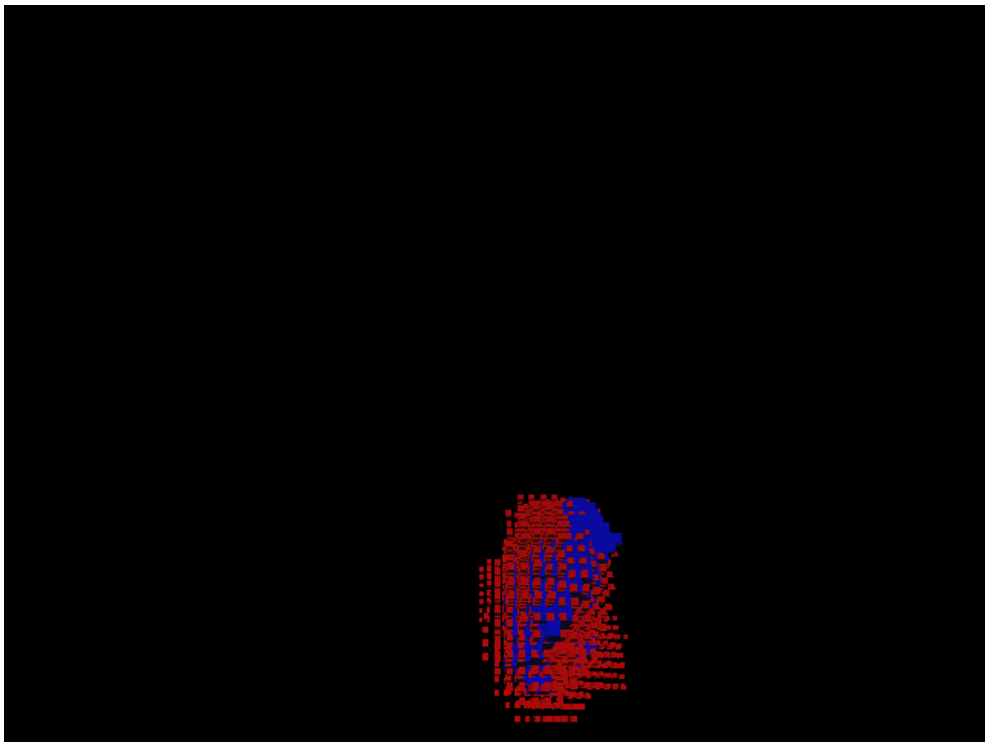


Figure 6.13: A graphical description of the error related to the output of the system for the test frame. True positive are represented as blue voxels, while false negatives are represented as smaller red voxels.

In frame five the human is walking between the couch and the table. The output of the system shown in figure 6.14 displays the ability of the system to accurately segment the human from nearby nonhuman objects. The true positive accuracy is nearly two thirds of the foreground voxels, which adds more evidence to the conjecture that the cameras are slightly misregistered. This is because the greatest accuracy occurs in the middle of the scene where measurements were taken for the initial camera registration. As the subject moves toward the boundaries of the scene, the accuracy decreases.

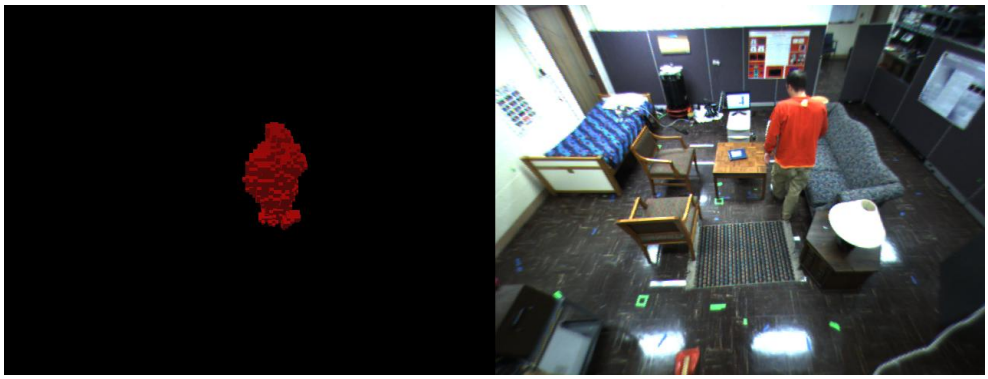


Figure 6.14: The fifth test frame. The output of the system is shown in the left image.

Table 6.12. Test frame five confusion matrix statistics.

Confusion Matrix Frame 5		Ground Truth	
		Foreground	Background
Our System	Foreground	63.43%	0.04%
	Background	36.57%	99.96%

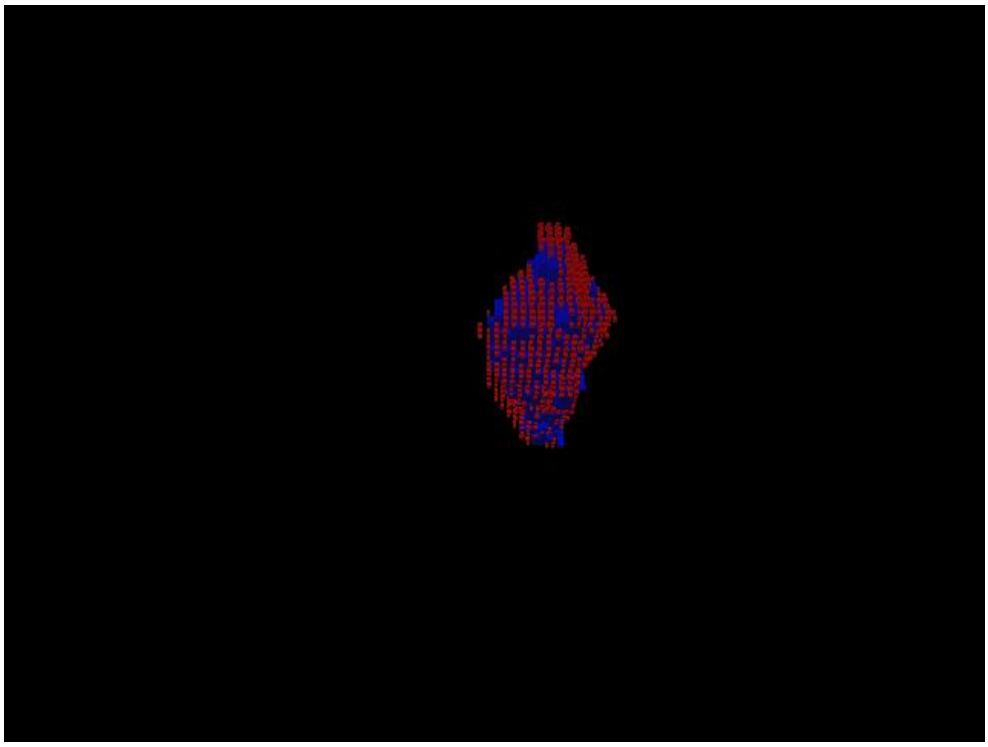


Figure 6.15: A graphical description of the error related to the output of the system for the test frame. True positive are represented as blue voxels, while false negatives are represented as smaller red voxels.

The sixth test frame again tests an important real-world problem figure 6.16. As the person sits down, the couch moves back a few inches. In most change detection algorithms this would result in a great deal of change voxels as output. This system does not register the couch's small movement as significant enough to warrant a detected change. It should be noted that part of the person's legs sink into the couch and are therefore not registered in the change volumes. By missing some of the legs of the person, the true positive accuracy is again a bit low as shown in table 6.16.

Figure 6.16: The sixth test frame. The output of the system is shown in the left image.

Table 6.13. Test frame six confusion matrix statistics.

Confusion Matrix Frame 6		Ground Truth	
		Foreground	Background
Our System	Foreground	33.00%	0.05%
	Background	67.00%	99.95%

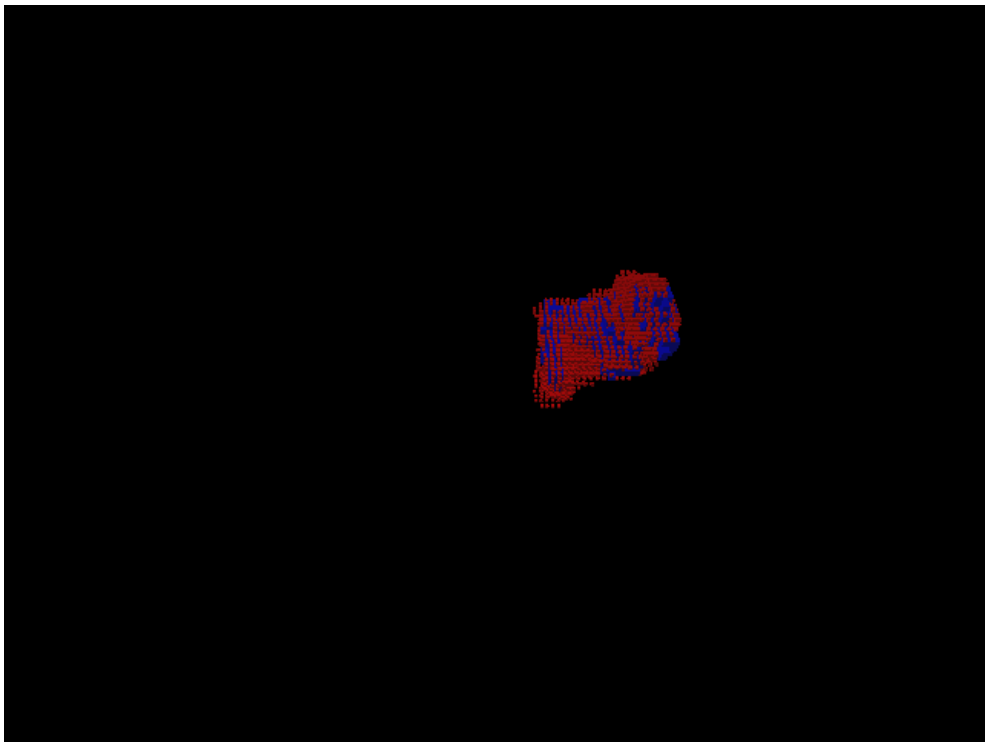


Figure 6.17: A graphical description of the error related to the output of the system for the test frame. True positive are represented as blue voxels, while false negatives are represented as smaller red voxels.

The final test frame again has the person sitting in a chair. This time step is shortly after the person moves the chair, so the human and chair are recognized as a single object, figure 6.18. The true positive rate is again around two thirds of the foreground voxels, but the false positive rate is slightly higher due to the chair also being classified as part of the human, table 6.14.

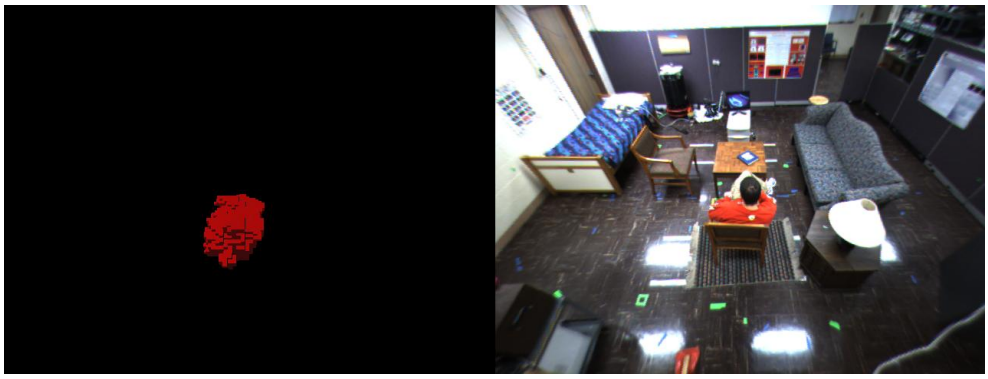


Figure 6.18: The seventh test frame. The output of the system is shown in the left image.

Table 6.14. Test frame seven confusion matrix statistics.

Confusion Matrix Frame 7		Ground Truth	
		Foreground	Background
Our System	Foreground	67.02%	0.08%
	Background	32.98%	99.92%

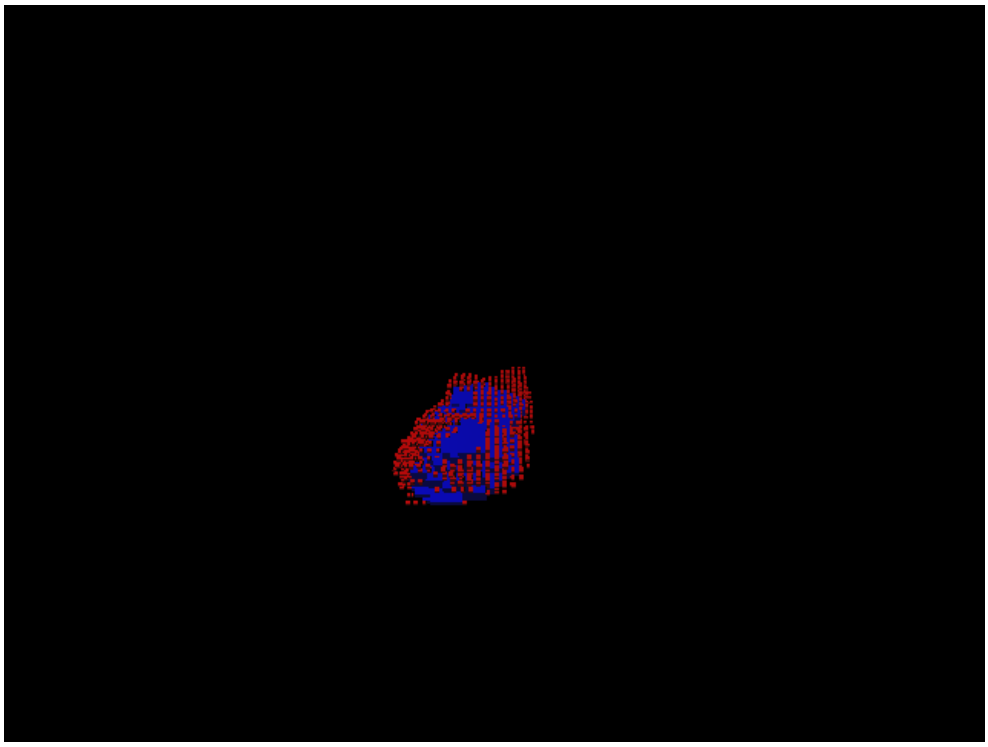


Figure 6.19: A graphical description of the error related to the output of the system for the test frame. True positive are represented as blue voxels, while false negatives are represented as smaller red voxels.

The accuracy of the system across all test images is shown in table 6.15. Though the true positive rate is slightly below 50%, the figures graphically display the system output fits the form of the human, but is slightly smaller than desired. The accuracy would improve if the cameras were registered more accurately in three-dimensional space. Again, this accuracy could be increased synthetically using morphology, but I believe that it is important to show the raw capabilities of this system.

Table 6.15. Confusion matrix statistics for all test data combined.

Confusion Matrix All Frames		Ground Truth	
		Foreground	Background
Our System	Foreground	47.67%	0.04%
	Background	52.33%	99.96%

6.2.1.2. Feature Extraction Accuracy

The output of this system will be used as a reliable input source for the location of a person in an environment. While this higher level system could use the entire voxel space directly, it is more likely that salient features will be extracted from the voxel data. Two such features are the height and centroid of the human.

The height of the person is often used to determine the posture of the person. When the height is high, the person is standing or walking. If the height is low, the person may have fallen or be lying on the ground. If the height is somewhere in between, the person is probably sitting in a chair or on a couch.

For the seven test frames, the height of the output of the system was compared to the height of the ground truth. Tables 6.16 and 6.17 show the error of height in voxels and cm. The height is usually accurate within a couple voxels, but there was a larger error in frame four. This larger error was due to the error in the registration of the two stereo

pair cameras. In frame four, the person is a great distance from camera two, leading to greater error.

Table 6.16. Statistics showing the error between the height of the human output from the system and that of the ground truth, measure in voxels.

Error of Height (voxels)	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6	Frame 7	Mean	Median
Error	1.0	2.0	3.0	4.0	2.0	2.0	1.0	2.1	2.0

Table 6.17. Statistics showing the error between the height of the human output from the system and that of the ground truth, measure in centimeters.

Error of Height (cm)	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6	Frame 7	Mean	Median
Error	5.0	10.0	15.0	20.0	10.0	10.0	5.0	10.7	10.0

The centroid of the character is again a useful feature for activity recognition. Knowing where in the environment the person is currently located can help classify the current state/activity as well as define the proximity to nearby objects. The output of this system is usually within a few voxels or around 12 cm. The significant outliers are related to frame three and six when the person is sitting in the chair and on the couch. Because portions of the legs are missing in each case, the mean is incorrectly shifted.

Table 6.18. Statistics showing the error between the centroid of the human output from the system and that of the ground truth, measure in voxels.

Error Distance From Centroid (voxels)	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6	Frame 7	Mean	Median
Error in X	0.2	0.9	0.2	1.2	0.2	3.5	1.4	1.1	0.9
Error in Y	0.9	2.5	0.4	1.4	0.6	4.1	0.7	1.5	0.9
Error in Z	1.2	2.0	1.6	1.0	1.0	0.4	1.2	1.2	1.2
Error	1.5	3.4	1.6	2.1	1.2	5.4	1.9	2.4	1.9

Table 6.19. Statistics showing the error between the centroid of the human output from the system and that of the ground truth, measure in centimeters.

Error Distance From Centroid (cm)	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6	Frame 7	Mean	Median
Error in X	1.0	4.4	1.0	6.1	0.90	17.4	6.9	5.4	4.4
Error in Y	4.5	12.9	1.9	6.8	3.1	20.5	3.5	7.6	4.5
Error in Z	5.8	10.0	7.8	4.9	5.2	2.2	5.9	6.0	5.8
Error	7.4	16.9	8.1	10.3	6.2	27.0	9.7	12.2	9.7

It is also important to show the stability of the features extracted over time. The next series of figures demonstrate the consistency of the output of this system. Figure 6.20 displays the height of the segmented human throughout the sequence. It is easy to see when the subject is walking/standing or sitting by the abrupt change in height.

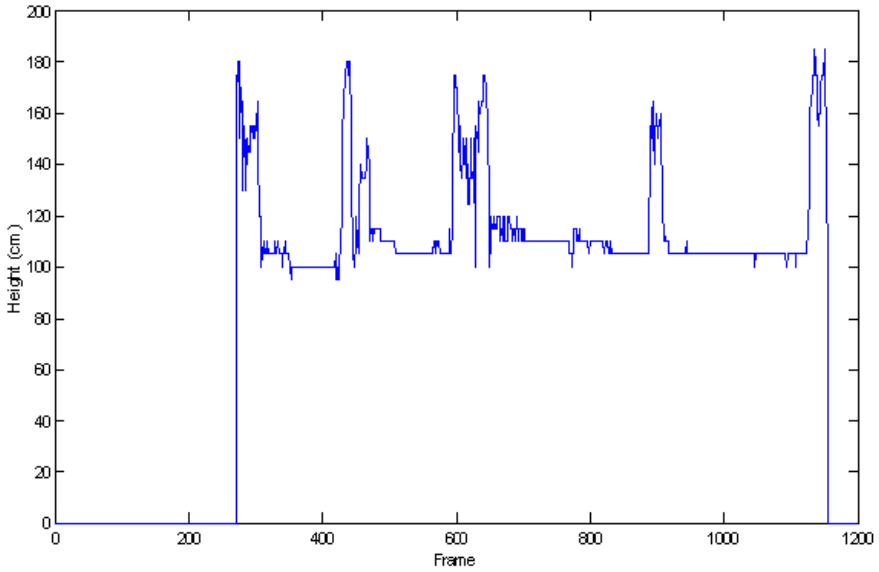


Figure 6.20: A graph displaying the height of the human output from the system. The feature is very stable and the transitions between sitting and standing are obvious.

It is also interesting to note the mean of the centroid in the z (height) dimension, figure 6.21. This feature can be used in tandem with the height feature to determine when the person is lying down on the couch or in the case of this sequence, when the subject rests his feet onto the coffee table.

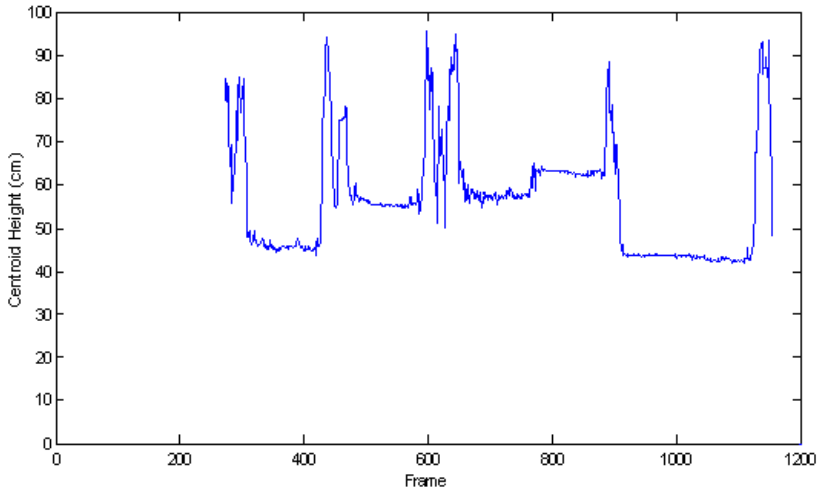


Figure 6.21: A graph displaying the height of the centroid of the human output from the system. The feature is also very stable and the transitions between sitting and standing are obvious. A change in position even while sitting can be noticed around frame 770.

The accuracy of the centroid has been shown earlier in this section. It is also important to have a smooth trajectory of the person's location through the scene over time. Figure 6.22 displays the path of the person through the scene over time with respect to the ground plane. This figure shows the sequence broken into subsequences of the subject moving from one location to another such as chair to couch. The trajectories are very smooth with the only jumps can be attributed to quick movements of the subject such as leaning over to turn on the television in figure 6.22 (b).

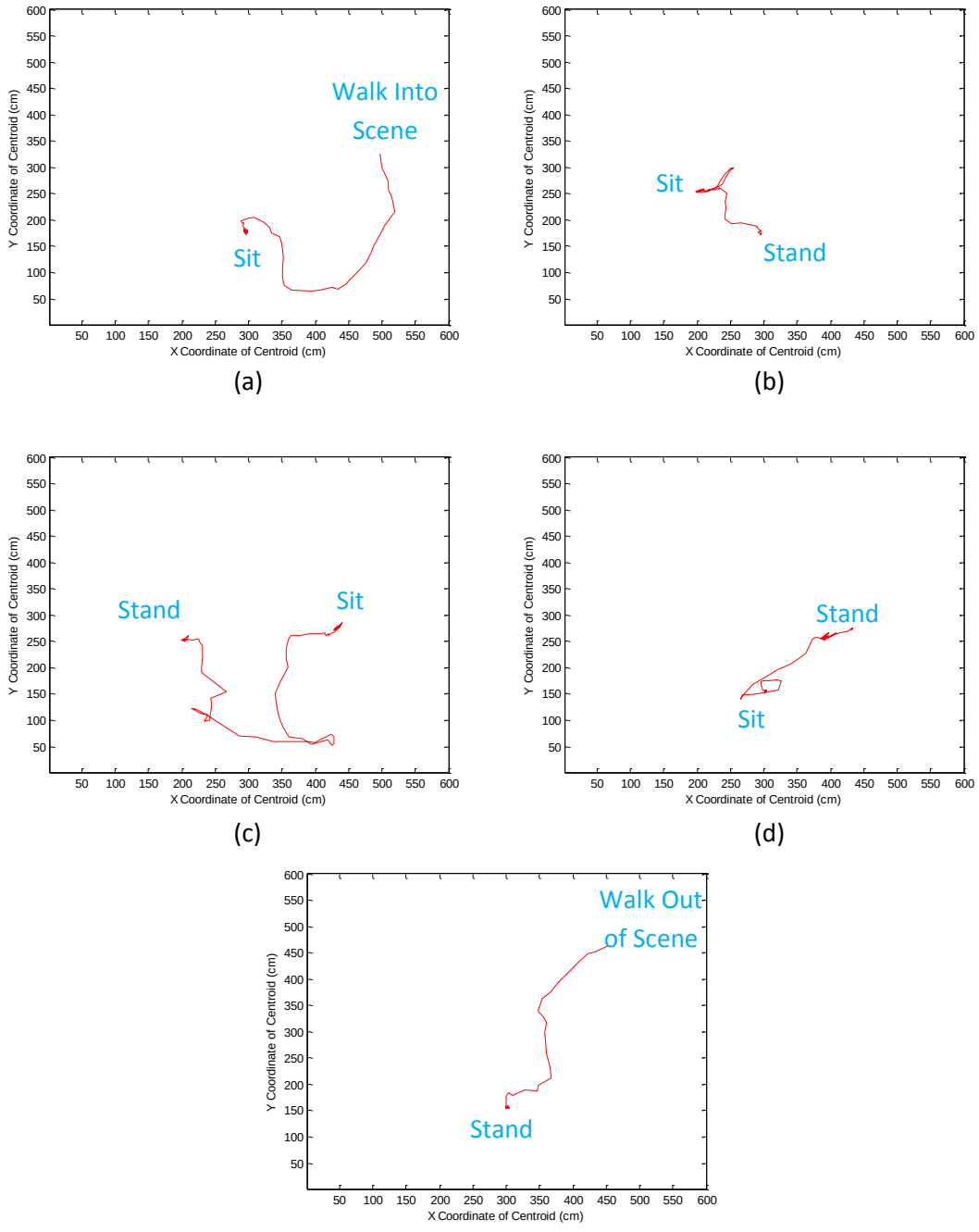


Figure 6.22: A graphical representation of the centroid of the subject throughout the sequence. The data set is broken into multiple sections. The human's movement through the current section is shown in red. (a) The subject enters the scene and sits on the chair. (b) The subject then stands, walks to the television and turns it on, and sits in

a different chair. (c) The subject stands and moves a chair, walks to a light and turns it on, then sits on the couch. (d) The subject then stands, walks to the chair that he moved; moves it back to its original position and sits. (e) The subject stands, walks to the television and turns it off, then exits the scene.

6.2.2. Experiments Using the Bootstrapped Data Sequence

This system has the unique ability of beginning operation with a human in the scene. This property is essential for a change detection algorithm or system to work outside of a controlled academic environment. To demonstrate this ability, the sequence used in the previous section is again used, this time starting at frame 356 while the subject is sitting in a chair. The system detects the human as soon as he stands in frame 430 and continues tracking him for the remainder of the sequence. Therefore, the output is nearly identical to that described in the previous section and shows that the system is just as accurate when bootstrapped, as with someone in the scene. Few algorithms can perform adequately with bootstrapped data, making this exceptionally robust. Because the output is nearly identical to that in the previous section and is therefore redundant, the output has been placed in appendix sections 8.1 and 8.2.

6.3. Processing Time

It is notable to discuss the processing time required to use this system. Stereo image capture and correspondence is performed very quickly using low-level programming provided by Point Grey. These operations take less than 50 milliseconds. The creation of the two three-dimensional representations and their intersections takes around 50 milliseconds. The remaining parts of the system including change detection, human detection and tracking, and background update require around 270 milliseconds. With these processing requirements, the system can process nearly three frames a second on a machine with an Intel Core 2 Quad CPU at 2.67 GHz with 4 GB of RAM. It should also be noted that only one of the four cores was used at runtime.

7. CONCLUSION

7.1. Summary

This dissertation has described a system with the ability to detect and track a human through a scene given changes in lighting or movement of nonhuman objects. To achieve these capabilities, this process uses stereo vision and voxel modeling to represent complex spatial and color information. Depth information is used to extract spatial information which is robust to changes in lighting conditions. Color information is critical to the technique's ability to detect and track a person.

Comparisons were made to other change detection systems consisting of either image space or world space models. A significant increase in accuracy was shown over image space models which used only screen space information. As well, the routine's three-dimensional accuracy was presented using hand segmented ground truth data as well the accuracy and reliability of features extracted from voxel representations.

All data sets, ground truth and output from the system defined in this dissertation can be found at http://cirl.missouri.edu/vision/. Other researchers may use the data as long as a reference is made to the website.

7.2. Future Work

The capabilities of this system are significant, but several additions could be made to increase the data output. The first of these would be the additional output of nonhuman moved objects. Because this information is already segmented at runtime, it would be trivial to add as output. They were omitted from the output because they were not required by the higher level human activity analysis algorithm.

Another extension to this work would be the ability to track multiple people in the scene at a time. This dissertation has shown the ability of this system to accurately detect and track a single person in the environment. It is my belief that with minimal modification, multiple people could be tracked at one time. Currently, if head volumes intersect with multiple changed segments, $\{O_1, \ldots, O_l, \ldots, O_L\}$, all the segments are considered a single person. If instead a component labeling was performed over these segments, the multiple humans would be detected.

A color descriptor would then be built for each human detected. From this point on, it becomes a classical tracking situation, including all tracking difficulties with the most obvious being when two or more human segments touch. Again, a three-dimensional representation provides much richer information about the scene that could be used to detect these situations. As an example, when two people touch, their combined segment should be the close to sum of their previous individual segments. As well, the new color descriptor will be the addition of their individual color descriptors.

Finally, it is often convenient to have the background objects segmented and classified. In the case of human activity analysis, object labels can increase the accuracy of human state classification. For example, if the centroid and height of the human is in a medium range, the person may be sitting down or they may be bending over to tie their shoe. If the additional information is given that the person's centroid is on top of the couch, it is more likely that the person is sitting on the couch, instead of tying a shoe on the couch. Also, the linguistic output of an activity analysis system could output a richer vocabulary such as, "The person is sitting on the couch."

Similar to nonhuman object labeling, human recognition could be applied. Recognizing a specific person could aid both tracking and provide a richer linguistic output. Of course, the recognition algorithm would need to take into account the special scenario of this system, most importantly that the cameras are near the ceiling aiming downward. Therefore, it is unlikely that the person's entire face would ever be visible.

8. APPENDIX

The experiments performed on the bootstrapped sequence show that the method described in this dissertation can begin working with a person in the scene. Because the output is nearly identical to the output in section 6.2.1, it has been placed in this appendix to not disturb the flow of the dissertation. This information has been added to this appendix so the reader can corroborate the system output for both data sets.

8.1. Volumetric Experiment

The seven frames used for experiments in section 6.2.1 are again used here. The confusion matrix is displayed for each frame, as well as the true positive and false negative voxels in three-dimensional space. The results in this section are nearly identical to those in 6.2.1.1, showing that the system is equally accurate whether it is bootstrapped with someone in the scene or not.

Table 8.1. Test frame one confusion matrix statistics.

Confusion Matrix Frame 1		Ground Truth	
		Foreground	Background
Our System	Foreground	53.38%	0.02%
	Background	46.62%	99.98%

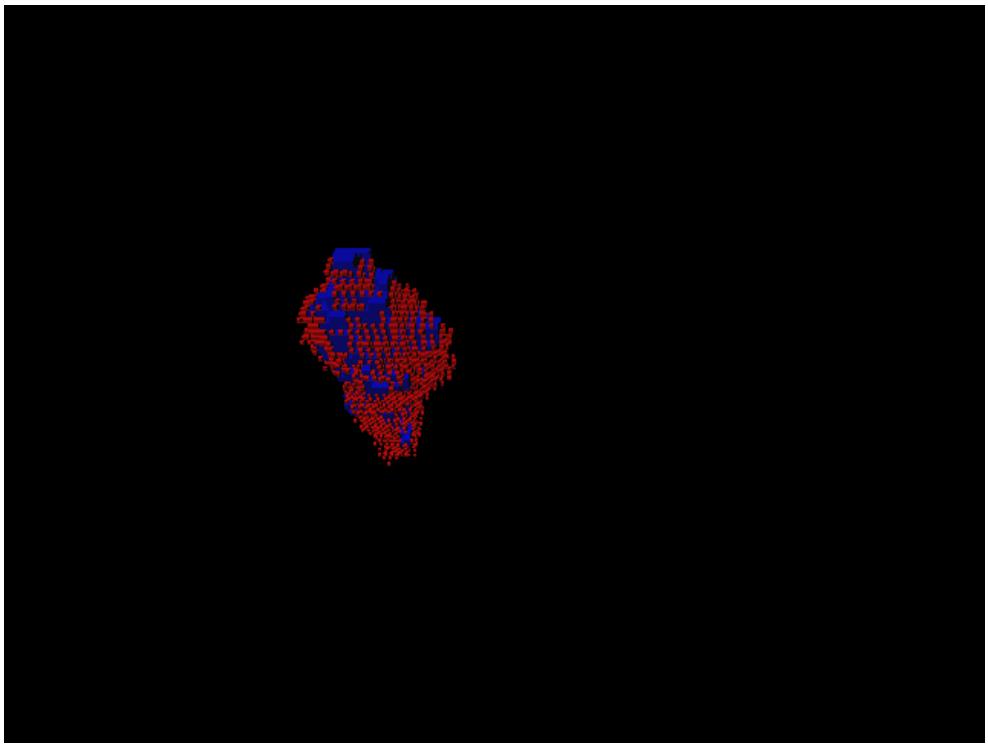


Figure 8.1: A graphical description of the error related to the output of the system for the first test frame. True positive are represented as blue voxels, while false negatives are represented as smaller red voxels.

Table 8.2. Test frame two confusion matrix statistics.

Confusion Matrix Frame 2		Ground Truth	
		Foreground	Background
Our System	Foreground	36.88%	0.04%
	Background	63.12%	99.96%

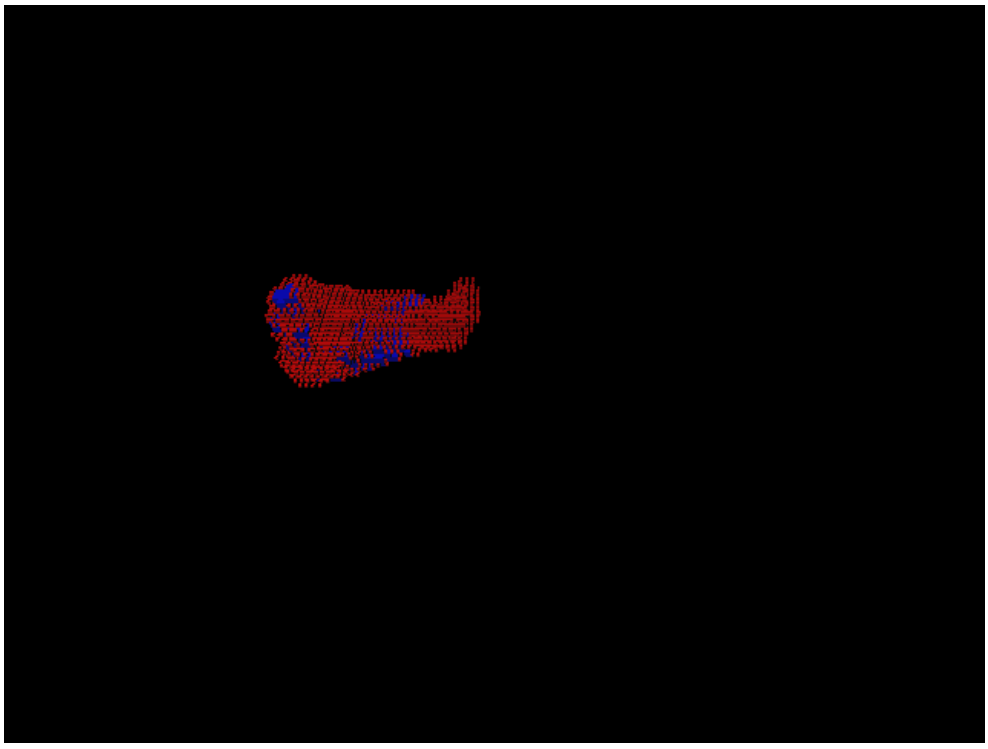


Figure 8.2: A graphical description of the error related to the output of the system for the second test frame. True positive are represented as blue voxels, while false negatives are represented as smaller red voxels.

Table 8.3. Test frame three confusion matrix statistics.

Confusion Matrix Frame 3		Ground Truth	
		Foreground	Background
Our System	Foreground	37.30%	0.0%
	Background	62.70%	100.0%

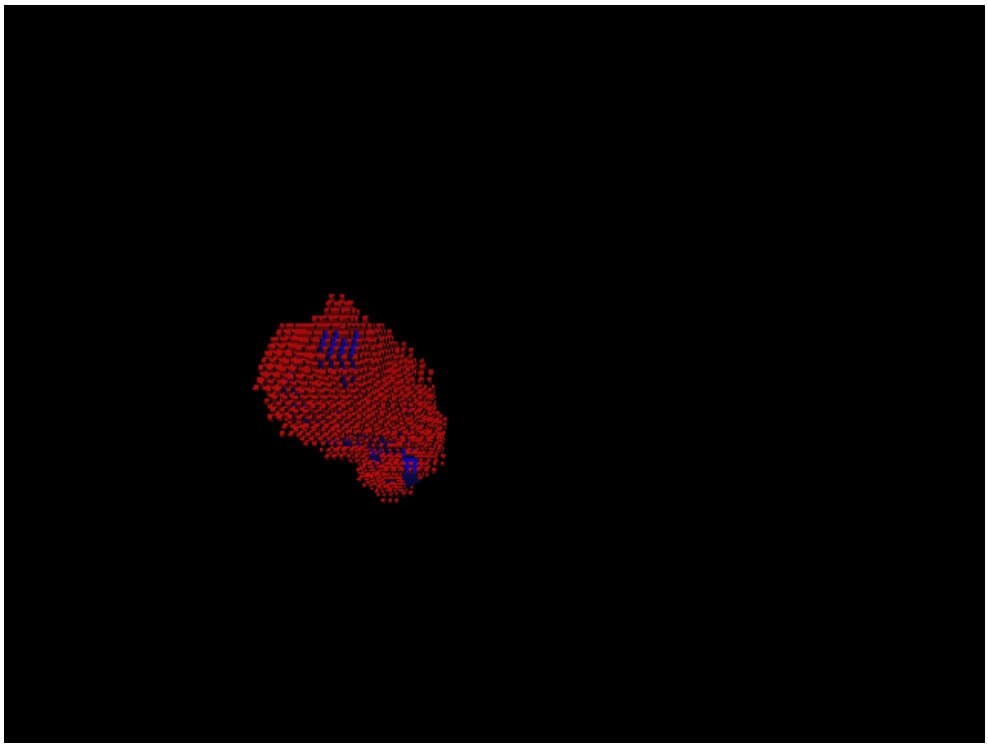


Figure 8.3: A graphical description of the error related to the output of the system for the third test frame. True positive are represented as blue voxels, while false negatives are represented as smaller red voxels.

Table 8.4. Test frame four confusion matrix statistics.

Confusion Matrix Frame 4		Ground Truth	
		Foreground	Background
Our System	Foreground	41.62%	0.03%
	Background	58.38%	99.97%

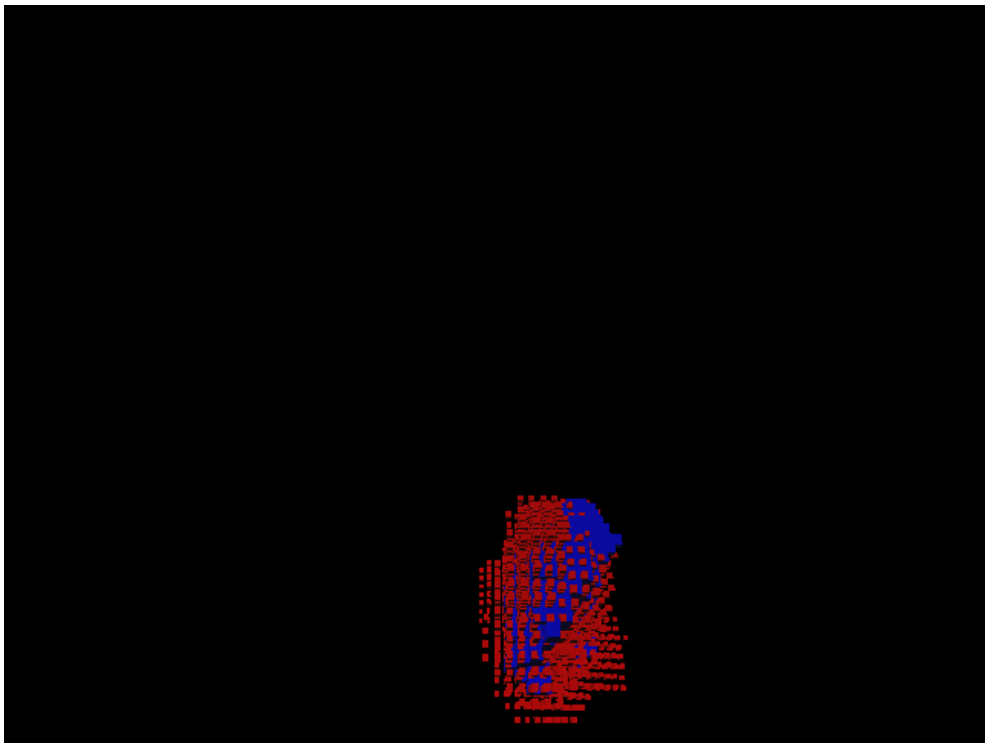


Figure 8.4: A graphical description of the error related to the output of the system for the fourth test frame. True positive are represented as blue voxels, while false negatives are represented as smaller red voxels.

Table 8.5. Test frame five confusion matrix statistics.

Confusion Matrix Frame 5		Ground Truth	
		Foreground	Background
Our System	Foreground	63.43%	0.04%
	Background	36.57%	99.96%

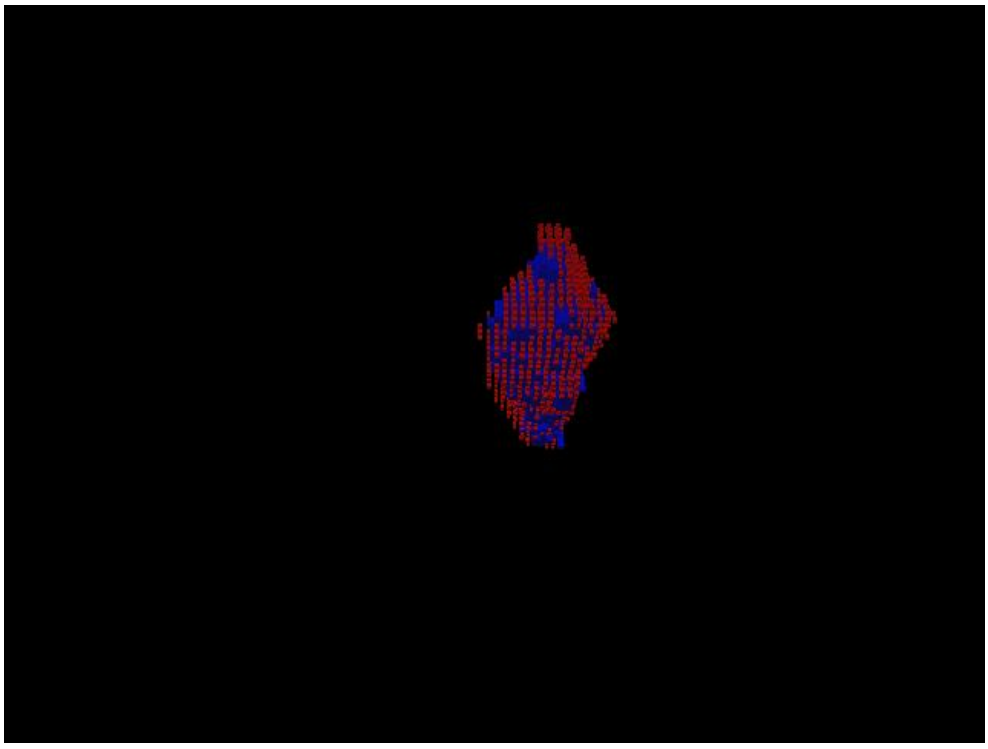


Figure 8.5: A graphical description of the error related to the output of the system for the fifth test frame. True positive are represented as blue voxels, while false negatives are represented as smaller red voxels.

Table 8.6. Test frame six confusion matrix statistics.

Confusion Matrix Frame 6		Ground Truth	
		Foreground	Background
Our System	Foreground	33.0%	0.05%
	Background	67.0%	99.95%

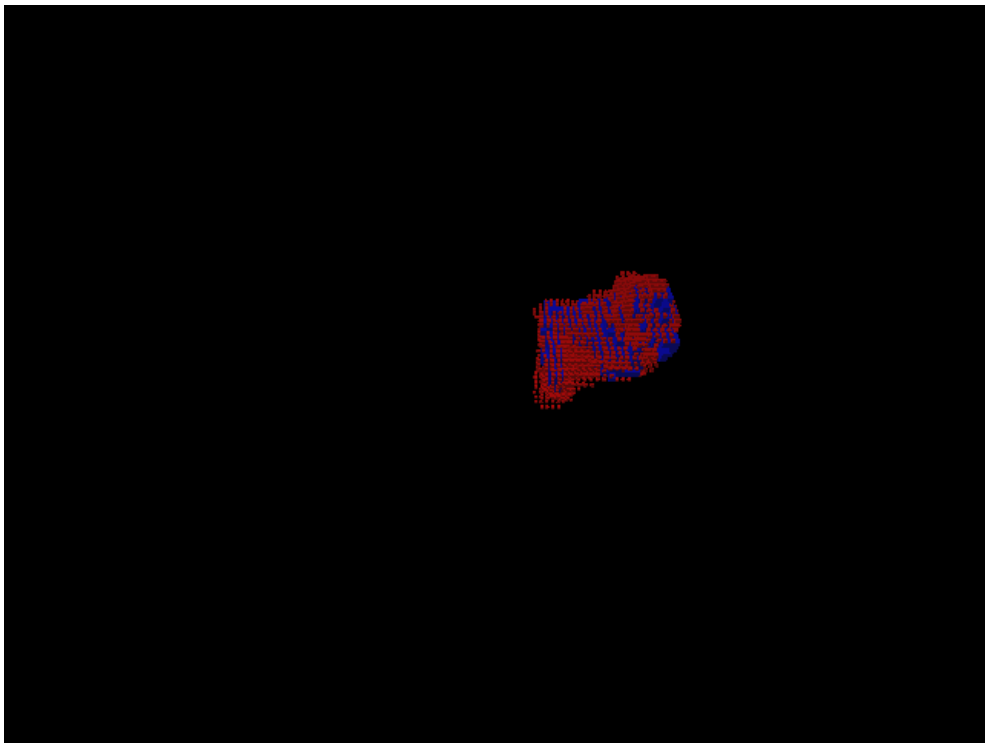


Figure 8.6: A graphical description of the error related to the output of the system for the sixth test frame. True positive are represented as blue voxels, while false negatives are represented as smaller red voxels.

Table 8.7. Test frame seven confusion matrix statistics.

Confusion Matrix Frame 7		Ground Truth	
		Foreground	Background
Our System	Foreground	67.02%	0.08%
	Background	32.98%	99.92%

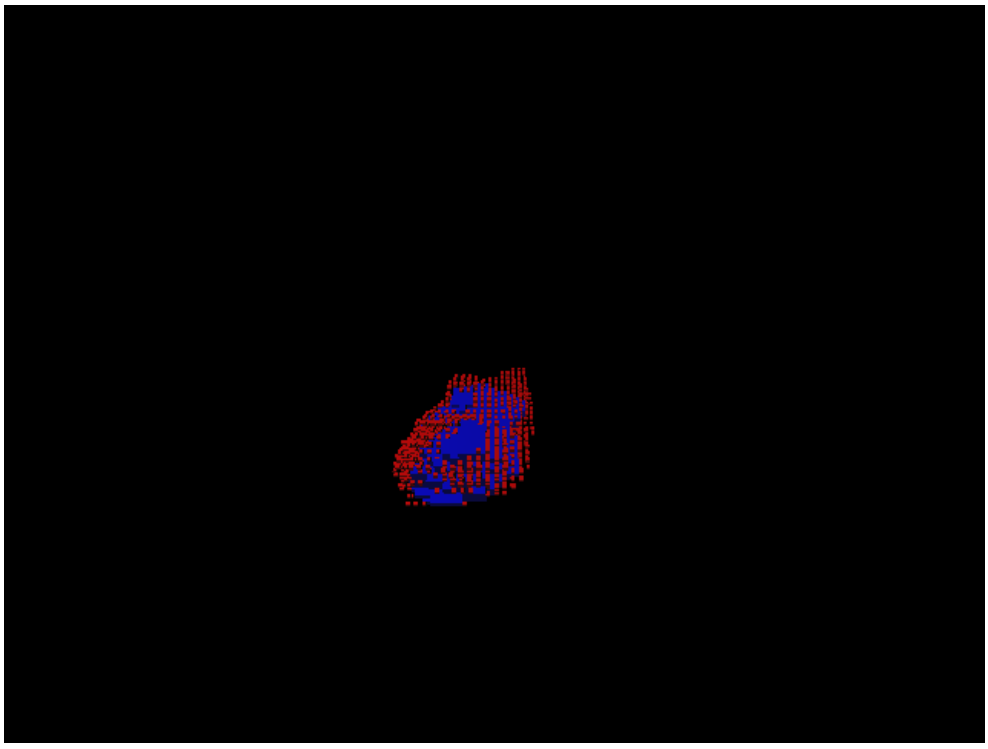


Figure 8.7: A graphical description of the error related to the output of the system for the seventh test frame. True positive are represented as blue voxels, while false negatives are represented as smaller red voxels.

Table 8.8. Confusion matrix statistics from all test data.

Confusion Matrix All Frames		Ground Truth	
		Foreground	Background
Our System	Foreground	47.61%	0.04%
	Background	52.39%	99.96%

8.2. Feature Extraction Accuracy

Like the results in 6.2.2.1, the features extracted from the bootstrap data set are nearly identical to those shown in section 6.2.1.2. The values in the following tables and graphs give further proof that the system is just as reliable being bootstrapped with a human in the scene as it is when it is begun with no one in the scene.

Table 8.9. Statistics showing the error between the height of the human output from the system and that of the ground truth, measure in voxels.

Error of Height (voxels)	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6	Frame 7	Mean	Median
Error	1.0	2.0	3.0	4.0	2.0	2.0	1.0	2.1	2.0

Table 8.10. Statistics showing the error between the height of the human output from the system and that of the ground truth, measure in centimeters.

Error of Height (cm)	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6	Frame 7	Mean	Median
Error	5.0	10.0	15.0	20.0	10.0	10.0	5.0	10.7	10.0

Table 8.11. Statistics showing the error between the centroid of the human output from the system and that of the ground truth, measure in voxels.

Error Distance From Centroid (voxels)	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6	Frame 7	Mean	Median
Error in X	0.1	0.9	0.2	1.2	0.2	3.5	1.4	1.1	0.9
Error in Y	1.0	2.5	0.4	1.4	0.6	4.1	0.7	1.5	1.0
Error in Z	1.4	2.0	1.6	1.0	1.0	0.4	1.2	1.2	1.2
Error	1.7	3.4	1.6	2.1	1.2	5.4	1.9	2.5	1.9

Table 8.12. Statistics showing the error between the centroid of the human output from the system and that of the ground truth, measure in centimeters.

Error Distance From Centroid (cm)	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6	Frame 7	Mean	Median
Error in X	0.6	4.4	1.0	6.1	0.90	17.4	6.9	5.3	4.4
Error in Y	4.9	12.9	1.9	6.8	3.1	20.5	3.5	7.7	4.9
Error in Z	7.1	10.0	7.8	4.9	5.2	2.2	5.9	6.2	5.9
Error	8.6	16.9	8.1	10.3	6.2	27.0	9.7	12.4	9.7

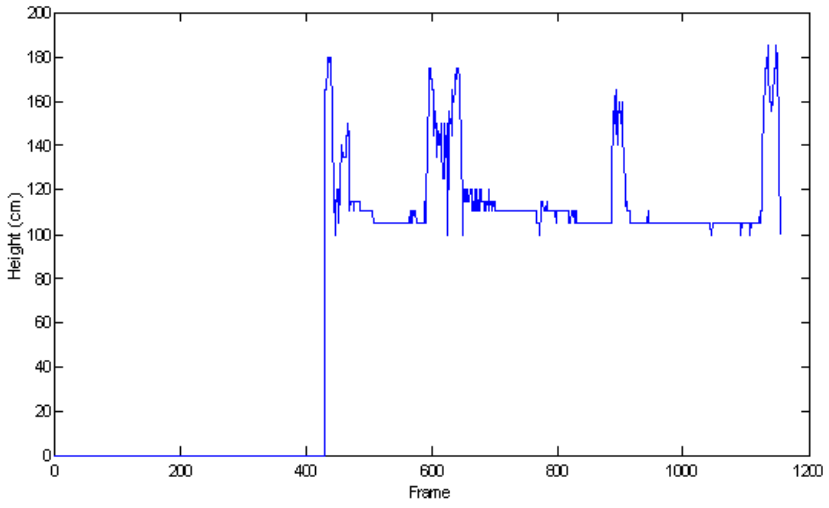


Figure 8.8: A graph displaying the height of the human output from the system. The feature is also very stable, similar to the data shown in figure 6.20.

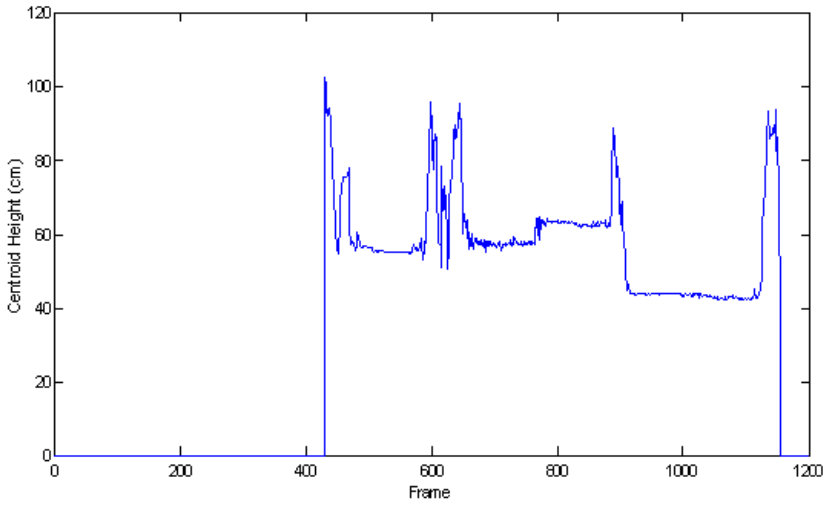


Figure 8.9: A graph displaying the height of the centroid of the human output from the system. The feature is again very stable as was displayed in figure 6.21.

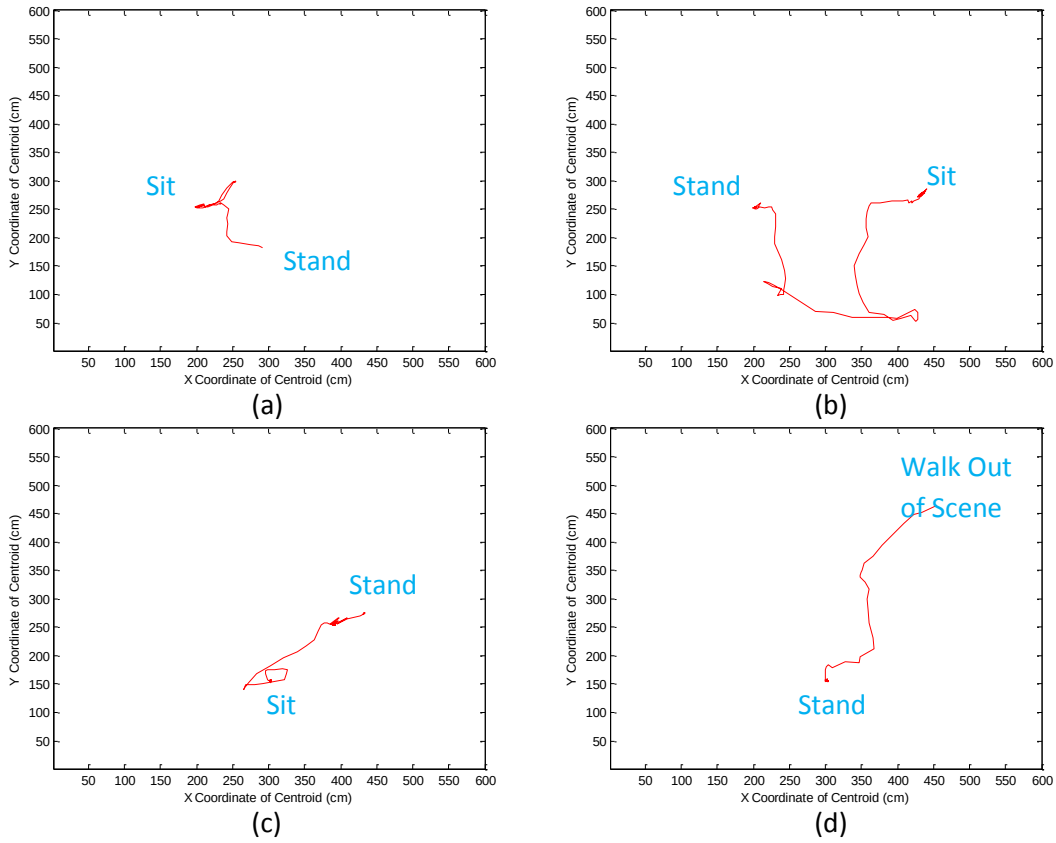


Figure 8.10: A graphical representation of the centroid of the subject throughout the sequence. The data set is broken into multiple subsets. The human's movement through the current subset is shown in red. (a) The subject stands, walks to the television and turns it on, and sits in a different chair. (b) The subject stands and moves a chair, walks to a light and turns it on, then sits on the couch. (c) The subject then stands, walks to the chair that he moved; moves it back to its original position and sits. (d) The subject stands, walks to the television and turns it off, then exits the scene.

9. Bibliography

[1] C. Stauffer and W. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," in *Computer Society Conference on Computer Vision and Pattern Recognition*, 1999.

[2] N.M. Oliver, B. Rosario, and A.P. Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 831-843, 2000.

[3] L. Wang, L. Wang, Q. Zhuo, H. Xiao, and W. Wang, *Adaptive Eigenbackground for Dynamic Background Modeling*.: Springer Berlin / Heidelberg, 2006, pp. 670-675.

[4] J. Rymel, J. Renno, D. Greenhill, J. Orwell, and G.A. Jones, "Adaptive Eigen-background for object," in *International Conference on Image Processing*, 2004, pp. 1847-1850.

[5] R. Li, Y. Chen, and X. Zhang, "Fast Robust Eigen-Background Updating for Foreground Detection," in *International Conference on Image Processing*, 2006, pp. 1833-1836.

[6] C. Eveland, K. Konolige, and R.C. Bolles, "Background Modeling for Segmentation of Video-Rate Stereo Sequences," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1998, pp. 266-271.

[7] R. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face Detection in Color Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 696-706, May 2002.

[8] Ming-Hsuan Yang, "Face Detection," in *Encyclopedia of Biometrics*.: Springer, 2009, pp. 303-308.

[9] C. Huang, H. Ai, Y. Li, and S. Lao, "High-Performance Rotation Invariant Multiview Face Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 671-686, April 2007.

[10] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and Practice of Background Maintenance," in *International Conference on Computer*

Vision, 1999, pp. 255-261.

[11] L. Li, W. Huang, I. Gu, and Q. Tian, "Foreground Object Detection from Videos Containing Complex Background," in *Proceedings of the eleventh ACM international conference on Multimedia*, Berkeley, CA, USA, 2003, pp. 2-10.

[12] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, no. 7, pp. 773-780, May 2006.

[13] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. Cambridge, Mass: MIT Press, 1964.

[14] G. Cai and J. K. Aggarwal, "Tracking Human Motion in Structured Environments Using a Distributed-Camera System," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1241-1247, Nov. 1999.

[15] T.J. Darrell, G.G. Gordon, M. Harville, and J.I. Woodfill, "Integrated Person Tracking Using Stereo, Color, and Pattern Detection," *International Journal on Computer Vision*, vol. 37, no. 2, pp. 175-185, 2000.

[16] A. Mittal and L.S. Davis, "M2Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene," *International Journal on Computer Vision*, vol. 51, no. 3, pp. 189-203, 2003.

[17] D. Beymer and Konolige K., "Real-Time Tracking of Multiple People Using Continuous Detection," in *International Conference on Computer Vision*, 1999.

[18] M.Z. Brown, D. Burschka, and G.D. Hager, "Advances in Computational Stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 993-1008, 2003.

[19] R. Muñoz-Salinas, E. Aguirre, and M. García-Silvente, "People Detection and Tracking Using Stereo Vision and Color," *Image and Vision Computing*, vol. 25, no. 6, pp. 995-1007, 2007.

[20] H. Kawanaka, H. Fujiyoshi, and Y. Iwahori, "Human Head Tracking in Three Dimensional Voxel Space," in *International Conference on Pattern Recognition*, 2006, pp. 826-829.

[21] G. H. Ball and D. J Hall, "A Clustering Technique for Summerizing Multivariate Data," *Behavioral Science*, vol. 12, pp. 153-155, 1967.

[22] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithm*. New York: Plenum Press, 1981.

[23] R. Krishnapuram and J.M. Keller, "A possibilistic approach to clustering," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 2, pp. 98-110, 1993.

[24] J. C. Dunn, "Well Separated Clusters and Optimal Fuzzy Partitions," *Journal of Cybernetics*, vol. 4, pp. 95-104, 1974.

[25] D. L. Davies and D. W. Bouldin, "Cluster Separation Measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 2, pp. 95-104, 1979.

[26] J.C. Bezdek and N.R. Pal, "Some New Indexes of Cluster Validity," *IEEE Transactions of Systems, Man, and Cybernetics - Part B*, vol. 28, no. 3, pp. 301-315, 1998.

[27] J.C. Bezdek and R.J. Hathaway, "VAT: a tool for visual assessment of (cluster) tendency," in *Proceedings of the 2002 International Joint Conference on Neural Networks*, Honolulu, HI, USA, 2002, pp. 225-2230.

[28] T.C. Havens, J.C. Bezdek, J.M. Keller, and M. and Popescu, "The relationship of VAT, CLODD, single linkage, and Dunn's validity index," *In Preparation*.

[29] T Kohonen, *Self-Organizing Maps*.: Springer-Verlag, 1995.

[30] D. O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*.: Psychology Press, 1949.

[31] W. A. Perkins, "A Model-Based Vision System for Industrial Parts," *IEEE Transactions on Computers*, vol. 27, no. 2, pp. 126-143, February 1978.

[32] E. Natonek and C. Baur, "Southwest Symposium on Image Analysis and Interpretation," , 1994.

[33] I. J. Mulligan, A. K. Mackworth, and P. D. Lawrence, "A Model-based Vision System for Manipulator Position Sensing," in *Workshop on Interpretation of 3D Scenes*, 1989, pp. 186-193.

[34] N. K. Kiriakos and M. S. Steven, "A Theory of Shape by Space Carving," *ternational Journal of Computer Vision*, vol. 38, pp. 307-314, 2000.

[35] B. Curless and M. Levoy, "A Volumetric Method for Building Complex Models from Range Images," in *International Conference on Computer Graphics and Interactive Techniques*, 1996, pp. 303-312.

[36] E. Grosso, G. Sandini, and M. Tistarelli, "3-D Object Reconstruction Using Stereo and Motion," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 6, pp. 1465-1476, 1989.

[37] G. Godin, J.-A. Beraldin, J. Taylor, L. Cournoyer, M. Rioux, S. El-Hakim, R. Baribeau, F. Blais, P. Boulanger, J. Domey, and M. Picard, "Active Optical 3D Imaging for Heritage Applications," *IEEE Computer Graphics and Applications*, vol. 22, no. 5, pp. 24- 35, 2002.

[38] K. Mühlmann, D. Maier, J. Hesser, R. Männer, K. Muhlmann, D. Maier, J. Hesser, and R. Manner, "Calculating Dense Disparity Maps from Color Stereo Images, an Efficient Implementation," *International Journal of Computer Vision*, vol. 47, pp. 79-88, 2002.

[39] R. P. Wildes, "Direct Recovery of Three-Dimensional Scene Geometry from Binocular Stereo Disparity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 761-774, August 1991.

[40] Vladimir Kolmogorov and Ramin Zabih, "Computing Visual Correspondence with Occlusions Using Graph Cuts," in *Eighth IEEE International Conference on Computer Vision*, Vancouver, BC, Canada, 2001, pp. 508-515.

[41] S. S. Intille and A. F. Bobick, "Incorporating Intensity Edges in the Recovery of Occlusion Regions," in *International Conference on Pattern Recognition*, Jerusalem, Israel, 1994, pp. 674-677.

[42] Z. Zhang and T. Kanade, "Determining the Epipolar Geometry and its Uncertainty: A Review," *International Journal of Computer Vision*, vol. 27, no. 2, pp. 161-195, 1998.

[43] Point Grey Research. [Online]. http://www.ptgrey.com

[44] J. Bouguet. (2004) Camera Calibration Toolbox for Matlab. [Online].

http://www.vision.caltech.edu/bouguetj/calib_doc/

[45] Kurt Konolige, "Small Vision Systems: Hardware and Implementation," in *Eighth International Symposium on Robotics Research*, 1997.

[46] Y. Jia, Y. Xu, W. Liu, C. Yang, Y. Zhu, X. Zhang, and L. An, "A Miniature Stereo Vision Machine for Real-Time Dense Depth Mapping," in *Lecture Notes in Computer Science*.: Springer Berlin / Heidelberg, 2003, pp. 268-277.

[47] Point Grey Stereo Vision Software. [Online]. http://www.ptgrey.com/products/triclopsSDK/triclops.pdf

[48] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.

[49] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-Squares Fitting of Two 3-D Point Sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, pp. 698-700, Sep. 1987.

[50] S. Umeyama, "Least-Squares Estimation of Transformation Parameters Between Two Point Pattems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376-380, April 1991.

[51] C. L. Jackins and S. L. Tanimoto, "Oct-Trees and Their Use in Representing Three-Dimensional Objects," *Computer Graphics, and Image Processing,* vol. 14, no. 3, pp. 249-270, November 1980.

[52] J. Serra, *Image Analysis and Mathematical Morphology*.: Academic Press, 1984.

[53] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis.," *IEEE Transaction Pattern Analysis and Machine Intelligence*, vol. 24, pp. 603-619, 2002.

[54] F. J. Estrada, A. D. Jepson, and C. Chennubhotla, "Spectral embedding and min-cut for image," in *British Machine Vision Conference*, 2004.

[55] j. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transaction Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 888-905, 2000.

[56] R. Dennie and A. Telea, "Patch-type Segmentation of Voxel Shapes using Simplified

Surface Skeletons," *Computer Graphics Forum*, vol. 27, no. 7, pp. 1837-1844, 2008.

[57] M. Donoser and H. Bischof, "3D Segmentation by Maximally Stable Volumes (MSVs)," in *ernational Conference on Pattern Recognition*, 2006, pp. 63-66.

[58] E.M van Rikxoort, Y. Arzhaeva, and B. van Ginneken, "Automatic Segmentation of the Liver in Computed Tomography Scans with Voxel Classification and Atlas Matching," *3D Segmentation In The Clinic: A Grand Challenge*, pp. 101-108, 2007.

[59] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, "A Survey on Pixel-Based Skin Color Detection Techniques," *Pattern Recognition*, vol. 40, no. 3, pp. 1106-1122, March 2007.

[60] M. M. Fleck, D. A. Forsyth, and C. Bregler, "Finding naked people," in *Proceedings of the European Conference on Computer Vision*, 1996, p. 593–602.

[61] S. J. Schmugge, S. Jayaram, M. C. Shin, and L. V. Tsap, "Objective evaluation of approaches of skin detection using ROC analysis," *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pp. 41-51, October 2007.

VITA

Robert H. Luke III was born on March 17, 1980 in St. Thomas, U.S. Virgin Islands. He received his Bachelor's Degree in Computer Science with honors in 2002, his Master's Degree in Computer Science in 2005 and his Ph.D. in Computer Engineering in 2010, all from the University of Missouri. He was a research assistant to Dr. James Keller between 2002-2006 and a National Library of Medicine Pre-doctoral Fellow between 2006-2010. His research interests include Computational Intelligence, Computer Vision, and Image Processing. When not playing video games or his guitar, he can be found rooting for his favorite soccer teams over a few beers with friends and family.