8-1-2009

Recognition of human interactions using limb-level feature points

Michael David Dudley

Follow this and additional works at: http://scholarworks.rit.edu/theses

Recommended Citation
Recognition of Human Interactions Using Limb-Level Feature Points

by

Michael David Dudley

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering

Supervised by

Dr. Andreas Savakis
Department of Computer Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, New York

August 2009

Approved by:

Dr. Andreas Savakis, Professor and Department Head
Primary Advisor – RIT Department of Computer Engineering

Dr. Juan Cockburn, Associate Professor
Secondary Advisor – RIT Department of Computer Engineering

Dr. Muhammad Shaaban, Associate Professor
Secondary Advisor – RIT Department of Computer Engineering
Title: Recognition of Human Interactions Using Limb-Level Feature Points

I, Michael David Dudley, hereby grant permission to the Wallace Memorial Library to reproduce my thesis in whole or part.

__________________________  _______________________
Michael David Dudley        Date
Acknowledgements

I would like to thank my advisor, Dr. Andreas Savakis, for his continued support and guidance throughout the entire thesis process, and my committee members Dr. Juan Cockburn and Dr. Muhammad Shaaban for their analysis and critique.

I must also thank Sooraj, Chris, Greg, Bennet, Matt, Phil, William, Matt, Carl, Mike, Natasha, Joe, Ali, and Jess for helping me create the video dataset. You were all great actors!
Abstract

Human activity recognition is an emerging area of research in computer vision with applications in video surveillance, human-computer interaction, robotics, and video annotation. Despite a number of recent advances, there are still many opportunities for new developments, especially in the area of person-person and person-object interaction. Many proposed algorithms focus on recognizing solely single person, person-person or person-object activities. An algorithm which can recognize all three types would be a significant step toward the real-world application of this technology.

This thesis investigates the design and implementation of such an algorithm. It utilizes background subtraction to extract the subjects in the scene, and pixel clustering to segment their image into body parts. A location-based feature identification algorithm extracts feature points from these segments and feeds them to a classifier which identifies videos as activities. Together these techniques comprise an algorithm that can recognize single person, person-person and person-object interactions. This algorithm’s performance was evaluated based on interactions in a new video dataset, demonstrating the effectiveness of using limb-level feature points as a method of identifying human interactions.
Table of Contents

Recognition of Human Interactions ................................................................. 1
Thesis Release Permission Form ................................................................. 2
Acknowledgements ...................................................................................... 3
Abstract .......................................................................................................... 4
Table of Contents .......................................................................................... 5
List of Illustrations ......................................................................................... 6
List of Tables ................................................................................................. 8
Chapter 1: Introduction .................................................................................. 9
  2.1 Supporting work ..................................................................................... 13
Chapter 2: Background ................................................................................. 11
  2.1 Supporting work ..................................................................................... 13
Chapter 3: Methodology ................................................................................. 17
  3.1 Initialization ......................................................................................... 22
  3.2 Background Subtraction ....................................................................... 25
  3.3 Locating Subjects ................................................................................. 26
  3.4 Pixel clustering .................................................................................... 29
  3.5 Feature point extraction ........................................................................ 34
  3.6 Activity classification .......................................................................... 41
  3.7 Video dataset ....................................................................................... 42
  3.8 Implementation .................................................................................... 46
Chapter 4: Results and Analysis ................................................................. 47
  4.1 Overall Performance ........................................................................... 48
  4.2 Background Subtraction ....................................................................... 52
  4.3 Pixel clustering .................................................................................... 55
  4.4 Feature point extraction ........................................................................ 61
  4.5 Comparison to Other Work ................................................................. 68
  4.6 Real-Time Considerations .................................................................... 70
Chapter 5: Conclusions ................................................................................ 72
  5.1 Future Work ....................................................................................... 73
Bibliography .................................................................................................... 74
List of Illustrations

Figure 1: The algorithm’s processing stages ................................................................. 17
Figure 2: Single person body model with distance features labeled ......................... 19
Figure 3: Single person body model with angle features labeled ............................... 20
Figure 4: Person-object activity angle and distance features ...................................... 21
Figure 5: Person-person activity distance feature ....................................................... 22
Figure 6: Background frame and frame 36 from subject 8’s Standing video ............... 23
Figure 7: Example 27 point Sukharev grid over HSV colorspace, from 3 views ........... 24
Figure 8: Sample frame after background subtraction (Subject 8, Standing, frame 36) .. 25
Figure 9: Example frame for column density histogram (Subject 15, Approaching, frame 49) ............................................................................................................................. 26
Figure 10: Column density histogram for example frame shown in Figure 9 .......... 27
Figure 11: Bounding box, left side, midline and right side (Subject 8, Standing, frame 36) ............................................................................................................................. 28
Figure 12: Example frame after pixel clustering (Subject 8, Standing, frame 36) .......... 29
Figure 13: Example frame after merging pixel clusters into blobs (Subject 8, Standing, frame 36) ............................................................................................................................. 30
Figure 14: Example frame after skin labeling (Subject 8, Standing, frame 36) ............ 31
Figure 15: Example frame after limb labeling (Subject 8, Standing, frame 36) ............ 32
Figure 16: Example frame after feature extraction (Subject 8, Standing, frame 36) .... 33
Figure 17: Hierarchical body model for assigning blobs to limbs ............................... 34
Figure 18: Frame 36 of Subject 8’s Standing video, with his hands in his pockets ....... 35
Figure 19: Frame 63 of Subject 10’s Jumping video, with his arms raised above his head ............................................................................................................................. 36
Figure 20: The static background scene ...................................................................... 46
Figure 21: Example of noise around foreground edges (Subject 4, Jumping, frame 50) . 52
Figure 22: Example of disconnected feet (Subject 3, Jumping, frame 30) ................. 53
Figure 23: Example of very poor background subtraction (Subject 10, Jumping, frame 63) ............................................................................................................................. 54
Figure 24: Example of shadow effects from ball on table (Subject 7, Picking Up, frame 15) ............................................................................................................................. 55
Figure 25: Example of oversegmentation in the legs (Subject 9, Waving Two, frame 40) ............................................................................................................................. 56
Figure 26: Example of oversegmentation, clustered pixels on left and merged blobs on right (Subject 14, Walking, frame 45) .............................................................................. 57
Figure 27: Example of undersegmentation in the legs (Subject 5, Picking Up, frame 24) ............................................................................................................................. 58
Figure 28: Example of good (blue) and bad (red) skin identification (Subject 8, Standing, frame 36) ............................................................................................................................. 59
Figure 29: Example of good (blue) and bad (red) skin identification (Subject 10, Pointing, frame 31) ............................................................................................................................. 60
Figure 30: Example of Subject 1’s bad skin identification (Subject 1, Standing, frame 20) ............................................................................................................................. 61
Figure 31: Example of Subject 12’s bad skin identification (Subject 12, Standing, frame 34) ............................................................................................................................. 62
Figure 32: Example of good object segmentation during occlusion (Subject 7, Picking Up, frame 15) ............................................................................................................................................. 60
Figure 33: Example of object not being found (Subject 2, Setting Down, frame 20) ...... 60
Figure 34: Example of good feature identification in frontal view (Subject 11, Standing, frame 61) ............................................................................................................................................. 61
Figure 35: Example of good feature identification in side view (Subject 11, Walking, frame 56) ............................................................................................................................................. 62
Figure 36: Example of bad chest feature point extraction (Subject 9, Running, frame 32) ............................................................................................................................................. 62
Figure 37: Example of bad chest feature point extraction (Subject 1, Waving Two, frame 26) ............................................................................................................................................. 63
Figure 38: Example of good hand feature point extraction (Subject 1, Waving Two, frames 2, 9, 14, 20) ............................................................................................................................................. 64
Figure 39: Example of bad hand feature point extraction (Subject 3, Setting Down, frame 35) ............................................................................................................................................. 65
Figure 40: Example of bad waist feature point extraction (Subject 12, Jumping, frame 25) ............................................................................................................................................. 66
Figure 41: Example of bad feet feature point extraction (Subject 8, Running, frame 34) 66
Figure 42: Feature vector length distribution for vectors identified as single person activities ............................................................................................................................................. 67
Figure 43: Feature vector length distribution for vectors identified as person-person activities ............................................................................................................................................. 67
Figure 44: Feature vector length distribution for vectors identified as person-object activities ............................................................................................................................................. 68
List of Tables

Table 1: Symbols used in Expectation Maximization equations ........................................ 30
Table 2: List of features used ................................................................................................. 38
Table 3: Activities by activity type ....................................................................................... 42
Table 4: Activity types and descriptions .............................................................................. 43
Table 5: Volunteers for the video database varied in height, ethnicity, gender, and clothing style .............................................................................................................................................. 45
Table 6: Activity abbreviations used in classification tables .............................................. 49
Table 7: Color codes used in classification tables .................................................................. 49
Table 8: Classifications for dataset videos, run over entire dataset ................................... 49
Table 9: Number of videos correctly and incorrectly classified, grouped by activity type ........................................................................................................................................ 49
Table 10: Frame classifications for each activity, all dataset videos included .................... 50
Table 11: Classifications for dataset videos, with poorly segmented videos excluded ...... 50
Table 12: Number of videos correctly and incorrectly classified, grouped by activity type, with poorly segmented videos excluded .............................................................................. 51
Table 13: Frame classifications per activity, with poorly segmented videos excluded .... 51
Chapter 1: Introduction

Automatic recognition of human activities in video is an increasingly active field of computer vision research, spurred by significant demand in the military, security and commercial sectors. Applications for this technology vary widely, ranging from automated surveillance to home video gaming systems, presenting researchers with a heterogeneous and sometimes conflicting set of requirements. As such many different algorithms have been proposed and analyzed.

Many of these algorithms focus on identifying activities performed by a single person in a scene. When multiple people are present, some algorithms examine the movement of them in the scene, but not what each individual is doing. An algorithm which can analyze several subjects in a scene, recognizing person-person and person-object interactions in addition to single person activities would be a step closer to advanced real-world applications of this technology. Current algorithms and techniques are discussed in Chapter 2.

The objective of this thesis is to investigate the design of an algorithm using limb-level features for identifying single-person activities, activities involving two people, and activities between a person and an object. The methodology behind the algorithm’s design is discussed in detail in Chapter 3. The algorithm’s performance was analyzed in depth and its potential for real-time implementation was considered in Chapter 4. Conclusions and ideas for future work are located in Chapter 5.
There were very few datasets available that were appropriate for objectively measuring this algorithm’s performance. Existing datasets do not include the desired activities or were recorded in such a way that they are unusable for this thesis. A new dataset was created for this and future research. It consisted of fourteen volunteers performing twelve activities in an indoor setting. More information on the dataset is presented in Section 3.7.
Activity recognition algorithms can be classified based on the type of activity they recognize and the level of detail at which they operate [1]. There are algorithms which can identify single-person activities [2], [3], multiple-person interactions [4], [5], [6], [7], and person-object interactions [8], [7], [9]. Some infer a person’s actions by examining their silhouette [2], [10], [11], [12], whereas others are able to distinguish between actions based on individually-identified limbs [6]. A subclass of activity recognition, gesture recognition, operates at a finer level of detail to recognize complex finger and hand motions such as those used in sign language [13], [14].

There are many challenges that make human activity recognition difficult. Depending on the environment, an algorithm may need to contend with changing lighting conditions and dynamic shadows during segmentation. Movement unrelated to the subject should not distract the recognition algorithm. Very often an algorithm must perform temporal segmentation, i.e., determine when an activity begins and ends in a continuous video stream [15], [16].

When identifying features of interest, occlusion is a major problem. Even in single-person activity recognition with no obstructing foreground elements, the hand of a person walking by a camera will appear and disappear as it passes behind the body. During close interaction, such as two people hugging, occlusion is even more of an issue.

The variation in people’s heights, weights and clothing is a potential hazard. A tall person wearing a suit looks very different to an activity recognition algorithm from a short person wearing a dress. Subtle differences in postures become very apparent in
computer analysis of human movement. The outlier cases of people using alternate modes of transportation (e.g., inline skates or scooters), children, and people with disabilities further complicate the matter. The heterogeneity of subjects is a significant obstacle to deploying activity recognition algorithms in real-world applications.

Algorithms that identify human activities typically require several layers of processing. They must segment the scene’s subjects from the background, identify features of interest, and classify the activities being performed. The processing techniques applied at each level are largely dependent on the types of activities being classified and the target environment. For example, using background subtraction for segmentation would not work well in an outdoor environment with swaying trees. However it could perform very well indoors where the background is static.

Sometimes researchers ignore real-world issues such as real-time performance and scalability when developing new algorithms. Certain algorithms can be trained to recognize a set of core activities [2], [5]. This makes the algorithms application specific, but it makes testing significantly easier. Algorithms using context-free grammars to describe activities that are more easily extended to include new activities have been explored recently [6].

An activity recognition algorithm could easily take several minutes to process a single frame of video on a fairly powerful desktop computer. Yet in real-time situations the algorithm must balance complexity, video quality, and latency to achieve sufficient performance. When the desired performance is achieved it will be useful in a broader range of applications, such as immediately notifying security in response to a situation detected on a surveillance camera.
The fact remains that automatic human activity recognition is a broad and complex area of research. The wide range of approaches researchers take on the problem is indicative of this. Any new research helps to advance the state-of-the-art by identifying techniques to pursue and techniques to avoid.

2.1 Supporting work

There has been a large amount of research performed on single-person activity recognition, but comparatively little on activity recognition in scenes with multiple interacting persons, especially at the limb level of detail [17]. Yet it is often the case that there are multiple persons in a surveillance camera’s field of view. Enabling existing algorithms to recognize the activities of multiple non-interacting persons is relatively simple, as it only requires a new tracking mechanism that can identify multiple targets simultaneously. Identifying direct (e.g., shaking hands) and indirect (e.g., approaching and departing) interactions, however, requires more infrastructure. New segmentation algorithms and more detailed activity descriptions all must be developed for a system to effectively recognize interactions.

There are several common techniques for segmenting persons from the background in a video. Background subtraction is typically used in scenes with static backgrounds due to its simplicity and low computational complexity. Motion-based techniques, such as optical flow, are quite robust but come at the cost of significant processing time [18]. A variety of other elaborate algorithms have been proposed which segment based on shape and appearance.

Several methods of performing temporal segmentation have been developed as well. One technique, applied by Weinland et al. in [19], is to use minima in the global
motion energy of a scene. This can potentially lead to over-segmentation due to small fluctuations in limb velocities, but overall it seems to work very well. In [15] Rui and Anandan use optical flow to detect temporal discontinuities as activity boundaries. A much simpler technique is to label periods of time based on the dominant action, as in [2]. This method places more emphasis on the subject’s pose rather than his movement, and assumes that a recognizable activity is being performed during a particular time frame. Though it is relatively imprecise for rapidly changing activities, it does provide a general sense of what has occurred.

Occlusion can be handled several ways. In [20] Khan and Shah classify pixels in each frame by determining which pixel cluster they belong to from a previous frame. When pixels are occluded their cluster remains in memory, so that when they reappear they are still assigned to the appropriate cluster. They also provide a method for detecting when a new person has entered the scene. Other probabilistic approaches use an appearance model to distinguish between multiple persons in a scene [21].

An increasing amount of research is being performed in using multiple cameras for activity recognition. The availability of multiple viewpoints of a scene makes it much easier to develop three dimensional models of the subject in the video. As a result, some view-invariant algorithms have been developed [3]. The downside to volumetric modeling is that it comes at the cost of increased complexity and computational requirements. Since view-dependent algorithms can provide comparable accuracy with much better real-time performance in single view applications it is reasonable to continue investigation into these methods.
Activity recognition is still an emerging field and nearly all research has been targeted at recognizing the actions of adults. The KidsRoom [22], an interactive storytelling environment, was one application of coarse activity recognition with children. Though it recognized some simple actions, such as crouching and making a “Y” with their arms, it did not address the issue of tracking adults and children in the same scene. This is a potential issue when working on identifying interaction activities.

J. K. Aggarwal at the University of Texas at Austin has performed a significant amount of research in the field of activity recognition. He and his colleagues have published papers on human segmentation, motion analysis, activity recognition [23] and activity semantics, among other areas.

In [23] Park and Aggarwal describe an algorithm for the tracking and segmentation of multiple persons in a scene. The algorithm is divided into several layers. Background subtraction is applied followed by pixel-color classification. Blobs are then formed, tracked, and eventually assigned to body parts. One key aspect of Park and Aggarwal’s classification algorithm is its use of skin detection to help identify body parts. It is a simple but effective method of increasing the algorithm’s accuracy.

Park and Aggarwal claim a 97% pixel classification accuracy, which is comparable to other algorithms, but their work is not designed for real-time applications. The pixel classification uses an iterative approach and takes a considerable amount of time, even on a fast machine. However, it is able to process frames sequentially without any user input, a requirement for continuous surveillance applications.

This algorithm does not include an activity classification step. Ryoo and Aggarwal investigated semantic classification using both context-free grammars [6] and
event hierarchies [7]. Semantic descriptions are much closer to natural language descriptions of activities and can provide more information to the user about what occurred in a scene. Grammars provide an efficient, scalable method of creating these descriptions through the application of a set of recursive rules. The development of optimal grammars for activity recognition is an ongoing research effort.

Boeheim and Savakis developed a real-time single-person activity recognition algorithm using feature point detection [2]. They used background subtraction to obtain a thresholded silhouette of the person and skeletonized it. Using masks they identified feature points, such as the head, hands, and feet, and constructed a six segment model of the person. Finally they fed the locations of the feature points to an artificial neural network for classification.

A logical extension to Boeheim’s algorithm would be to enable multiple-person and person-object activity recognition, such that the system could identify activities from any of the three activity types. The body model used in Boeheim’s work is easily extendible to person-person and person-object activities. Park and Aggarwal [23] achieved good segmentation performance with their background subtraction and pixel clustering algorithms, so incorporating these into Boeheim’s algorithm could yield improved classification accuracy. These techniques together are also potentially viable for real-time implementation. A detailed discussion of the design of the investigated algorithm follows in Chapter 3.
Chapter 3: Methodology

The first steps in designing any computer algorithm are to identify the algorithm’s primary task and to use functional decomposition to break it down into smaller tasks. An algorithm for identifying the activities of humans in a scene must, at the highest level, locate features of interest and use that information to classify the activities. Locating the features of interest is the primary task the algorithm must perform, a task which can be broken down into identifying the foreground, distinguishing between individuals in the scene, segmenting them into body parts, and analyzing the segmented parts for the feature coordinates.

These basic steps invite a pipeline approach for processing the video frames. A video frame can be processed and passed on to the next stage independently of other frames. Each stage can be broken down into smaller steps, as depicted in Figure 1. This design enables a programmer to make modifications to the algorithm without affecting other portions of it. It also lends itself to multi-core processing, where frames of video
can be passed between cores to maximize parallelism and overall throughput. The following sections describe the algorithm in detail.

It is critical to consider the target environment when choosing techniques that accomplish these tasks. This research was targeted at indoor environments, e.g. a room with constant lighting, a static background and relatively little clutter. An assumption of a fixed camera with frontal and side views of the subjects also greatly simplified the algorithm’s design.

The representation of the subjects in a scene not only affects algorithm design but is also a critical factor in determining the algorithm’s classification accuracy. A good model will capture all of the relevant information in a scene; here that consists of the subjects’ poses and movements. Simple five or six segment body models have been shown to work well for representing most human activities [12]. A six-segment body model was used, as in Boeheim’s work. It encodes the relative positions and movement of the limbs and torso, the critical information for the activities being identified.

The feature vector derived from this model included the location of the head, hands and waist relative to the chest, the location of the feet relative to the waist, the movement of the chest, the distance between the chest and the object (for person-object activities), and the distance between subjects (for person-person activities). These parameters are depicted in Figure 2 through Figure 5.
Figure 2: Single person body model with distance features labeled
Figure 3: Single person body model with angle features labeled
Figure 4: Person-object activity angle and distance features
Prior to the main loop, the algorithm must initialize several models and starting data. This process is described in the next section.

3.1 Initialization

When the algorithm first begins it runs through several initialization routines. It precomputes three models to be used while the main processing loop is running: a background model for background subtraction, a skin model for skin identification and an object model for object identification.
The background model is generated by measuring the color of each pixel in an empty scene, in the same manner as Ryoo and Aggarwal [7]. An example of an empty scene is shown in Figure 6. A number of frames are examined and the mean and variance of each pixel is calculated. Twenty frames generally provide enough information for calculating these values. The Hue-Saturation-Value (HSV) was used to represent the background because it maps naturally to human perception of color and performs better for segmentation in later steps.

The skin and object models are simply parameters for classifying skin and objects by color. The skin model was determined by manually labeling skin within the dataset and determining the average HSV values. The object model parameters were obtained in the same manner. These models are extremely simplistic but simple to implement and surprisingly effective.

A number of other values used in the algorithm were determined empirically, such as the background subtraction threshold and the foreground morphological filter windows. These values are highly dependent on the particular video being processed, so
their values need to be modified whenever the algorithm is used on new video or in a new environment.

The pixel clustering’s starting vectors must also be initialized before the algorithm begins processing frames of video. Park and Aggarwal [23] used random initial vectors for the Expectation Maximization algorithm, but this makes consistent performance for testing difficult due to the fact that algorithm results are not repeatable. A Sukharev grid was used to determine the initial color vectors [24], depicted in Figure 7. That is, a cubic grid of points evenly distributed throughout the HSV color space were used as the vectors initially fed into the pixel clustering algorithm. This reduces the possibility that a particular color is missed when the colors vectors converge.

![Figure 7: Example 27 point Sukharev grid over HSV colorspace, from 3 views](image)

When the number of vectors being initialized is not a perfect cube, the points are initialized along the Hue axis first, as this is the most important color channel for segmenting the human subjects. The second axis was Saturation, and then Value. The intent is to spread out the initialization vectors throughout the color space. This increases the probability that the vectors will converge on all the colors available in the image without missing some.

The number of vectors for the pixel clustering Expectation Maximization (EM) algorithm was chosen to over-segment the subject. This enables the algorithm to determine better pixel clusters which are merged later. A value of 14 was chosen for
segmenting a single subject, assuming two colors for the hair, skin, shirt, pants, shoes and object regions.

### 3.2 Background Subtraction

Figure 8: Sample frame after background subtraction (Subject 8, Standing, frame 36)

After initialization, the algorithm enters the main loop. The first step after obtaining a new frame from the input device is to extract the foreground, as shown in Figure 8. This is accomplished by determining the Mahalanobis distance between the pixels in the current frame and the background model [23].

\[
d = \sqrt{(x - \mu)^T S^{-1} (x - \mu)}
\] (1)

The distance \(d\) is computed from the input pixel vector \(x\), the pixel mean \(\mu\), and the pixel covariance matrix \(S\). If the distance is above a predetermined threshold the pixel is labeled as a foreground pixel, resulting in a binary mask. The mask is processed using morphological filtering to remove noise and holes in the mask. A disk structuring element was used. The mask is eroded with a disk of radius 2, then dilated with a disk of radius 4, and finally eroded again with a disk of radius 2. This sequence of operations tended to remove the noise without removing detail around the subject’s silhouette.
The next step after obtaining the foreground mask is to locate the individual subjects in the scene, since the algorithm supports more than one subject.

### 3.3 Locating Subjects

Locating the subjects in the scene is simply a matter of generating a histogram of column densities in the mask and then finding the peaks [23]. The mask pixels in each column of the image are summed, and the resulting vector is averaged. The averaging window was chosen to be the estimated width of subjects in the scene, in order to produce a histogram with a smooth derivative. Figure 9 and Figure 10 show what the histogram looks like for a given foreground.

![Figure 9: Example frame for column density histogram (Subject 15, Approaching, frame 49)](image)
The peaks are thresholded to remove any noise, and the result is used to calculate the left and right sides of the subjects’ bodies as well as their midlines. The algorithm simply looks for where the peaks begin and end, and then chooses the maximum of the peak as the midline.

Notice that the maxima are not centered within the peaks. This is beneficial for calculating the subjects’ midline. Looking at the right subject in Figure 9 we see that her foot is sticking out, skewing the centroid of her bounding box to the left. However with the column density histogram we see that the maximum is located properly above her midline.

The top and bottom of the subject were located in a similar manner to the left and right, but the histogram is not calculated. Instead, the algorithm stepped through the rows of mask pixels from the top and from the bottom until it passed 5% of the pixels in either direction. This made the algorithm more robust against outlier noise.
The divisions between the head and torso and then between the torso and legs are based strictly on the height of the silhouette. In [23] Park and Aggarwal determined that a human’s head is 16% of their height, and that their legs are 55% percent, providing simple ways for estimating the neck and waist location. These bounds do move when the subject puts their hands above their head, but the different is only a few pixels, so it does not present a problem for the algorithm.

![Figure 11: Bounding box, left side, midline and right side (Subject 8, Standing, frame 36)](image)

The end result of the bounds calculation is shown in Figure 11. The far left and far right bounds extend past the body to ensure that no part of the body is cut off during segmentation. This is particularly important when the arms are extended outwards and they are not clear in the column density histograms. In these cases the extra width ensures that the arms are not inadvertently excluded.
At this point the algorithm processes each subject individually. The pixel clustering portion of the algorithm groups foreground pixels into blobs based on color, with the goal of segmenting the subject’s image into individual body parts, such as the head or hands. Figure 12 shows a frame of video after pixel clustering.

In order to cluster pixels by color the algorithm must have a set of colors where it will assign the pixels. Given that the primary colors in the foreground image are that of the skin and clothing of the subjects, it is not possible to know beforehand what these colors will be. The colors must be determined through an assumption. If we assume that each color in the image has a probability distribution, we can determine the likelihood that a pixel’s color belongs to ones of those distributions.

Each color is treated as a mixture of Gaussians. The vectors calculated during initialization are used in an Expectation Maximization algorithm, which maximizes the probability that the color set represents all the colors in the image. This probability is calculated with the following equations, using the notation in [23]:

![Figure 12: Example frame after pixel clustering (Subject 8, Standing, frame 36)
\[ p(v) = \sum_{r=1}^{C_0} p(v|\omega_r)P(\omega_r) \]  

\[ p(v|\omega_r) = (2\pi)^{-d/2}|\Sigma_r|^{-1/2} \exp \left[ -\frac{(v - \mu_r)^T \Sigma_r^{-1} (v - \mu_r)}{2} \right], \quad r = 1, \ldots, C_0 \]

In Equation 3, \( p(v|\omega_r) \) is the probability that pixel \( v \) belongs to color class \( \omega_r \), calculated from a simple \( d \)-dimensional Gaussian distribution. In Equation 2, \( p(v) \) represents the overall color distribution of pixel \( v \); it describes how likely the pixel \( v \) belongs to any one of the color classes.

Expectation Maximization alternates between an Expectation step and a Maximization step. Essentially the Expectation step calculates the probabilities of each color class given the pixels in the image, and the Maximization step updates the colors to maximize these probabilities. Eventually the color classes converge to values that maximize the Prior probabilities.

### Table 1: Symbols used in Expectation Maximization equations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v )</td>
<td>Vector (representing 3 channel pixel)</td>
</tr>
<tr>
<td>( C_0 )</td>
<td>Number of color classes</td>
</tr>
<tr>
<td>( P(\omega_r) )</td>
<td>Prior probability of the ( r )th color class ( \omega_r )</td>
</tr>
<tr>
<td>( \Sigma_r )</td>
<td>Covariance matrix of the ( r )th color class ( \omega_r )</td>
</tr>
<tr>
<td>( \mu_r )</td>
<td>Mean of the ( r )th color class ( \omega_r )</td>
</tr>
<tr>
<td>( p(v) )</td>
<td>Probability of pixel ( v ) being the color it is given a set of color distributions</td>
</tr>
<tr>
<td>( p(v</td>
<td>\omega_r) )</td>
</tr>
</tbody>
</table>

The Expectation step consist of calculating \( P(\omega|v_k, \hat{\theta}) \). \( P(\omega|v_k, \hat{\theta}) \) gives us the same probabilities calculated in Equation 2, but as a normalized percentage of the total distribution:
The maximization step consists of updating parameters \( \hat{P}(\omega_i), \hat{\mu}_i \) and \( \hat{\Sigma}_i \). These are updated individually for each color classes \( i = 1..C_0 \). The new prior probability \( \hat{P}(\omega_i) \) for color class \( \omega_i \) is the average probability calculated in Equation 4 for all pixels \( v_k \). The new mean \( \hat{\mu}_i \) is the average color of all the pixels in the image weighted by the normalized probabilities \( \hat{P}(\omega_i|v_k, \hat{\theta}) \). The new covariance \( \hat{\Sigma}_i \) is calculated in the same manner as the mean.

\[
\hat{P}(\omega_i) \leftarrow \frac{1}{n} \sum_{k=1}^{n} \hat{P}(\omega_i|v_k, \hat{\theta})
\]

\[
\hat{\mu}_i \leftarrow \frac{\sum_{k=1}^{n} \hat{P}(\omega_i|v_k, \hat{\theta})v_k}{\sum_{k=1}^{n} \hat{P}(\omega_i|v_k, \hat{\theta})}
\]

\[
\hat{\Sigma}_i \leftarrow \frac{\sum_{k=1}^{n} \hat{P}(\omega_i|v_k, \hat{\theta})(v_k - \hat{\mu}_i)(v_k - \hat{\mu}_i)^T}{\sum_{k=1}^{n} \hat{P}(\omega_i|v_k, \hat{\theta})}
\]

A maximum a posteriori classifier is used to assign the pixels to each color class. The class of each pixel is the class which maximizes \( P(\omega_r|v) \):

\[
\omega_L = \text{arg max}_r \log(P(\omega_r|v)), \quad 1 \leq r \leq C
\]

During processing it is important to check for singular or nearly singular covariance matrices. When the algorithm determines that a covariance matrix’s determinant is below a threshold, the matrix is reinitialized to the identity matrix.
Once the pixels have been assigned a known color, they are grouped by their connectivity to neighboring pixels of the same color, forming blobs. Characteristics about these blobs are calculated, including blob areas and blob neighbors. This information is used to merge blobs together to form larger blobs, reducing noise and better representing the underlying object. The algorithm loops through the list of blobs and finds blobs which are smaller than a given threshold, measured as the area of the blob in pixels. The small blobs are merged into the neighboring blob with the most shared perimeter. After merging the blobs, a frame looks like Figure 13. After this merging is complete the resulting blobs are characterized once again for use in later stages of the algorithm.

Figure 13: Example frame after merging pixel clusters into blobs (Subject 8, Standing, frame 36)

The blobs’ colors are analyzed and labeled as skin or non-skin. Figure 14 shows the blobs labeled as skin from the image in Figure 13.
The area between the subject’s waist and knees is masked to reduce confusion between the legs and the object in person-object activities. This does not impact the feature extraction stage of the algorithm because it does not use any pixel information from this region.

Once the subject has been segmented into the limbs of interests, to look like Figure 15, enough information has been extracted from the frame to identify feature points.
3.5 Feature point extraction

The features used to classify the activity being performed are extracted after blobs have been formed. Figure 16 shows features marked with black dots with white outlines. The right hand feature is missing.

![Figure 16: Example frame after feature extraction (Subject 8, Standing, frame 36)](image)

These features include the location of the head, the hands, the waist, and the feet. These are all calculated in polar coordinates relative to the chest, the most stable feature point. Distances are normalized by the distance between the chest and waist. The change in location of the chest is also used. In person-person activities the change in distance between the two subjects is used. In person-object activities the location of the object relative to the chest is used.

In order to determine which blobs represent which features, a hierarchical body model is used. The body parts are organized as depicted in Figure 17. The subject is segmented into three regions: the head region, the torso region, and the legs regions. Blobs are assigned to these regions based on their vertical location in the subject’s silhouette. The head region consists of all blobs whose centroids are in the highest 16%
of pixel rows. The torso region includes all blobs between the 16% and the 45% rows from the top, while the leg region contains all the remaining blobs at the bottom of the subject.

![Hierarchical body model for assigning blobs to limbs](image)

**Figure 17: Hierarchical body model for assigning blobs to limbs**

The torso region is then segmented into chest and arm regions, depending on whether the blobs are labeled as skin or not.

At this point there is enough information to locate the chest feature. The torso blobs are morphologically closed to remove holes and the largest blob is selected as the chest. The centroid of this blob is used as the chest feature coordinates. The chest tends to be the most stable feature point because it relies only on the existence of blobs in the center of the subject’s silhouette. For this reason it was chosen as the root feature point relative to the other feature points are calculated relative to.

The next feature to be extracted is the location of the head. The head region blobs are further segmented into face or hair blobs, based on their location and whether they are skin. Blobs whose centroids are located between the left and right side of the person (calculated in the foreground extraction stage) and are skin are called face blobs. Non-
skin blobs in the same area are called hair, and skin blobs outside of that area are arm blobs. The centroid of all the face blobs is used as the face feature coordinates.

The legs are also segmenting into either hands or legs, again based on whether they are skin or not. This makes it possible to identify the hands. The most difficult part of locating the hands is determining how to classify the arm blobs as a left arm or right arm. To do this, the algorithm first determines if the arm blobs are to the left side of the body, to the right side, or over the middle of the body. The centroid of all the center arm blobs is calculated, and assigned to the arm depending on which side of the subject’s midline it is.

Next the left and right arms are examined individually. The arm blobs are morphologically opened and closed to remove noise and false skin classification. The largest blob is selected as the best arm or hand candidate, and its location is compared to the location of the chest. If it is above the chest, the top-left or -right extreme is used. If it is below, the bottom-left or -right is used. The extrema are calculated simply by examining the outer column or row of pixels and then selecting the pixel at one end or the other. For example the bottom-left extrema is found by selecting the left-most pixel in the bottom row. This relatively simple calculation locates the hand quite well in both frontal and side views of the subject.

The waist is located next. The vertical coordinate is calculated as the midpoint between the centroids of the torso and legs. The horizontal coordinate of the waist is simply the midline of the subject. This is important in the setting-down and picking-up activities, where the subject bends over and their waist is centered over their legs, but not their torso.
The feet location algorithm is relatively simple. Blobs that were previously grouped into the leg region are assigned to either the left or right groups. This is not the subject’s left or right, but the side of the person from the camera’s view that the leg is on. The blobs to the left of the legs centroid are considered to be part of the left leg, whereas those to the right are considered part of the right leg. The leg is then morphologically closed, and the bottom-left and left-bottom extrema are calculated (bottom-right and right-bottom for the right leg). The extreme that is farthest from the centroid of the legs is used as the foot location.

The object is located using center of the bounding box for the largest blob labeled as the object. This is better than using the centroid because when the subject is holding the object, his or her hand occludes a portion of the object, which distorts the centroid.

After all the feature points have been extracted in Euclidean pixel coordinates, they are converted to polar coordinates. This representation better models the relation of the features to each other. The head, waist and hands are all converted to be relative to the chest. The feet are relative to the waist. All distances are normalized by the distance between the chest and the waist, since it’s the most distance between the two most stable feature points.

A summary of the features used by the algorithm is defined in Table 2.
Table 2: List of features used

<table>
<thead>
<tr>
<th></th>
<th>Subject A</th>
<th>Subject B</th>
</tr>
</thead>
<tbody>
<tr>
<td>d A:Head</td>
<td>(\phi_A:Head)</td>
<td>(d_B:Head)</td>
</tr>
<tr>
<td>(\Delta X_A)</td>
<td>(\Delta Y_A)</td>
<td>(\Delta X_B)</td>
</tr>
<tr>
<td>(\Delta Y_A)</td>
<td>(\Delta Y_B)</td>
<td>(\Delta Y_B)</td>
</tr>
<tr>
<td>d A:LeftHand</td>
<td>(\phi_A:LeftHand)</td>
<td>(d_B:LeftHand)</td>
</tr>
<tr>
<td>(\phi_A:LeftHand)</td>
<td>(\phi_B:LeftHand)</td>
<td>(\phi_B:LeftHand)</td>
</tr>
<tr>
<td>d A:RightHand</td>
<td>(\phi_A:RightHand)</td>
<td>(d_B:RightHand)</td>
</tr>
<tr>
<td>(\phi_A:RightHand)</td>
<td>(\phi_B:RightHand)</td>
<td>(\phi_B:RightHand)</td>
</tr>
<tr>
<td>(\phi_A:Waist)</td>
<td>(\phi_B:Waist)</td>
<td>(\phi_B:Waist)</td>
</tr>
<tr>
<td>d A:LeftFoot</td>
<td>(\phi_A:LeftFoot)</td>
<td>(d_B:LeftFoot)</td>
</tr>
<tr>
<td>(\phi_A:LeftFoot)</td>
<td>(\phi_B:LeftFoot)</td>
<td>(\phi_B:LeftFoot)</td>
</tr>
<tr>
<td>d A:RightFoot</td>
<td>(\phi_A:RightFoot)</td>
<td>(d_B:RightFoot)</td>
</tr>
<tr>
<td>(\phi_A:RightFoot)</td>
<td>(\phi_B:RightFoot)</td>
<td>(\phi_B:RightFoot)</td>
</tr>
<tr>
<td>Person-object</td>
<td>Person-person</td>
<td></td>
</tr>
<tr>
<td>d A:Object</td>
<td>(\phi_A:Object)</td>
<td>(\Delta d_B:Subjects)</td>
</tr>
</tbody>
</table>

Additional normalization is performed on the data in order to equalize the weights of the various features during classification. The goal is for each feature’s expected value to fall between 0 and 100. The maximum and minimum values for each feature are measured over the entire dataset and then the features are normalized over those ranges. Lastly, the angles features were weighted by a factor of 2 and the chest movement by a factor of 1.5, as this tended to generate better classification results.

The atan2 function common in many mathematical libraries was used for some calculations. It is defined as:

\[
\text{atan2}(y, x) = \begin{cases} 
  \arctan(y/x) & x > 0 \\
  \pi + \arctan(y/x) & y \geq 0, \ x < 0 \\
  -\pi + \arctan(y/x) & y < 0, \ x < 0 \\
  \pi/2 & y > 0, \ x = 0 \\
  -\pi/2 & y < 0, \ x = 0 \\
  \text{undefined} & y = 0, \ x = 0 
\end{cases}
\]  

The unweighted, normalized equations for calculated features are as follows. \(X\) corresponds to Columns in a frame, \(Y\) corresponds to Rows in a frame.
\[ d_{\text{Waist}} = \sqrt{(Waist_x - Chest_x)^2 + (Waist_y - Chest_y)^2} \] (10)

\[ \phi_{\text{Waist}} = \arctan \left( \frac{Waist_x - Chest_x}{Waist_y - Chest_y} \right) \] (11)

\[ \Delta X_{\text{Chest}} = \frac{1}{d_{\text{Waist}}} \left( Chest_x - Chest_{x,1} \right) \] (12)

\[ \Delta Y_{\text{Chest}} = \frac{1}{d_{\text{Waist}}} \left( Chest_y - Chest_{y,1} \right) \] (13)

\[ d_{\text{Head}} = \frac{1}{d_{\text{Waist}}} \sqrt{(Chest_x - Head_x)^2 + (Chest_y - Head_y)^2} \] (14)

\[ \phi_{\text{Head}} = \arctan \left( \frac{Chest_x - Head_x}{Chest_y - Head_y} \right) \] (15)

\[ d_{\text{LeftHand}} = \frac{1}{d_{\text{Waist}}} \sqrt{(Chest_x - LeftHand_x)^2 + (Chest_y - LeftHand_y)^2} \] (16)

\[ \phi_{\text{LeftHand}} = \arctan \left( \frac{Chest_x - LeftHand_x}{Chest_y - LeftHand_y} \right) \] (17)

\[ d_{\text{RightHand}} = \frac{1}{d_{\text{Waist}}} \sqrt{(Chest_x - RightHand_x)^2 + (Chest_y - RightHand_y)^2} \] (18)

\[ \phi_{\text{RightHand}} = \arctan \left( \frac{Chest_x - RightHand_x}{Chest_y - RightHand_y} \right) \] (19)

\[ d_{\text{LeftFoot}} = \frac{1}{d_{\text{Waist}}} \sqrt{(Waist_x - LeftFoot_x)^2 + (Waist_y - LeftFoot_y)^2} \] (20)

\[ \phi_{\text{LeftFoot}} = \arctan \left( \frac{Waist_x - LeftFoot_x}{Waist_y - LeftFoot_y} \right) \] (21)

\[ d_{\text{RightFoot}} = \frac{1}{d_{\text{Waist}}} \sqrt{(Waist_x - RightFoot_x)^2 + (Waist_y - RightFoot_y)^2} \] (22)

\[ \phi_{\text{RightFoot}} = \arctan \left( \frac{Waist_x - RightFoot_x}{Waist_y - RightFoot_y} \right) \] (23)

\[ d_{\text{Object}} = \frac{1}{d_{\text{Waist}}} \sqrt{(Chest_x - Object_x)^2 + (Chest_y - Object_y)^2} \] (24)
\[ \Delta d_{\text{Object}} = d_{\text{Object}} - d_{\text{Object}_{-1}} \]

\[ \phi_{\text{Object}} = \arctan \left( \frac{\text{Chest}_X - \text{Object}_X}{\text{Chest}_Y - \text{Object}_Y} \right) \]

\[ \Delta \phi_{\text{Object}} = \phi_{\text{Object}} - \phi_{\text{Object}_{-1}} \]

\[ d_{\text{Subjects}} = \frac{1}{d_{\text{Waist}}} (\text{Chest}_A_X - \text{Chest}_B_X) \]

\[ \Delta d_{\text{Subjects}} = d_{\text{Subjects}} - d_{\text{Subjects}_{-1}} \]

Once all of the features have been extracted for a video, the change in distance between the subjects and the motion of the subjects are calculated. The chest motion is tracked frame to frame. This is important for distinguishing between the very similar Standing and Jumping activities. The distance between the subjects is based on the distance between their chest feature coordinates.

Features which disappear and reappear greatly reduce the number of features that the algorithm can use during the classification stage. A simple feature history was implemented to reduce the effects of this. If a feature is missing, the algorithm will search up to 5 previous frames for the last known feature value. Since the activities studied here are relatively slow moving, this is a good approximation of where the features are when they are momentarily lost.

Overall the feature extraction algorithm is mostly location based. Skin and object identification play a crucial role in determining the hands, face and object location. The algorithm does have some flaws, such as when a subject bends over and his head is confused with his arm, but overall the algorithm performs well in both frontal and sideways facing views of the subjects.
### 3.6 Activity classification

Once the feature vector has been extracted the algorithm uses a k-Nearest Neighbor classifier to determine the activity. The activity classifier loads all of the extracted feature data and each feature vector is labeled as a single person activity, person-person activity or person-object activity. This determination is made by examining the available features within each feature vector. A vector is labeled as a person-object vector if the object location feature is not zero. The same applies for person-person activities, when the second person feature vector is not zero. All other frames are put in the single person activity group. Separating the feature vectors in this manner increases the activity classifier’s accuracy by preventing it from attempting to match incompatible feature vectors.

Each frame was classified by a plurality vote of the nearest k=3 neighboring feature vectors in the dataset. Each frame of video for a particular activity counts as a vote for that activity, and the video is labeled as the activity with the plurality of votes.

Several criteria were used to determine the confidence of each frame vote. The first criterion was the number of features that could be compared between feature vectors. If the two feature vectors did not share at least 10 features, the vote was thrown out. A value of 10 was chosen since in most cases the algorithm is expected to extract the head, chest, waist, and feet, a total of 10 features.

The distance of the closest neighbor in the dataset must be below a certain threshold, determined through experimentation. Frames whose distances to the current frame are above a threshold are eliminated from consideration. The only impact of this in
KNN is when there are no good matching frame candidates, in which case the frame would be labeled as unclassifiable.

Another factor used to determine confidence was the distance between the nearest neighbor and the second nearest neighbor. When this difference was above a certain threshold then there was a high probability that the nearest neighbor was the correct choice, so the frame was not classified by plurality vote, but by the nearest neighbor.

### 3.7 Video dataset

The video database consists of fourteen individuals performing twelve activities. A child was also recorded performing the activities as a fifteenth subject for future work, and was not included in this thesis’ analysis.

The database consists of three types of activities: single person, person-person and person-object. In [25] Gavrila identified three sets of generic action verbs appropriate for testing human activity recognition algorithms, divided into single-person actions, interactions with objects, and interactions between two persons.

Twelve single person, person-person and person-object activities were identified as simple, representative examples of common actions, as outlined in Table 3. Starred activities indicate activities analyzed by Boeheim and Savakis. Table 4 contains detailed descriptions of the activities.

<table>
<thead>
<tr>
<th>Table 3: Activities by activity type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Single person</strong></td>
</tr>
<tr>
<td>Standing*</td>
</tr>
<tr>
<td>Jumping*</td>
</tr>
<tr>
<td>Walking*</td>
</tr>
<tr>
<td>Running*</td>
</tr>
<tr>
<td>Waving one hand*</td>
</tr>
<tr>
<td>Waving two hands*</td>
</tr>
</tbody>
</table>

42
Table 4: Activity types and descriptions

<table>
<thead>
<tr>
<th>Activity</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standing</td>
<td>Single person</td>
<td>One subject stands facing the camera.</td>
</tr>
<tr>
<td>Jumping</td>
<td>Single person</td>
<td>One subject jumps in place facing the camera.</td>
</tr>
<tr>
<td>Walking</td>
<td>Single person</td>
<td>One subject walks across the camera’s field of view.</td>
</tr>
<tr>
<td>Running</td>
<td>Single person</td>
<td>One subject runs across the camera’s field of view.</td>
</tr>
<tr>
<td>Waving One</td>
<td>Single person</td>
<td>One subject stands facing the camera and waves one hand.</td>
</tr>
<tr>
<td>Waving Two</td>
<td>Single person</td>
<td>One subject stands facing the camera and waves both hands.</td>
</tr>
<tr>
<td>Pointing</td>
<td>Person-person</td>
<td>Two subjects face each other across the camera’s field of view, and one subject points at another.</td>
</tr>
<tr>
<td>Shaking Hands</td>
<td>Person-person</td>
<td>Two subjects stand facing each other across the camera’s field of view and shake hands.</td>
</tr>
<tr>
<td>Approaching</td>
<td>Person-person</td>
<td>Two subjects walk toward each other across the camera’s field of view.</td>
</tr>
<tr>
<td>Departing</td>
<td>Person-person</td>
<td>Two subjects walk away from each other across the camera’s field of view.</td>
</tr>
<tr>
<td>Picking Up</td>
<td>Person-object</td>
<td>One standing subject picks up a designated object from a table.</td>
</tr>
<tr>
<td>Setting Down</td>
<td>Person-object</td>
<td>One standing subject sets a designated object down on a table.</td>
</tr>
</tbody>
</table>

The subjects were all given identical instructions on the activities to perform. In general most subjects performed the activities in roughly the same manner, with varying levels of enthusiasm, but there were some marked variations. Subject 8 performed the Standing activity with his hands in his pockets, as shown in Figure 18, while all the others left their hands at their sides.
Most subjects also performed the Jumping activity with their arms at their sides. When Subject 10 performed the Jumping activity he raised his arms above his head, as shown in Figure 19.

The end result of these variations is some noise in the extracted feature set. This is not an uncommon occurrence in real-world applications, as there is no way to guarantee that a given person will perform an activity as one expects. Generally an algorithm should be designed to be robust against noise such as this.

The subjects’ height, ethnicity, gender and clothing all varied, as they would if the subjects’ had been chosen from a crowd in a public place. Sample images of each subject
are provided in Table 5. The videos were filmed in a laboratory environment, with constant lighting conditions and a static background. Background videos were obtained immediately prior to each activity to enable later background subtraction. Each activity was trimmed to start immediately before the subject began an activity, and stop immediately after. The videos were resized to 320x240 pixels and stored in uncompressed 24-bit RGB format. The total duration of the videos was 12020 frames (401 seconds), excluding background sample frames.

Table 5: Volunteers for the video database varied in height, ethnicity, gender, and clothing style

![Volunteers for the video database varied in height, ethnicity, gender, and clothing style](image)

All video sequences were filmed in a laboratory environment with a static background. The background was plain and relatively uniform with little clutter, as can be seen in Figure 20.
Ten subjects varying in gender, skin color, and height perform the interactions in front of a stationary camera. They are instructed to perform each activity based on a simple predetermined description and possibly a demonstration. A red ball and table is used in the person-object interactions.

3.8 Implementation

The algorithm was implemented using MATLAB. MATLAB provides an extensive library of math and image processing functions, reducing the amount of programming and debugging necessary for prototyping a complex algorithm. After the algorithm was implemented, the algorithm was analyzed and problem areas were identified. These results with a detailed discussion of the strengths and weaknesses of each stage are presented in Chapter 4.
Algorithm verification was generally performed through visual inspection. The behaviors of the background subtraction, pixel clustering and feature extraction stages were easily determined by examining the resulting output. Problems such as high background noise, oversegmentation, or improperly located feature points are difficult to quantify automatically without ground truth images, which are in turn very labor intensive to generate. Visual inspection was a reliable, rapid alternative for ensuring that the algorithm was operating as intended.

The algorithm as a whole was tested by running it on the entire video dataset to extract the feature vectors, and then feeding the vectors to a classifier. The classifier used a leave-one-out scheme where all of the videos for the subject present in the currently analyzed video are excluded from the dataset. This more accurately represents the algorithm’s performance since the classifier does not have any *a priori* knowledge of the subject whose activities it is identifying.

The overall classification accuracy of the algorithm was calculated as the number of videos correctly identified out of the total number of videos analyzed. The percentage of correctly identified frames was also calculated. This value is a better indication of how well an individual activity was identified, since a higher percentage of correctly identified frames indicates a higher confidence in the identified activity.

The classification accuracies were also broken down by activity and subject. This made it much easier to identify specific videos that the algorithm was having trouble with. When the videos for three subjects were identified as being poorly classified due to
bad segmentation, classification was performed on the remaining dataset to see what its hypothetical performance would be had all the videos been segmented properly.

Due to poor initial classification performance, a great deal of time was spent investigating sources of error. Individual stages of the algorithm were analyzed in depth to determine where the algorithm was failing. Some areas were determined to be limitations of the proposed algorithm, whereas others are potentially correctable. The following sections describe the results in more detail and enumerate the problems and possible solutions with the proposed algorithm.

4.1 Overall Performance

The algorithm identified 120 videos out of 168 correctly, for an overall classification accuracy of 71%. The algorithm’s overall classification performance is summarized in Table 8 through Table 13. Table 8 and Table 10 depict the classification per video in the dataset. Correct classifications are highlighted in green, whereas incorrect classifications are highlighted in red. The incorrect classifications are marked with the activity that the videos were mistaken for. The count column marked with a pound sign (#) indicates the number of correctly identified videos for a particular activity (row) or subject (column). The percentage column marked with a percentage symbol (%) indicates the percentage of videos, out of 14 total per activity or out of 12 total per subject. These fields are highlighted according to the percentage colors in Table 7.

Table 11 and Table 13 show the classification per frame in the dataset. The fields are highlighted again according to the percentage colors in Table 7. The percentage accuracy column marked with a percent symbol (%) indicates the number of frames correctly identified out of the total classified frames. The Unclassified column indicates
the number of frames out of the total number of frames for an activity that were not classified. The Total column lists the total number of frames in the dataset for each activity.

Table 6: Activity abbreviations used in classification tables

<table>
<thead>
<tr>
<th>Activity</th>
<th>Single Person</th>
<th>Person-person</th>
<th>Person-object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abbr.</td>
<td>Activity</td>
<td>Abbr.</td>
<td>Activity</td>
</tr>
<tr>
<td>Standing</td>
<td>S</td>
<td>Pointing</td>
<td>P</td>
</tr>
<tr>
<td>Jumping</td>
<td>J</td>
<td>Shaking Hands</td>
<td>SH</td>
</tr>
<tr>
<td>Walking</td>
<td>W</td>
<td>Approaching</td>
<td>A</td>
</tr>
<tr>
<td>Running</td>
<td>R</td>
<td>Departing</td>
<td>D</td>
</tr>
<tr>
<td>Waving One</td>
<td>W1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Waving Two</td>
<td>W2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7: Color codes used in classification tables

<table>
<thead>
<tr>
<th>Color</th>
<th>Classifications</th>
<th>Classification percentages</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Meaning</td>
<td>Correct</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8: Classifications for dataset videos, run over entire dataset

<table>
<thead>
<tr>
<th>Activity</th>
<th>Subject</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>#</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standing</td>
<td>J</td>
<td>J</td>
<td>J</td>
<td>J</td>
<td>J</td>
<td>J</td>
<td>PU</td>
<td>J</td>
<td>J</td>
<td>PU</td>
<td>J</td>
<td>7</td>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jumping</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>12</td>
<td>86</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Waving One</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>10</td>
<td>71</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Waving Two</td>
<td>J</td>
<td>J</td>
<td>J</td>
<td>J</td>
<td>J</td>
<td>J</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>11</td>
<td>79</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pointing</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>SH</td>
<td>SH</td>
<td>SH</td>
<td>SH</td>
<td>9</td>
<td>64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shaking Hands</td>
<td>SH</td>
<td>SH</td>
<td>SH</td>
<td>SH</td>
<td>SH</td>
<td>SH</td>
<td>PU</td>
<td>PU</td>
<td>PU</td>
<td>PU</td>
<td>12</td>
<td>86</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Approaching</td>
<td>SH</td>
<td>SH</td>
<td>SH</td>
<td>SH</td>
<td>SH</td>
<td>SH</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>8</td>
<td>57</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Departing</td>
<td>SD</td>
<td>SD</td>
<td>SD</td>
<td>SD</td>
<td>SD</td>
<td>SD</td>
<td>SD</td>
<td>SD</td>
<td>SD</td>
<td>SD</td>
<td>10</td>
<td>71</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picking Up</td>
<td>PU</td>
<td>PU</td>
<td>PU</td>
<td>PU</td>
<td>PU</td>
<td>PU</td>
<td>PU</td>
<td>PU</td>
<td>PU</td>
<td>PU</td>
<td>4</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Setting Down</td>
<td>PU</td>
<td>PU</td>
<td>PU</td>
<td>PU</td>
<td>PU</td>
<td>PU</td>
<td>PU</td>
<td>PU</td>
<td>PU</td>
<td>PU</td>
<td>120</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 9: Number of videos correctly and incorrectly classified, grouped by activity type

<table>
<thead>
<tr>
<th>Activity type</th>
<th>Correct</th>
<th>Incorrect</th>
<th>Total</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single person</td>
<td>64</td>
<td>20</td>
<td>84</td>
<td>76%</td>
</tr>
<tr>
<td>Person-person</td>
<td>42</td>
<td>14</td>
<td>56</td>
<td>75%</td>
</tr>
<tr>
<td>Person-object</td>
<td>14</td>
<td>14</td>
<td>28</td>
<td>50%</td>
</tr>
<tr>
<td>Overall</td>
<td>120</td>
<td>48</td>
<td>168</td>
<td>71%</td>
</tr>
</tbody>
</table>
Table 10: Frame classifications for each activity, all dataset videos included

<table>
<thead>
<tr>
<th>Activity</th>
<th>Standing</th>
<th>Jumping</th>
<th>Walking</th>
<th>Running</th>
<th>Waving One</th>
<th>Waving Two</th>
<th>Pointing</th>
<th>Shaking Hands</th>
<th>Approaching</th>
<th>Departing</th>
<th>Picking Up</th>
<th>Setting Down</th>
<th>%</th>
<th>Unclassified</th>
<th>Total frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standing</td>
<td>412</td>
<td>399</td>
<td>1</td>
<td>0</td>
<td>21</td>
<td>38</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>24</td>
<td>38</td>
<td>6</td>
<td>36</td>
<td>193</td>
<td>1134</td>
</tr>
<tr>
<td>Jumping</td>
<td>139</td>
<td>726</td>
<td>40</td>
<td>14</td>
<td>28</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>10</td>
<td>1</td>
<td>64</td>
<td>133</td>
<td>1134</td>
</tr>
<tr>
<td>Walking</td>
<td>22</td>
<td>57</td>
<td>395</td>
<td>37</td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>14</td>
<td>2</td>
<td>44</td>
<td>330</td>
<td>1134</td>
</tr>
<tr>
<td>Running</td>
<td>12</td>
<td>27</td>
<td>104</td>
<td>228</td>
<td>9</td>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>6</td>
<td>133</td>
<td>59</td>
<td>228</td>
<td>647</td>
</tr>
<tr>
<td>Waving One</td>
<td>174</td>
<td>66</td>
<td>5</td>
<td>73</td>
<td>42</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>43</td>
<td>0</td>
<td>64</td>
<td>70</td>
<td>1134</td>
<td></td>
</tr>
<tr>
<td>Waving Two</td>
<td>27</td>
<td>55</td>
<td>6</td>
<td>3</td>
<td>121</td>
<td>728</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>19</td>
<td>11</td>
<td>64</td>
<td>155</td>
<td>1134</td>
<td></td>
</tr>
<tr>
<td>Pointing</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>509</td>
<td>224</td>
<td>213</td>
<td>69</td>
<td>0</td>
<td>45</td>
<td>119</td>
<td>1134</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shaking Hands</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>28</td>
<td>723</td>
<td>14</td>
<td>134</td>
<td>60</td>
<td>20</td>
<td>64</td>
<td>155</td>
<td>1134</td>
<td></td>
</tr>
<tr>
<td>Approaching</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>97</td>
<td>163</td>
<td>284</td>
<td>136</td>
<td>0</td>
<td>31</td>
<td>235</td>
<td>921</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Departing</td>
<td>3</td>
<td>6</td>
<td>2</td>
<td>18</td>
<td>1</td>
<td>54</td>
<td>62</td>
<td>85</td>
<td>574</td>
<td>15</td>
<td>11</td>
<td>60</td>
<td>121</td>
<td>952</td>
<td></td>
</tr>
<tr>
<td>Picking Up</td>
<td>15</td>
<td>23</td>
<td>5</td>
<td>6</td>
<td>23</td>
<td>19</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>3</td>
<td>325</td>
<td>162</td>
<td>44</td>
<td>154</td>
<td>742</td>
</tr>
<tr>
<td>Setting Down</td>
<td>5</td>
<td>23</td>
<td>1</td>
<td>3</td>
<td>10</td>
<td>40</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>304</td>
<td>221</td>
<td>27</td>
<td>202</td>
<td>812</td>
<td></td>
</tr>
</tbody>
</table>

Overall frame classification accuracy: 60%

In order to determine how well the overall algorithm performs in the absence of extremely poor segmentation, subjects 1, 10, and 12 were excluded from the dataset and the classification algorithm was run once more. With these subjects removed the algorithm identified 112 out of 132 videos correctly, an accuracy of 85%. The single person and person-person videos were classified with accuracies of 91% and 93%, respectively, while the person-object activities were only identified 50% of the time.

Table 11: Classifications for dataset videos, with poorly segmented videos excluded

| Subject Activity | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | #  | %   |
|------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|
| Standing         | J | J | J | J |   |   |   |   |   |    |    |    |    |    |    | 7  | 64 |
| Jumping          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |
| Walking          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    | 11 |
| Running          | W |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    | 9  |
| Waving One       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    | 11 |
| Waving Two       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    | 11 |
| Pointing         |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | A  | 10 |
| Shaking Hands    |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | A  | 10 |
| Approaching      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    | 11 |
| Departing        |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | A  | 10 |
| Picking Up       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | SD |    |
| Setting Down     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | SD | 7  |
|                  |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | SD | 36 |
|                  | # | 10| 10| 9 | 10| 10| 11| 10| 11| 10 | 10 | 11 | 112|    |    |
|                  | % | 83| 83| 75| 83| 83| 92| 83| 92| 83 | 83 | 92 | 85 |    |    |    |
Table 12: Number of videos correctly and incorrectly classified, grouped by activity type, with poorly segmented videos excluded

<table>
<thead>
<tr>
<th>Activity type</th>
<th>Correct</th>
<th>Incorrect</th>
<th>Total</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single person</td>
<td>60</td>
<td>6</td>
<td>66</td>
<td>91%</td>
</tr>
<tr>
<td>Person-person</td>
<td>41</td>
<td>3</td>
<td>44</td>
<td>93%</td>
</tr>
<tr>
<td>Person-object</td>
<td>11</td>
<td>11</td>
<td>22</td>
<td>50%</td>
</tr>
<tr>
<td>Overall</td>
<td>112</td>
<td>20</td>
<td>132</td>
<td>85%</td>
</tr>
</tbody>
</table>

Table 13: Frame classifications per activity, with poorly segmented videos excluded

<table>
<thead>
<tr>
<th>Activity</th>
<th>Standing</th>
<th>Jumping</th>
<th>Walking</th>
<th>Running</th>
<th>Waving One</th>
<th>Waving Two</th>
<th>Pointing</th>
<th>Shaking Hands</th>
<th>Approaching</th>
<th>Departing</th>
<th>Picking Up</th>
<th>Setting Down</th>
<th>%</th>
<th>Unclassified</th>
<th>Total Frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standing</td>
<td>434</td>
<td>288</td>
<td>3</td>
<td>0</td>
<td>24</td>
<td>34</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>16</td>
<td>1</td>
<td>0</td>
<td>44</td>
<td>193</td>
</tr>
<tr>
<td>Jumping</td>
<td>134</td>
<td>659</td>
<td>30</td>
<td>1</td>
<td>20</td>
<td>32</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>65</td>
<td>133</td>
</tr>
<tr>
<td>Walking</td>
<td>31</td>
<td>369</td>
<td>48</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>45</td>
<td>330</td>
<td>825</td>
<td></td>
</tr>
<tr>
<td>Running</td>
<td>10</td>
<td>14</td>
<td>73</td>
<td>209</td>
<td>9</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>28</td>
<td>1</td>
<td>0</td>
<td>36</td>
<td>228</td>
</tr>
<tr>
<td>Waving One</td>
<td>83</td>
<td>12</td>
<td>0</td>
<td>750</td>
<td>31</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>79</td>
<td>70</td>
</tr>
<tr>
<td>Waving Two</td>
<td>13</td>
<td>11</td>
<td>2</td>
<td>0</td>
<td>122</td>
<td>667</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>68</td>
<td>155</td>
</tr>
<tr>
<td>Pointing</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>14</td>
<td>139</td>
<td>21</td>
<td>0</td>
<td>66</td>
</tr>
<tr>
<td>Shaking Hands</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>6</td>
<td>42</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>72</td>
<td>155</td>
</tr>
<tr>
<td>Approaching</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>73</td>
<td>56</td>
<td>312</td>
<td>71</td>
<td>0</td>
<td>0</td>
<td>42</td>
<td>235</td>
</tr>
<tr>
<td>Departing</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>60</td>
<td>11</td>
<td>103</td>
<td>444</td>
<td>0</td>
<td>0</td>
<td>58</td>
<td>121</td>
</tr>
<tr>
<td>Picking Up</td>
<td>5</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>217</td>
<td>165</td>
<td>38</td>
<td>38</td>
<td>154</td>
</tr>
<tr>
<td>Setting Down</td>
<td>2</td>
<td>11</td>
<td>3</td>
<td>0</td>
<td>9</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>207</td>
<td>182</td>
<td>28</td>
<td>72</td>
<td>202</td>
</tr>
</tbody>
</table>

The activity classification matrix (Table 11) makes it clear that the algorithm also had trouble with the Standing, Picking Up, and Setting Down activities. These activities were all classified correctly less than 65% of the time.

For the Standing activity the primary cause behind this low figure was the fact that the only feature that is very different from Jumping is the vertical chest movement. This makes distinguishing between the two activities very susceptible to noise. The Standing activity has very slight up and down chest movement due to jitter in the feature identification, so at times it looks more like Jumping. Weighting this vertical feature only amplifies the noise, and averaging the movement is a delicate operation because it could
very easily make the movement invisible. This is a limitation of pose-based algorithms when dealing with activities which are primarily distinguished by motion.

The difference between the Picking Up and Setting down activities are also limited, with the chief difference being in the object features (angle and distance between chest and object). However there was a much larger problem which lead to the failure to identify these activities. The object classification portion of the code was unable to reliably locate and track the object, so there was a significant loss of input data.

The overall performance of the algorithm is determined by the performance of each of its parts. The following sections describe the results from each stage of the algorithm in detail, with good and bad output from each.

4.2 Background Subtraction

The background subtraction stage performed sufficiently well. As with any basic background subtraction algorithm, noise in the image presented difficulties. Although morphological filtering eliminated this problem almost entirely, the edges around the persons were not perfect. Figure 21 shows an example of the noise around the foreground image.

Figure 21: Example of noise around foreground edges (Subject 4, Jumping, frame 50)
There was also the problem of the gray baseboard and marker tray present in the dataset videos. The gray is a very neutral color and tends to blend into the foreground in HSV space. Subject 3’s feet are disconnected in the frame of video in Figure 22.

![Figure 22: Example of disconnected feet (Subject 3, Jumping, frame 30)](image)

The end result is that most subjects’ feet are not connected to their main body silhouettes. The rest of the algorithm was designed with this in mind—the body’s foreground need not be entirely connected for the algorithm to work.

From the activity classification matrix it is clear that the algorithm performed especially poorly on subjects 1, 10, and 12. Examination of the processed videos revealed that these subjects were poorly segmented. In the case of subject 10 this was due to bad background subtraction on his white shorts, as is shown in Figure 23:
Figure 23: Example of very poor background subtraction (Subject 10, Jumping, frame 63)

Unfortunately this is an inherent limitation to background subtraction. The activity recognition algorithm cannot compensate for the loss of this much of the silhouette.

Shadows also caused some issues during background subtraction. The shadow detection code performed well enough to remove them in most cases, but it was not able to remove all of them. The shadow from the ball on the table is quite clear in Figure 24. Generally the shadows were not big enough to significantly impact the algorithm’s performance.

Figure 24: Example of shadow effects from ball on table (Subject 7, Picking Up, frame 15)
4.3 Pixel clustering

In many situations the pixel clustering algorithm made grouped the foreground pixels as desired. However there were some situations in which it performed very poorly. The problems stem from either oversegmentation, as in Figure 25, or undersegmentation, as in Figure 26.

Figure 25: Example of oversegmentation in the legs (Subject 9, Waving Two, frame 40)
Ultimately these oversegmented blobs are simply grouped into larger segments representing the torso or legs. However oversegmentation makes this grouping more difficult and resource intensive.

In other situations the subjects’ images were undersegmented. At times the blobbing algorithm would connect segments. This is undesirable because it does not provide sufficient detail to detect individual body parts. In Figure 27 the subject’s legs and chest are merged into a single blob.
The skin detection algorithm worked surprisingly well given its simplicity. Using only some simple color bounds the algorithm was able to identify skin in most situations. In situations where the skin algorithm failed, it failed rather spectacularly. In some situations chest was identified as skin, causing the arms and possibly the head to be grouped together into a single large skin blob. The algorithm was designed with clothed subjects in mind, so this completely confused it. In other situations no skin was found at all. Poor skin identification is the primary cause of bad segmentation in subjects 1 and 12’s videos. Figure 28 through Figure 31 show good and bad skin segmentation.
The skin identification problem is tied closely with the pixel clustering algorithm. One might propose an algorithm in which skin identification occurs before pixel
clustering, where the skin pixels are not clustered with the non-skin pixels. This is simply shifting the skin/non-skin confusion from the pixel clustering to the skin identification algorithm. It would be possible for the skin identification algorithm to label and remove a significant portion of the subject’s silhouette, reducing the effectiveness of the pixel clustering algorithm. The clustering is essentially divided between the skin identification code and the explicit pixel clustering code. The decision was made to let the pixel clustering algorithm do its work, and let the skin identification occur after the clustering.

The object identification algorithm performed similarly to the skin algorithm, although it identified the object correctly in fewer frames. The primary problem was occlusion. When the hand covered most of the ball there were insufficient pixels visible for the pixel clustering algorithm to create a color label for the object, and so the entire object was lost. This affected a large portion of frames, and the object identification algorithm only found a ball in 480 out of 888 frames (54%) of frames.

The large number of frames missing the object had a detrimental effect on the recognition of the other activities. These frames were considered single person activity frames, creating distracting noise in the dataset and reducing the amount of training data available for the person-object activities.
Figure 32: Example of good object segmentation during occlusion (Subject 7, Picking Up, frame 15)

Source frame

Pixels clustered

Merged pixel clusters

Identified object

Figure 33: Example of object not being found (Subject 2, Setting Down, frame 20)

Figure 32 shows a good example of object identification. The subject’s hand is clearly visible, but the object is still represented by a single blob. Figure 33 shows an
example of the object not being located in a frame, as the entirely black frame indicates. In this case it was not found because the visible portions of the object had been merged into the arm. Too much of the object was occluded and the pixel clustering algorithm merged the small object blobs into the larger arm blob.

4.4 Feature point extraction

The feature point extraction algorithm was designed to work in both frontal and side views of the subject. When the subject was segmented well, the feature points were extracted well, as shown in Figure 34 and Figure 35. The details in the feature point extraction algorithm are best evaluated on a feature-by-feature basis. The various parts of the body all move in very different ways, so each feature point was found using a different method.

Figure 34: Example of good feature identification in frontal view (Subject 11, Standing, frame 61)
A chest feature point was located in nearly all frames. In general it was stable, but sometimes due to bad segmentation it would jump around, as in Figure 36. Since it is the keystone point, the noise rippled throughout the points. All of the features appear to move to the classification algorithm, rather than just the chest feature point. The frames where the chest is badly identified contribute to noise in the feature training set.

The head was also found in most images, so long as skin was identified properly. In the Picking Up and Setting Down activities the head and arm sometimes got confused, because the head was outside of the expected region. In the Waving Two activity the
arms were sometimes confused with the head, when subjects moved their arms close to their heads as in Figure 37.

![Figure 37: Example of bad chest feature point extraction (Subject 1, Waving Two, frame 26)](image)

The arms were identified well in most orientations. Figure 38 is a sequence of frames showing Subject 1 raising his arms in the Waving Two activity. Note the problem with the hand identification in the third frame, where the arms are outstretched horizontally. This is a direct result of the extrema calculations used to find the hands, but fortunately the problem only manifests itself for the few frames containing this pose.
Figure 38: Example of good hand feature point extraction (Subject 1, Waving Two, frames 2, 9, 14, 20)
Figure 39 shows bad right hand identification (the cyan dot on top of the head). The head was positioned outside of the expected head region of interest, so the skin blob was considered part of the arm.

The activities in which the hands and head were confused were chiefly the Waving One, Waving Two, Picking Up and Setting Down activities. In Waving One and Waving Two the arms above the head sometimes strayed into the head region and were confused with the head. In Picking Up and Setting Down the head moved outside of the expected head region so the skin was identified as arms. These particular failures were anticipated as potential limitations of the algorithm. Better estimation of the expected location of the head would probably reduce this confusion.

The waist was found in nearly all frames, so long as the chest was identified properly. Since its position is calculated as the midpoint between the chest and the legs it generally only failed when the silhouette was incomplete.
The feet were very reliably identified. In the Walking and Running activities the movement may not exactly follow the feet, but heir movement is still sufficiently unique to separate them from the activities where the subject stands in one place.

Some analysis on the extracted vector quality was performed. The following figures show the distribution of the number of features per feature vector, by activity type. Ideally no features would be missing at all, and the graphs would show single bars at 14 features for single person activities, 29 features for person-person interactions, and 16 features for person-object interactions. However since the feature extraction was not perfect, these graphs provide some insight into the quality of the feature vectors.
Figure 42: Feature vector length distribution for vectors identified as single person activities

Figure 43: Feature vector length distribution for vectors identified as person-person activities
The peaks for the histograms show that in general 2 to 3 features were missing per vector. This has a negative effect on classification accuracy, since there is less data for the classification algorithm to use. Increasing the feature extraction reliability would increase the algorithm’s ability to distinguish between activities.

While the problems described above do hurt the performance of the algorithm, it is important to compare it to other work to see how it fits in to existing research. The following section looks at the performance in comparison with other similar algorithms.

### 4.5 Comparison to Other Work

Overall the algorithm’s classification was less accurate than that of other individual human activity recognition algorithms. The poor segmentation of some subjects and the missing objects in most frames severely limited its performance. Yet removing the badly
segmented subjects and classification accuracy to other algorithms by activity type shows that with a bit of improvement this algorithm could be on par with others.

When badly segmented subjects are not considered the algorithm achieved a 91% single person activity classification accuracy. This is only a few percentage points lower than results from other research. Boeheim [2] got an overall accuracy of 96% on the limited set of walking, running and jumping activities. Chen [12] classified 98% of slightly different activities, including sitting up, getting up, jumping, walking, crawling and push ups. In [11] Singh claimed 100% accuracy on walking, standing, sitting, squatting, pointing and lying down activities using silhouettes for identification.

These other single person activity recognition algorithms are no more complex than the algorithm proposed here. Overall the single person accuracy would need to increase by roughly 5% for this algorithm to be considered worthwhile.

In person-person activities the algorithm achieved 93% classification accuracy with badly segmented subjects removed. Park [5] was able to get an 86% recognition accuracy using only grayscale interaction videos, for activities like shaking hands and pointing. Sato [7] fared similarly at 86%, but his activities were performed in outdoor environments and consisted of tracking the movement between subjects, such as following, meeting or approaching. Ryoo [6] found that his algorithm classified 92% of approaching, departing, pointing, shaking hands, hugging, punching, kicking, and pushing activities. Ryoo’s work considered almost the exact same activities, and achieved nearly the same classification accuracy. In light of this, a 93% classification accuracy seems reasonable.
With all the subjects and with badly segmented subjects removed the person-object recognition accuracy was only 50%. In [9], Ryoo identified 91% of person-object activities, including carry, leave, steal, or throw-in-garbage. It is difficult to compare accuracy between this work and Ryoo’s because of the limited activity set tested here, and the object identification flaw discussed previously. Still, Ryoo’s work shows that recognition of person-object activities should achieve accuracies comparable with single person and person-person algorithms.

The research compared here varied widely in the types of techniques used. Aside from classification accuracy, there may be additional strengths or weaknesses for one algorithm or another, such as processing speed or scalability. The fact that the proposed algorithm in this thesis was designed to recognize single person, person-person, and person-object activities is one advantage this work has over existing approaches. The fusion of these different activity types makes the algorithm more general, so it is appropriate for more situations.

4.6 Real-Time Considerations

By and large the algorithm should work without modification in a real-time environment. The pipeline architecture lends itself to parallel processing, which would increase its performance greatly. The only additions necessary would be the code required to perform temporal segmentation of the activities. The very simplest algorithm would classify activities based on the most recent 30 to 60 frames. It would also need to identify when subjects enter and leave the scene, in order to reinitialize the pixel clustering algorithm to recognize the colors of the subjects’ clothing. If more accuracy on the start and stop times of activities were required the temporal segmentation code could incorporate techniques
that have been researched by others, such as motion trajectories [16], spatial flow discontinuities [15], and energy minima [19].

The detailed analysis of this research and the comparison of results to other papers revealed areas which could be the topic of future research. Chapter 5 draws several conclusions as a result of this research and enumerates a number of possible directions that this work could be taken in.
Chapter 5: Conclusions

This thesis investigated the use of a limb-level feature extraction algorithm for classification of single person, person-person and person-object activities. The algorithm utilized background subtraction, pixel clustering, and feature extraction to create feature vectors that a classifier used to identify activities.

Segmentation of the subjects into limbs did not improve classification accuracy over previous silhouette-based approaches, and actually caused the algorithm to fail in some situations. If problems with over- or undersegmentation were corrected, the classification accuracy would be comparable to other algorithms.

The six-segment body model, with the addition of a change in distance between subjects feature, was shown to be a valid approach for identifying person-person activities. It is likely that this model is applicable to person-object activities as well, but improved object recognition and tracking would be required to validate this.

The algorithm’s limitations, such as background subtraction failing on white shorts on a white background, or the difficulty in applying pose-based classification to activities distinguished primarily based on movement, were limited to specific subjects or activities. These problems are well defined but require future research to correct.

The video dataset was very useful in analyzing the algorithm’s performance, and is suitable for future work in human activity recognition. This contribution makes it much easier for others to gain an objective view of their work on a medium sized dataset.
5.1 Future Work

There are several key stages in the algorithm where improvements would provide a significant increase in classification accuracy. Since the primary sources of misclassification were poor skin identification, poor object identification and poor segmentation, changes to these areas would have the most impact on the overall performance of the algorithm.

The skin and object identification algorithms used only the most basic of techniques, so more sophisticated models and tracking mechanisms could greatly reduce the errors in the current algorithm. The pixel clustering algorithm could be modified to incorporate the tracking and correlation used by Park and Aggarwal [23] to improve segmentation. If classification accuracy were up to par, a real-time implementation would be worth considering. The design of this algorithm is appropriate for multi-core processors, so a multi-threaded approach could yield good performance.

This work can be further extended to recognize more activities. Provided that the problem areas of the algorithm are worked out, it would be interesting to see how easily the algorithm can be modified to work on another activity set.

The field of human activity recognition is still very young and has many different areas to explore. Some types of algorithms end up working well, while others do not. As more research is performed, computer scientists gain a deeper understanding of how computer vision can be applied to the problem. The development of robust, accurate, scalable, real-time, general-purpose human activity recognition algorithms will be an important milestone in computer vision research.
Bibliography


