LINGUISTIC SUMMARIZATION OF HUMAN ACTIVITY

A Dissertation presented to the Faculty of the Graduate School

University of Missouri

In Partial Fulfillment

Of the Requirements for the Degree

Doctor of Philosophy

---

by

DEREK ANDERSON

Dr. James Keller and Dr. Marjorie Skubic, Dissertation Supervisors

JULY 2010

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

LINGUISTIC SUMMARIZATION OF HUMAN ACTIVITY

Presented by Derek Anderson

A candidate for the degree of Doctor of Philosophy

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

____

Professor James Keller

____

Professor Marjorie Skubic

____

Professor Mihail Popescu

____

Professor Marilyn Rantz

____

Professor Zhihai He

Thank you Melissa, Elliott, and Stefan for learning to live with the unexpected. It is your love and support that has helped me remain focused and inspired at each step along this long journey. These have been the most exciting and rewarding years of my life.

ACKNOWLEDGEMENTS

I would like to thank my close friend and colleague Robert Luke. Countless debates during extended coffee breaks ended up being the catalyst for much of the following work. It is a rarity to find two people who can work together so well on topics they both enjoy. An interesting aspect of this relationship is the inability to ever initially agree on how to view or solve a problem. I expect that you will disagree with this.

I would like to thank my mentor, Dr. James Keller. I have matured due to the quality and impact of your teaching, leadership, and very generous exposure to leading professionals and organizations. In particular, it is your unique (fuzzy) approach to problem solving and advice to never accept anything as fact but to verify it oneself that I will remember. You have helped me discover what I want to do in life. It is now my job to figure out how to make it a reality and leave a mark.

I would like to thank Dr. Marjorie Skubic. When I first ventured into this bizarre world of graduate life you took me under your wing and taught me how to organize my thoughts, write professionally, and navigate through this complex but rewarding environment. I am very grateful for all the support and doors you opened.

Lastly, I would like to thank Dr. Marilyn Rantz for working with us crazy Engineers and making this important work possible with your expert knowledge. You have been great to work with and you have taught me much about collaborative research.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS ii

LIST OF ILLUSTRATIONS vi

LIST OF TABLES xi

ABSTRACT xiv

Chapter

1. INTRODUCTION 1
    1) Motivation
    2) Overview
    3) Novel Research Contributions
2. BACKGROUND 6
    1) Introduction
    2) Fuzzy Set Theory
    3) Fuzzy Logic
    4) Change Detection and Human Segmentation from Video
    5) Graphical Models
    6) Related Work
    7) Summary
3. HUMAN SEGMENTATION IN IMAGE SPACE 72
    1) Introduction
    2) Texture Features
    3) Color Histogram Features
    4) Change Detection
    5) Background Update
    6) Results
    7) Summary
4. VOLUME ELEMENT SPACE 93

1) Introduction
2) Construction
3) Quality of Construction
4) Post-Processing
5) Feature Extraction
6) Environment Partitioning and Object Interaction
7) Summary
5. HUMAN STATE AND ACTIVITY RECOGNITION.................................... 123
1) Introduction
2) Hierarchical Framework for Linguistic Summarization
3) State Inference
4) Linguistic Summarization of State
5) Activity Inference and Summarization
6) Alert Generation
7) Hidden Markov Models
8) Summary
6. EXPERIMENTS.. 145
1) Introduction
2) Hierarchical Framework for Linguistic Summarization
3) Hidden Markov Model
4) Data Sets
5) Evaluation Metrics
6) Ground Truth
7) Summarization of Unknown State and Activity
7. RESULTS AND DISCUSSION.. 168
1) Introduction
2) State
3) Activity
4) Summary
8. CONCLUSIONS 189
1) Summary

2) Future Work

BIBLIOGRAPHY ………………………………………………………………………………………… 191

VITA ……………………………………………………………………………………………………… 197

LIST OF ILLUSTRATIONS

Figure Page

1. 1 High-level overview of the proposed human activity analysis framework. Blue components are problems investigated in this dissertation and light gray is preliminary work. 4

2.1 Support of a fuzzy set and a crisp set for the concept *tall* human, whose domain is inches. 8

2.2 Illustration of the fuzzy membership functions μ_A^T, μ_A^S, and μ_A^Z. 10

2.3 Illustration of the standard fuzzy operators complement, union, and intersection. 12

2.4 Illustration of the hedge concepts very, more or less, a little, and extremely. 14

2.5 Illustration of a fuzzy inference engine. Inputs, typically crisp, are fed into the system, crisp values are fuzzified, rules are fired using the rule base and the inference engine, and outputs are typically, but not always, defuzzified to obtain a crisp value that can be used by a system. 18

2.6 Computational stages in a Mamdani-Assilion fuzzy inference system (FIS). Rules are fired by calculating the Min of all the fuzzified inputs, the Min of the respective consequent fuzzy set and the rule firing strength is computed, the consequents are aggregated, using a Max operation at each discrete output element, and the centroid, e.g. defuzzification, is calculated. 19

2.7 Visualization of stages in the compositional rule of inference. 22

2.8 Stereo vision example. Image (a) is the raw camera output, (b) is the rectified image, (c) is the disparity map, and (d) is the three dimensional point cloud textured according to their image colors. Data was collected using Point Grey's bumblebee system. 34

2.9 Stereo vision-based change detection in voxel space. Images (a) and (b) show the camera viewing conditions, (c) is the initial stereo-based back-projection results, (d) is the refined voxel space, (e) is a new image containing the human, (f) is detected skin, (g) is skin voxels, (h) is head shape detection intersected with the set of skin voxels, (i) is a frame in which an object, the chair lowest in the image y (height) dimension, was recently moved and multiple initial change detection voxel islands are present, and (j) is the systems classification of the human (red) versus non-human objects (blue). 37

2.10 (a) Directed versus (b) undirected graphs. 41

2.11 Illustration of cliques in a graph. Vertices 1, 2, and 5 make up the largest clique, which is of size 3. All other pairs of vertices make up cliques of size 2. 42

2.12 Factorization of the joint distribution $P(E, F, G)$ into a directed graph. 43

2.13 DBN representation for an HMM with $T = 3$. Vertices whose names are prefixed with a Y are observed variables while vertices whose names are prefixed with an X are hidden variables. 47

2.14 An HMM represented as a DBN, in which all parameters are explicitly represented (nodes with outgoing dotted arcs). Notice that the parameters are stationary over time. 49

2.15 DBN with a mixture of Gaussians for the observed variables. 49

2.16 Hierarchy of different kinds of graphical models. 51

2.17 Global Markov property on sets V_A and V_B which are blocked by V_c. 52

2.18 Example MRF. 53

2.19 Standard MRF for low level computer vision tasks. The Y's are observations and the X's are hidden. 53

2.20 Example spatial and temporal silhouette features for kneeling (left) and walking (right) activities. Green pixels are the silhouette, light blue is the bounding box, purple and white lines are the motion vector (difference of means and medians respectively), and the red and blue ellipses are one and three standard deviation for the covariance matrix (again, with respect to the mean and median respectively). 58

2.21 Raw images, silhouettes, and silhouette features for standing, fallen, and bent over. Green regions are the foreground, red ellipses are the covariance tracked at one and three standard deviations, the light blue rectangle is the bounding box, and the purple lines are the mean-based motion vectors. 59

3.1 A graphical representation of the background model. (a) A sequence of ten images is input to the system. (b) A 3x3 mean filter is passed over the red, green and blue color planes of each input image. (c) Color and texture descriptors are then extracted from these images. (d) Finally, the means and standard deviations of the descriptors at each pixel are found over the input sequence. (e) During runtime, a 3x3 mean filter is passed over each new image. (f) Color and texture descriptors are then extracted from the image. (g) The foreground is then found for the new image using the background model. (h) The background model is then updated. 72

3.2 Silhouette extraction procedure. (a) Images are captured from a video camera in the sensor network. (b) A 3x3 mean filter is passed over the red, green and blue color planes of each input image. (c) Silhouettes are found in color and texture features. (d) Shadow regions are identified. (e) Shadows are removed from the output of silhouettes found from color features. (f) The results of the texture and color silhouettes are fused. (g) Morphological and logical operators are applied to the output of silhouettes to remove false alarms and fill-in silhouettes. (h) The fused data is then dilated to correspond to the size of the subjects in the image. (i) The final silhouette output image. 73

3.3 Images representing the amount of change according to each texture and color descriptors. (a) Background image (b) Test image (c) HS_v features Diff Image (d) C_b' features Diff Image (e) C_r' features Diff Image (f) C_g' features Diff Image 81

3.4 Multiple stages of change detection using color descriptors. (a) The confidence of change detection in color space. Notice that shadows are detected as change. (b) Pixels that have registered a change above the threshold of 2. (c) Shadows in the scene. (d) The change with respect to color descriptors after removing shadows and performing morphological operations. Although the output does not match the silhouette closely, it fills in parts of the silhouette missed by the texture descriptor. 85

3.5 A test sequence of images. Frames (a), (b) and (c) are three unprocessed images from the test sequence; (d), (e) and (f) are the hand-segmented silhouettes of the person; (g), (h) and (i) are the three silhouette images output from this system. 88

3.6 A test sequence of images. Frames (a), (b) and (c) are three unprocessed images from the test sequence; (d), (e) and (f) are the hand-segmented silhouettes of the person; (g), (h) and (i) are the three silhouette images output from this system. 89

3.7 A test sequence of images. Frames (a) and (d) are two unprocessed images from the test sequence; (b) and (e) are the hand-segmented silhouettes of the person; (c) and (f) are the two silhouette images output from this system. 90

3.8 Several images comparing the output of the Gaussian Mixture Model using Cb'Cr'Cg' color space, (top images), and those from the system defined in this paper, (bottom images). 91

4.1 Three-dimensional 8x4x7 volume element (voxel) space. 93

4.2 Illustration of camera pixel view ray testing (a)(b) versus pixel view volume-based testing (c). The red dotted lines are the camera pixel view vectors and the blue truncated pyramid is the pixel view volume. Pixel ray testing results in fewer overall voxels. 95

4.3 Illustration of the ray-sphere intersection test. The $\vec{v}_{t,j}$ voxel would be rejected by $\vec{R}_{c,m,n}$. 97

4.4 Illustration of the back-projection of a single pixel ray vector into voxel space. 98

4.5 Voxel person construction. Cameras capture the raw video from different viewpoints, silhouette extraction is performed for each camera, voxel sets are calculated from the silhouettes for each camera, and the voxel sets are intersected to calculate voxel person. 99

4.6 Raw images, (b) and (d), silhouettes, (a) and (c). Image (e) is the back-projection of the silhouettes, shown as blue and green voxel sets. Image (f) is voxel person, i.e. the intersection of the two camera silhouette pixel-voxel list sets. 100

4.7 (a) Visible shell, shown in green, and (b) visible shell along with the back-projected intersected object, shown in blue. 102

4.8 Visible and non-visible object error. Yellow, blue, and orange areas are the silhouette back-projected object. Blue is actual object, yellow is visible error, and orange is non-visible error. 103

4.9 Example scene containing multiple objects. Red, blue, and green circles are true objects. Orange islands are void space error. Gray regions are visible and non-visible object error. 103

4.10 Self-occlusion back-projection error. Yellow area is visible error, orange is non-visible error, blue is the true object, and red is self-occluded error. 104

4.11 Non-white circles are cameras and each is respectively color coded. White circles and gray regions are the back-projected object. In (a), a two camera setup, purple is the joint monitored space. In (b), a three camera setup, white is the joint monitored space. As the number of cameras increase, the joint viewable space decreases. Images (c), (d), (e), and (f) show camera color-coded back-projected view volumes. In the case that an object is in the middle of the viewing region, (c) and (d), the two and three camera setups are very similar. However, when the object is placed at another location in joint viewable space, (f), the superiority of the three camera setup is apparent. 106

4.12 Quality of construction based on orthogonality of camera center pixel view vectors. 106

4.13 Blanketed set for an ideal case of near orthogonal viewing construction of a human. Images (a) and (c) are the raw images, (b) and (d) are the silhouettes, (e) are the silhouette back-projections, (f) is the visible shell (red) and intersected set (gray), and (g) is the blanketed set. Take note of the loss of some true object volume. However, features, such as orientation, centroid, and height are still valid. 108

4.14 Blanketed set for a non-ideal case of viewing construction of a human. Images (a) and (c) are the raw images, (b) and (d) are the silhouettes, (e) are the silhouette back-projections, (f) is the visible shell (red) and intersected set (gray), and (g) and (h) are the blanketed set. Take note that the non-visible error in (f) is drastic. While the blanketed set, shown from two different angles, (g) and (h), has lost some true object volume, the resulting features, e.g. centroid, orientation, and height, are much more accurate. 110

4.15 (e)-(m) is nine horizontal (x-y plane) slices of voxel object construction quality for the camera configuration {(a),(b)} in (d). Brighter values in (e)-(m) represent higher quality. Map (e) is at a height of 1 foot, and each consecutive map is 1 foot higher in the z dimension (world up direction). 112

4.16 T-norm produced single x-y voxel plane for figure 4.15. 113

4.17 Example of low object construction quality, value of 0.05, for cameras {(a),(b)} in figure 4.15.d. The object is approximately half way along the line connecting cameras {(a),(b)}. Thus, a good majority of the object is constructed using view ray vectors pointing roughly towards each other. Green is the visible shell and blue is the intersected object. The object appears to be constructed properly when observed with respect to the two individual cameras views, (a) and (b). However, when the object is viewed from a different location in space, (c), it is apparent that the object is poorly constructed. 114

4.18 Dynamic selection of construction quality planes based on an objects current height for the case of predominantly downward viewing cameras. (a) is the t-norm produced plane for all nine planes in figure 4.15. (c) is the t-norm produced plane for only the bottom three planes, i.e. {(e),(g),(f)}. Red crosses are the location of the object in (b). (a) shows that the object has a very low quality and should be ignored, i.e. figure 4.17.c, while (c) shows that the object can be constructed with sufficient quality. 115

4.19 Examples illustrating the approximation of leg voxel sets during different activities. Assumed leg voxels are blue and a resolution of 5x5x5 inches is used. In (a), the human is standing. In (b), the humans legs and hands just touched the ground during a fall. In (c), the human is on the ground after a fall. The accuracy of this local feature depends on a systems ability to correctly infer context, i.e. decide if the person is standing, kneeling over, fallen, etc. 120

4.20 Partitioning of an environment into regions, whose adjacent edges are shown in red, and objects, shown in green, for activity analysis monitoring. 121

5.1 Linguistic summarization framework for human activity analysis. Video is collected from multiple cameras, silhouettes are extracted, and voxel persons are built. The next three steps are repeated for the number of activity levels tracked. For each activity level, features are extracted from voxel person and lower level activity summarizations, activity is inferred, and then it is linguistically summarized. ………… 125

5.2 Example state time series before (top) and after (bottom) the median filter. Here, Ʊ was chosen to be 3. ………… 129

5.3 Illustration of the assertion that human activity is naturally hierarchical. Shaded truncated pyramids show nested structure. For example, limping and brisk-walk are types of walking, walking and standing depend on the state upright, and upright depends on voxel person features. ………… 133

6.1 Fuzzy inference outputs for states plotted for a voxel person fall. The x-axis is time, measured in frames, and the y-axis is the fuzzy inference outputs (activity confidence). The red state sequence is **upright**, blue is **in-between**, and green is **on-the-ground**. The frame rate is 3 fps. The purple dashed vertical line is the manually identification of the location of a fall. ………… 147

6.2 Images from the student data set. Rows 1 and 2 are **SET1.a**, row 3 is **SET1.b** and row 4 is **SET1.c**. ………… 159

6.3 Example raw images (row 1), silhouettes (row 2), and voxel person (row 3) for a fall from the stunt actor data set. ………… 161

6.4 Example potential false alarm activities from **SET2.d**. ………… 162

6.5 Approach taken to measuring similarity between human ground truth intervals and system summaries. The x-axis is time. The summaries of interest are shown in green. Intervals in (a) are ground truth and (b) shows each system summary mapped to the human interval it overlaps with the most. In (c), all relevant summaries are contained in the human's interval, label 2, but an undesirable number are present. In (d), all summaries are correctly contained in the human's interval, there are not *very many* of them, and together they account for a *good* percentage of the human's total interval. In (e), the first green interval is a false alarm and the second green interval is a correct match. Cases (b), (d) and (e) are positive Metric 4 scenarios. ………… 164

7.1 Plot of inferred state for the stunt actor fall data set **SET2.a**. The state **upright** is red, **in-between** is yellow, **on-the-ground** is green, **lying-on-the-couch** is blue, and **on-the-chair** is purple. 169

7.2 Plot of inferred state and inferred activity for the stunt actor fall data set **SET2.a**. In the top image, the state **upright** is red, **in-between** is yellow, **on-the-ground** is green, **lying-on-the-couch** is blue, and **on-the-chair** is purple. In the bottom image, the activity **fall** is red, **standing** is yellow, **walking** is green, **lounging-on-the-couch** is blue, and **relaxing-on-the-chair** is purple. 178

7.3 Illustration of a significant type of frame-by-frame activity error encountered. At each frame, in (a), (b), and (c), the activity with the maximum membership grade is selected. Based on this index, a plot is created for the linguistic summarization system (green) versus the human's labels (blue). Moments of disagreement, which typically occur at summary endpoints, are indicated by red dashed lines. Images (b) and (c) are smaller time intervals from (a). They help simplify the understanding of this type of error. ………… 182

LIST OF TABLES

Table Page

3.1 Accuracy of the system described in this dissertation and a Gaussian Mixture Model. …………… 91

6.1 Antecedent linguistic variables and terms used for the inference of pose-based states. Each variable has its input clamped to the specified domain. ………………………………………………… 147

6.2 Terms for the consequent linguistic variables **upright**, **in-between**, **on-the-ground**, **on-the-chair**, and **lying-on-the-couch**. ……………………………………………………………………………… 148

6.3 Rules for recognizing the pose-based states **upright**, **in-between**, and **on-the-ground**. ………… 148

6.4 Rules for recognizing states based on object interaction and pose. ……………………………………… 150

6.5 Antecedent linguistic variables and terms used for the inference of pose and object interaction-based states. Input is clamped to the domain [0,1]. ……………………………………………… 150

6.6 Terms identified by the nurses for the linguistic variable time duration. Input is clamped to the domain [0, 86400]. ………………………………………………………………………………………… 151

6.7 Terms for the consequent linguistic variables **fall**, **standing**, **walking**, **relaxing-on-the-chair**, and **lounging-on-the-couch**. ………………………… 152

6.8 Linguistic variable terms for summary activity confidence ($\pi_{A^l_{max},Sum'_{l,g}}$) with domain [0,1]. …. 153

6.9 Linguistic variable terms for distance traveled per second in the current summary ($\varphi^{\vec{c}_t}_{Sum'_{l,g}}$). Input is clamped to the domain [0,2]. ………………………………………………………………… 153

6.10 Linguistic variable terms for number of seconds taken to move a fixed distance with respect to the current summary ($\vartheta^{\vec{c}^{Legs}_{t,(x,y)}}_{1,30}$). Input is clamped to the domain [0,10]. ……………………………… 153

6.11 Linguistic variable terms for recent relative difference change in voxel person's speed. The minimum of the feature and the value 1 is computed. Input is clamped to the domain [0,1]. 154

6.12 Linguistic variable terms for recent oscillating behavior, $OSC_{moderate,A^1_2,A^1_3}$. Input is clamped to the domain [0,8]. …………………………………………………………………………………………… 154

6.13 Rules for **fall** recognition, created in collaboration with the nursing staff. ……………………………. 155

6.14 Rules used for recognizing **standing** and **walking**. 156

6.15 Rules used for recognizing **relaxing-on-the-chair** and **lounging-on-the-couch**. 156

6.16 Inhibitory rules used for conflict resolution. 157

6.17 Number of state labels for all data sets. 165

6.18 Number of actiivty labels for all data sets. 165

7.1 Comparison between state summaries for the linguistic summarization system, for summary confidences τ_2 and τ_5, and the human ground truth for all data sets. 170

7.2 Comparison between state summaries for the linguistic summarization system, for summary confidences τ_2 and τ_5, and the human ground truth for the false alarm data set **SET2.d**. 171

7.3 Frame-by-frame state decision comparison between the linguistic summarization system and the human ground truth for all data sets and $M(i,j)/\sum_{j=1}^{6} M(i,j)$. 173

7.4 Percentage of frame-by-frame decisions remaining after linguistic summarization of state for all data sets. The lower the value the better the system performance. 174

7.5 Maxtirx cell format for tables 7.6 and 7.7. The left column is for $\pi_{A_i^l, Sum_{l,g}'} > \tau_2$ and the right column is for $\pi_{A_i^l, Sum_{l,g}'} > \tau_5$. While tables 7.6 and 7.7 include different combinations of activities, depending on what is present in the data set, the order and color of entries does not change. 174

7.6 Variation of metric 4 matching success criteria for the linguistic summarization system and the human ground truth for all data sets. 175

7.7 Variation of the metric 4 matching success criteria for the linguistic summarization system and the human ground truth for the stunt actor fall data set, {**SET2.a**, **SET2.b**, **SET2.c**}. Only **upright**, **in-between**, and **on-the-ground** are shown because the other states are not performed for these data sets. 176

7.8 Comparison between activity summaries for the linguistic summarization system, for summary confidences τ_2 and τ_5, the HMM system, and the human ground truth for all data sets. 178

7.9 Frame-by-frame activity decision comparison between the linguistic summarization system, HMM-based system, and the human ground truth for all data sets and $M(i,j)/\sum_{j=1}^{6} M(i,j)$. 180

7.10 Percentage of frame-by-frame decisions remaining after linguistic summarization fo activity for all data sets. The lower the value the better. 182

7.11 Maxtirx cell format for tables 7.12 and 7.13. 183

7.12 Variation of metric 4 matching success criteria for the linguistic summarization system, for $\pi_{A_i^l, Sum_{l,g}'} > \tau_2$, and the human ground truth for all data sets. This metric should be analyzed in conjunction with 7.8 and 7.9, which highlight false alarms. 184

7.13 Variation of the metric 4 matching success criteria for the linguistic summarization system, for $\pi_{A_i^l, Sum'_{l,g}} > \tau_5$, and the human ground truth for all data sets. This metric should be analyzed in conjunction with 7.8 and 7.9, which highlight false alarms. ………………………………………………….. 185

7.14 Confusion matrix for the activity **fall** according to the linguistic summarization system and the HMM-based system with respect to the human ground truth for all data sets. …………………… 186

ABSTRACT

The thesis advanced herein is that linguistic summarization is essential for the reliable succinct modeling and inference of human activity. It is also asserted that the inherent and unavoidable uncertainty is linguistic and fuzzy. Advantages of the proposed work include the generation of human interpretable confidence values, improved rejection of unknown activity, information reduction, complexity management, and the recognition of adverse events. Specifically, a computer vision-based hierarchical soft-computing linguistic summarization framework is proposed. First, images are summarized through the identification of a human and a three-dimensional object called voxel person is constructed. Next, approximate reasoning is used to linguistically summarize the state of the human at each moment, i.e. image, using features extracted from voxel person. Subsequently, temporal linguistic summarizations are produced from the state membership time series. State summaries are used to infer activity, which are also linguistically summarized and subsequently used in a hierarchical similar fashion to recognize additional specific types of higher level activity. A system comprised of two levels is described for the goal of elderly activity recognition. The system parameters are designed under the supervision of nurses. The results are compared to probabilistic graphical models for three data sets consisting of student and nurse trained and supervised stunt actor activities.

Introduction

1.1 Motivation

The recognition of human activity is a trivial task for most humans. However, the design of a robust real-time computer vision system capable of yielding similar meaningful function is anything but trivial. In order to make this high-level goal a reality, one must sufficiently address multiple unsolved areas of computer vision. Computer vision is generally discussed in terms of low-level, e.g. change detection, medium-level, e.g. region and object extraction, and high-level tasks, e.g. behavior analysis. In the abstract, this task can be conceptualized as a chain of mathematical transformations between real world observations, e.g. raster images acquired from multiple cameras over the visible band of the electromagnetic spectrum, to internal artificial summarizations regarding objects and their activity. The research outlined herein has application to artificial intelligence in the broad sense for the areas of knowledge representation [1][2] and computing with linguistic summarizations [2][3]. More specifically, it advances the area of surveillance, e.g. security [4][5][6][7][8][9][10][11][12] or "well-being" monitoring of at-risk populations for elders [13][2][1][14][15][16][17][18][19][20][21][22]. While the majority approach automated activity analysis as an instance of sequential data analysis, finite state automaton, and probability theory [23][24][25][19][20][26], it is viewed here as a hierarchical utilization of fuzzy set theory, approximate reasoning, and linguistic summarization [2][1][18][17][15].

In this work, focus is placed on more than just objective measurable system performance/output, such as a confusion matrix for fall recognition. Significant areas such as system tractability, understandability of system inner workings and its acquired knowledge, ability for humans to understand the output and interface with a system, as well as generalizability of the system for addressing other tasks are of importance. The goal is not to literally reproduce how humans perform activity analysis or search for a black box that merely performs function approximation. This work is a look into more significant questions, such as what is human activity; what low-level elements are needed in order to model and recognize human activity; is activity a binary concept or something that is always occurring to a degree; is activity recognition the same according to different individuals; how does context influence activity recognition; how is a computational process and its results compared to human performances; what is the mathematical nature of the domain uncertainty and how is it incorporated; and how is the soundness and correctness of one hypothesis versus another tested? While this thesis is used to derive a system for a specific task, i.e. recognition of falls from elders, the thesis is a broader platform for learning and advancing computer vision.

1.2 Overview

The field of computer vision suffers from information overload and redundancy. It is common to have multiple cameras monitoring a single environment (a living room), multiple environments per installation (an apartment), 307,200 pixels per image for a 640x480 image resolution, 10 or more frames per second, multiple channels per pixel, and seconds, minutes, or days of video. As a result, real-time processing is very difficult to achieve and algorithmic scalability is a significant factor. Computer vision systems must be equipped with information reduction and summarization tools to identify or transform this otherwise initial large stream of redundant data into a considerably smaller amount of salient relevant information. This thesis addresses summarization in the context of human activity analysis in the following five ways. Figure 1.1 shows the sequencing of these components.

1. The first (low and mid-level) computer vision step is change detection (figure 1.1.S2). In this, a privacy protected account of the human in image space is obtained [27].

2. The next (medium-level) computer vision task is object construction (figure 1.1.S3-S4). In this step, a volume element (voxel) object is built and refined using multiple cameras, silhouettes, and back-projection [1][2][28][16].
3. The next (high-level) computer vision task uses fuzzy sets to model features and fuzzy logic to infer activity at each moment/frame (figure 1.1.S5-S6) [1][18].
4. While the above steps reduce the large original stream of image pixels to fewer numbers of objects and decisions regarding the activity of those objects at each moment, they do not explicitly assist with the matter of information redundancy over time. The next (high-level) computer vision step is a reduction in the time domain into a significantly smaller number of rich succinct linguistic descriptions (figure 1.1.S7) [2].
5. The final task is the repeated application of feature extraction, fuzzy inference, and linguistic summarization (figure 1.1.S5-S7). The system presented in this dissertation for fall recognition undergoes two passes of linguistic summarization. The first pass yields summarizations of state, such as "Derek is *upright* in the living room for *a while*", while the second pass produces summarizations of activity, such as "Derek has *fallen* in the kitchen". For a particular pass, feature extraction is carried out on a variable length sliding window of voxel person and linguistic summarizations [2]. Concepts at higher levels of abstraction are built using lower level decisions, i.e. the activity *fall* is detected using state summarizations regarding *on-the-ground.*

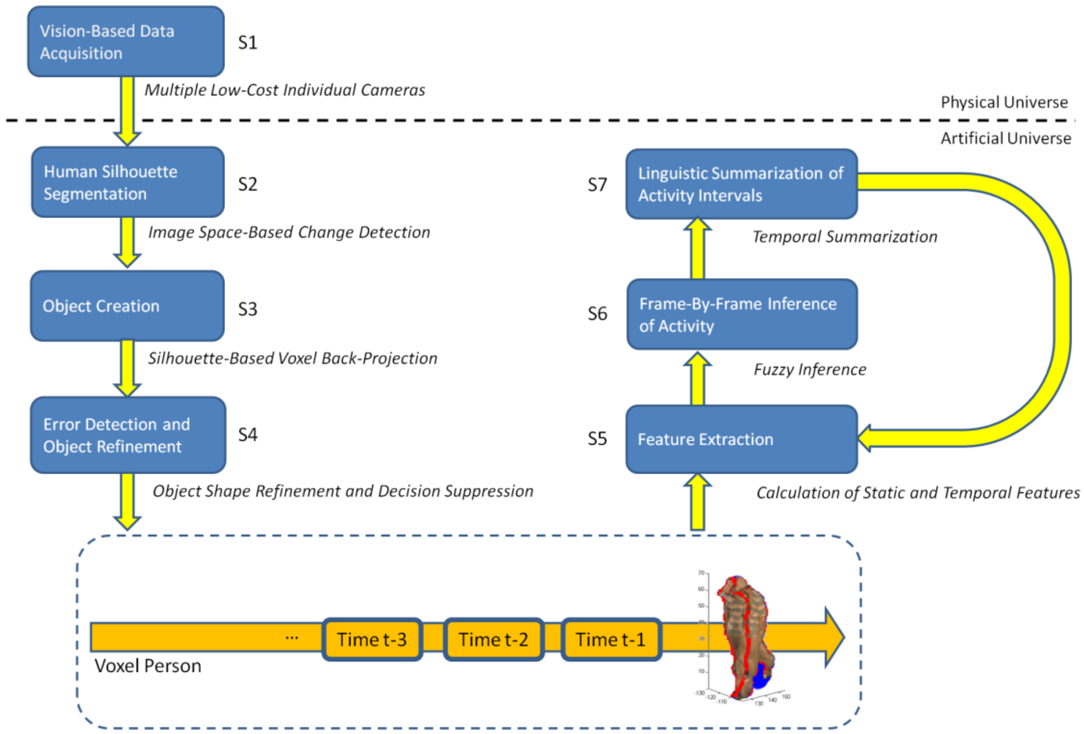


Fig. 1.1. High-level overview of the proposed human activity analysis framework. Blue components are problems investigated in this dissertation and light gray is preliminary work.

1.3 Novel Research Contributions

Producing and computing with linguistic summarizations of video is the major contribution of this work. While steps 1 and 2 (figure 1.1.S2-S5) are advanced herein, steps 3-5 (figure 1.1.S6-S7) are the most novel. The following is a research contribution breakdown:

- Human object construction and feature extraction in volume element (voxel) space
- Adoption of fuzzy set theory and approximate reasoning for addressing the natural uncertainty inherent in the robust modeling and inference of human activity

- Production and interpretation of state and activity confidences, which are useful for detecting alert conditions and rejecting a larger class of unknown human activity
- Linguistic summarization of human activity for understanding and reporting decisions made by a computational system to humans
- Linguistic summarization of human activity for the tasks of information reduction and complexity management
- Hierarchical framework for computing and reasoning with linguistic summarizations.

Background

2.1 Introduction

This dissertation is possible because of two significant theories. The major contribution of this dissertation, linguistic summarization of video, is above all based on fuzzy set theory and fuzzy logic. The second significant area is change detection, which is necessary in order to find people in video. A brief and relevant background to these important areas is provided in this chapter.

In addition, a background to probabilistic graphical models, specifically hidden Markov models (HMMs), and other related activity analysis work is provided. This makes it possible to compare and contrast important theoretical differences between this dissertation and the predominant existing theory of human activity analysis. The linguistic summarization system developed here is also compared in the results section to HMMs.

2.2.1 Fuzzy Set Theory – Introduction and Basic Concepts

Uncertainty is an unavoidable aspect of our physical and artificial universes, or more importantly of our ability to measure, model and infer. A well known physical example comes from the area of quantum mechanics. Heisenberg's principle of uncertainty states that one cannot simultaneously know both the position and the momentum of a given object to arbitrary precision, thus demonstrating fundamental limitations of measurement [29].

Uncertainty is also inescapable in the area of computer vision. David Marr's principle of least commitment stresses that it is costly to have to later undo mistakes that are a result of over committing earlier [30]. There is much utility in algorithmically embracing uncertainty, e.g. iteratively making *soft* versus *hard* (binary) decisions/commitments. This philosophy is not restricted in any form to computer vision. An example of this theory in the area of clustering is James Bezdek's fuzzy c-means [31].

Of specific interest to this work is the observation that uncertainty is unavoidable in human activity analysis. This conclusion is based on the reality that activity definitions are non-crisp in nature, subjective, and context and domain specific. No two individuals have the exact same understanding of an activity. The problems of internal representation, observation, inference, and transitioning between activities are a few examples of the inherent uncertainty. The reliable recognition of human activity over long time periods in dynamic environments requires uncertainty not to be suppressed but incorporated into the design of the system. The majority of related existing activity analysis theory is based on the principle that uncertainty can be solely modeled using probability theory [21][22][23][32][20][26]. While the frequentist view of probability is of great utility to science, it is not able to address all types of uncertainty in the domain of activity analysis. The different approach introduced here is based on fuzzy sets, linguistic variables, and the compositional rule of inference.

Fuzzy set theory, introduced by Lotfi A. Zadeh in 1965 [33], is an extension of classical set theory. The memberships of elements in a set are allowed to vary in their degree, e.g. $[0,1]$, instead of being restricted to two values, $\{0,1\}$, as in classical set theory. The universe of discourse, X, is any collection of objects, concepts, or mathematical constructs. For example, X may be the set of real numbers, $\mathbb{R}$, the set of natural numbers, N, a discrete set of colors, $\{red, green, blue, orange, \dots\}$, etc. A fuzzy set A is a fuzzy subset of X, $A \subseteq \mathrm{X}$, characterized by a membership function

$$\mu_A \colon \mathrm{X} \to [0,1],$$

where $\mu_{\emptyset}(x) = 0$ and $\mu_{\mathrm{X}}(x) = 1$, $\forall x$. For a particular element, $x \in \mathrm{X}$, the membership is typically denoted as $\mu_A(x)$ or $A(x)$. A fuzzy set is $normal$ if its $height$ is equal to 1, where $height$ is defined as

$$height(A) = supremum_{x \in X}(\mu_A(x)).$$

The $support$ of a fuzzy set is

$$support(A) = \{x | x \in X \text{ and } \mu_A(x) > 0\}.$$

The $support$ of A is a set of elements which are the elements that belong to some degree, $\mu_A(x) > 0$, to the fuzzy set A. Figure 2.1 illustrates fuzzy and crisp sets for the concepts *medium* and *tall* human, whose domain is inches.

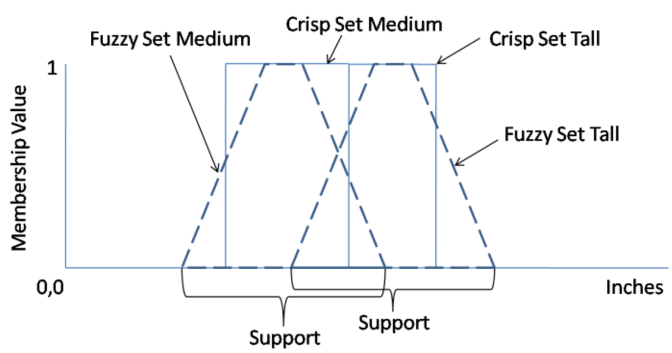


Fig. 2.1. Support of a fuzzy set and a crisp set for the concept *tall* human, whose domain is inches.

As it relates to human perception, Zadeh has examined the uncertainty inherent in linguistic concepts, such as the fuzziness in describing a person's height, its role in approximate reasoning, and more recently its role in computing with words and perceptions [34][35][36][37][3]. Uncertainty can manifest itself in a variety of ways, including imprecision, nonspecificity, vagueness, inconsistency, etc [38]. Fuzzy sets provide a way to address different types of uncertainty, as probability theory provides a way to address likelihood, frequency, proportion, or strength of belief [39]. A classical example of the difference is the problem of the potability of a liquid, posed by Bezdek in [40][41]. In this example, an individual has been in the desert for a week without drink and comes across two bottles. Bottle A has a membership degree in the class of potable ("suitable for drinking") liquids of 0.91, while bottle B has a probability of 0.91%, meaning that over a long run of experiments, 9% of these experiments had a liquid that was not potable. When presented with these two bottles and the descriptions of their contents, one would perceive a fuzzy membership of 0.91 as rather high and similar to that of a perfectly potable liquid, while a probability of 0.91% means that the drinker has roughly a one in ten chance of drinking something not potable. Under these circumstances, Bezdek describes that most rational subjects would prefer bottle A.

In [39][42], Zadeh draws important relationships between possibility theory and fuzzy sets, motivated by a desire to use it for discussing the meaning contained in natural languages and fuzzy propositions. Specifically, Zadeh discusses connections between possibility measures, possibility distributions, and fuzzy restrictions. This interpretation also makes it possible to further study additional important distinctions between fuzzy and probability theory.

Common and relevant membership functions include: trapezoidal (μ_A^T), s-shaped (μ_A^S), and z-shaped (μ_A^Z) membership functions (shown in figure 2.2). The membership of an element $x \in \mathrm{X}$ in the fuzzy set A, according to μ_A^T, μ_A^S, and μ_A^Z, is

$$\mu_A^T(x) = maximum\left(minimum\left(\frac{(x-a)}{(b-a)}, 1, \frac{(d-x)}{(d-c)}\right), 0\right)$$

$$\mu_A^Z(x) = \begin{cases} 1 & x \le a \\ 1 - 2\left(\frac{x-a}{b-a}\right)^2 & a \le x \le \frac{a+b}{2} \\ 2\left(\frac{b-x}{b-a}\right)^2 & \frac{a+b}{2} \le x \le b \\ 0 & x \ge b \end{cases}$$

$$\mu_A^S(x) = \begin{cases} 0 & x \le a \\ 2\left(\frac{x-a}{b-a}\right)^2 & a \le x \le \frac{a+b}{2} \\ 1 - 2\left(\frac{b-x}{b-a}\right)^2 & \frac{a+b}{2} \le x \le b \\ 1 & x \ge b \end{cases}$$

where $a \leq b \leq c \leq d$ are real-valued parameters of the various membership functions.

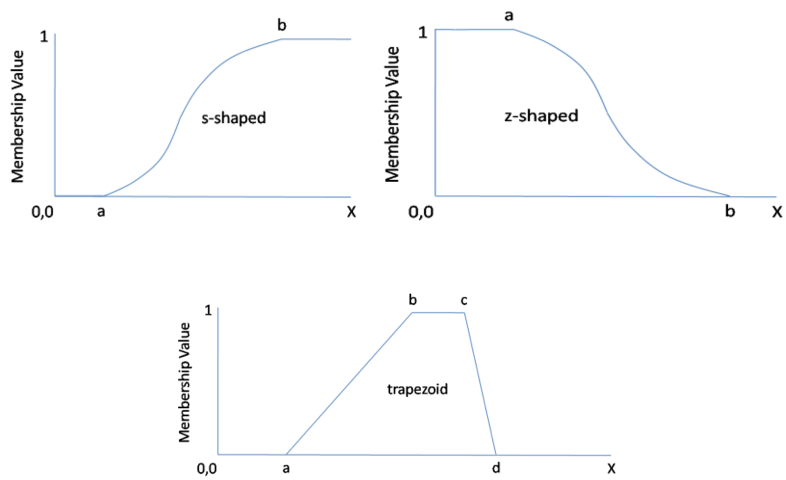


Fig. 2.2. Illustration of the fuzzy membership functions μ_A^T, μ_A^S, and μ_A^Z.

2.2.2 Fuzzy Set Theory – Standard Fuzzy Operators

Standard operations on fuzzy sets include the t-norm (intersection) and t-conorm (union), binary operators, as well as the unary operation of complement. Others, such as aggregate operators, can be found in [38]. The complement of a fuzzy set A, $\mu_{\bar{A}}(x)$, is a function that performs the mapping

$$\mu_{\bar{A}}: [0,1] \to [0,1],$$

whose axioms can be found in [38][33]. In the crisp case, $\mu_A(x) \to \{0,1\}$, the complement is the set of all elements in X that do not belong to A. In the more general fuzzy case, this function expresses the degree to which an element in X does not belong to A. The standard complement operator is

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x).$$

A family of complement operators, such as the well known Sugeno and Yager operators, and their axioms, can be found in [38].

The next operator is the t-norm (intersection). Give two fuzzy sets, A and B, where $A, B \subseteq \mathrm{X}$, $\mu_{A\cap B}(x)$, the t-norm is the mapping

$$\mu_{A\cap B}: [0,1] \times [0,1] \rightarrow [0,1],$$

satisfying appropiate axioms [38], where $\times$ denotes the Cartesian product. The t-norm describes how much an element belongs to both A and B. The standard t-norm is the Min (minimum),

$$\mu_{A\cap B}(x) = Min[\mu_A(x), \mu_B(x)].$$

Other operators, such as the algebraic product, bounded difference, and others can be found in [38].

The last standard operator to be discussed is the t-conorm (union). The t-conorm of the two fuzzy sets A and B, $\mu_{A\cup B}(x)$, where $A, B \subseteq \mathrm{X}$, is the mapping

$$\mu_{A\cup B}: [0,1] \times [0,1] \rightarrow [0,1],$$

satisfying suitable axioms [38]. The t-conorm describes how much an element belongs to either A or B. The standard t-conorm is the Max (maximum),

$$\mu_{A\cup B}(x) = Max[\mu_A(x), \mu_B(x)].$$

Other common operators, such as the algebraic sum, bounded sum, etc, can be found in [38]. Graphically, these three operators are illustrated in Fig. 2.3.

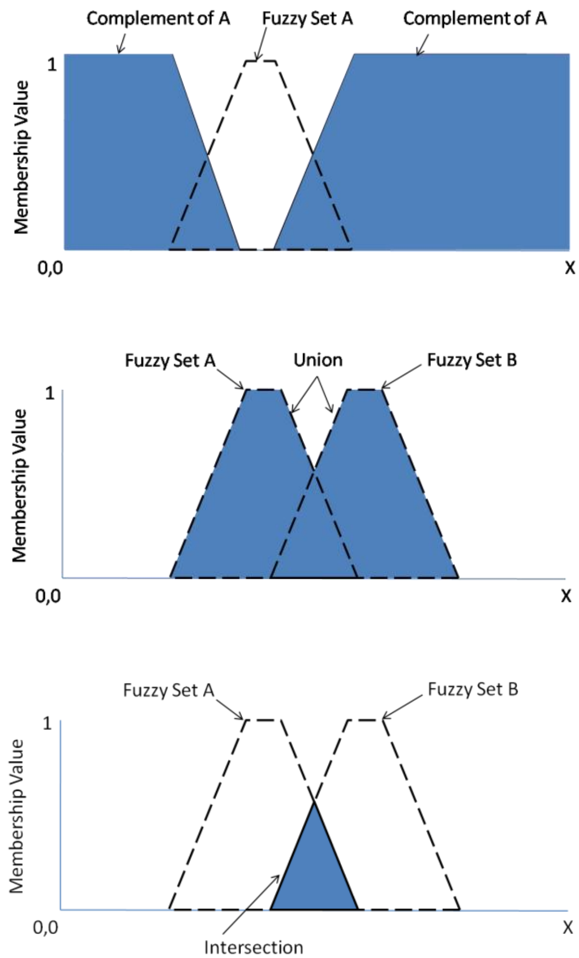


Fig. 2.3. Illustration of the standard fuzzy operators complement, union, and intersection.

An important aspect of fuzzy set theory stems from the exact choice of $(c_A, i_{A \wedge B}, u_{A \vee B})$. Of particular interest are the laws of the excluded middle (LEM) and contradiction (LOC). The LEM is

$$A \cup \bar{A} = \mathrm{X},$$

and the LOC is

$$A \cap \bar{A} = \phi.$$

While these two laws are satisfied in classical crisp set theory, the selection of many fuzzy operator triples, for example $(c_A, i_{A \wedge B}, u_{A \vee B}) = (1 - \mu_A(x), Min[\mu_A(x), \mu_B(x)], Max[\mu_A(x), \mu_B(x)])$, does not satisfy the LEM and the LOC. Consider $\mu_A(x) = 0.7$ and the LEM, then

$$\mu_A(x) \vee c_A(\mu_A(x)) = Max\big(\mu_A(x), 1 - \mu_A(x)\big) = Max(0.7, 0.3) = 0.7.$$

In addition, for the LOC,

$$\mu_A(x) \wedge c_A(\mu_A(x)) = Min\big(\mu_A(x), 1 - \mu_A(x)\big) = Min(0.7, 0.3) = 0.3.$$

An analysis of what properties of classical set theory do not extend to a given fuzzy set theory can be found in [38][33].

2.2.3 Fuzzy Set Theory – Linguistic Variables and Terms

A simple yet powerful concept is that of a linguistic variable. A linguistic variable is given by a quintuple (v, T, X, g, m) [34][35][36][37]. The name of the linguistic variable is v, T is a set of linguistic terms (names of linguistic values of v), X is the universe of discourse, g is a syntactic rule (grammar) for generating the names of values of X, and m is a semantic rule that assigns to each linguistic term $t \in T$ its meaning, $m(t)$, which is a fuzzy set on X. Alternatively, a linguistic variable can be described in terms of fuzzy restrictions. Let $R(v)$ denote a fuzzy restriction associated with variable v. In order to express the notion that t, $t \in T$, a fuzzy subset of X, plays the role of a fuzzy restriction in relation to v, $R(v) = t$ is written. A fuzzy set can be thought of as an elastic constraint on the values that may be assigned to a linguistic variable. An example is the linguistic variable “height of a person’s centroid”, in which the linguistic terms low, medium, and high, are all defined as membership functions over the domain $X = [0\ inches,\ 96\ inches]$.

An important related concept is that of a hedge [43]. Informally, a hedge is a modifier to a term. In the case of the linguistic variable height of a person's centroid and the term set $\{low, medium, high\}$, example hedges might include $\{very, more\ or\ less, a\ little, extremely\}$. Therefore, when describing height, descriptions such as "very high", "more or less high", etc can be used. Formally, a hedge, h_T, is a function that performs the mapping

$$h_T: [0,1] \rightarrow [0,1],$$

where T is a fuzzy set and used instead of A to emphasize a linguistic hedge. For example, if the membership function for $high$ is μ_{high} their example hedges include

$$\mu_{very\ high} = \left(\mu_{high}(x)\right)^2,$$

$$\mu_{more\ or\ less\ high} = \left(\mu_{high}(x)\right)^{\frac{1}{2}},$$

$$\mu_{a\ little\ high} = \left(\mu_{high}(x)\right)^{1.3},$$

$$\mu_{extremely} = \left(\mu_{high}(x)\right)^3.$$

Figure 2.4 shows mathematical representations for the above list of hedge concepts for a triangular membership function (a trapezoid membership function where $b = c$).

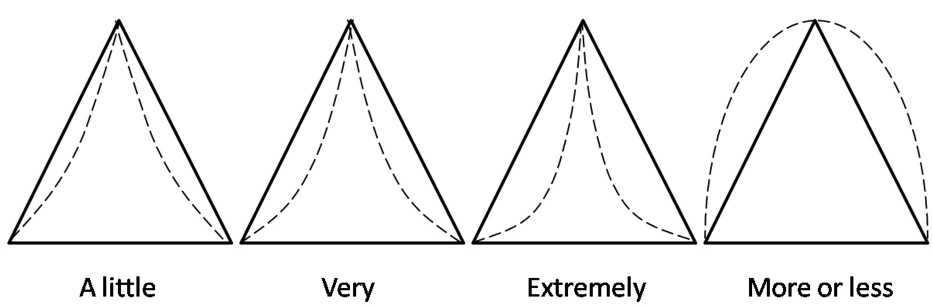


Fig. 2.4. Illustration of the hedge concepts very, more or less, a little, and extremely.

2.2.4 Fuzzy Set Theory – Fuzzy Relations and Composition

An important concept necessary for fuzzy logic is that of a fuzzy relation. Before providing the definition of a fuzzy relation, an understanding of the Cartesian product is required. The binary Cartesian product is defined as

$$X \times Y = \{(x, y) | \, x \in X, y \in Y\}$$

This is easily extended to m sets. Given an arbitrary number of sets, $X_1, \dots, X_m$, the set of all m-tuples $(x_1, \dots, x_m)$, such that $x_1 \in X_1, \dots, x_m \in X_m$, is written as $X_1 \times X_2 \times \dots \times X_m$, or

$$\times_{i=1}^{m} X_i,$$

and when $X_i = X_j$, for $\forall i, j \in \{1, \dots, m\}$, it is denoted by X^m. A crisp binary relation, R, is subset of $X \times Y$, i.e. $R \subseteq X \times Y$, and the m-ary relation is denoted as $R \subseteq X_1 \times X_2 \times \dots \times X_m$. In [33][34], Zadeh extended the notion of a crisp relation to a fuzzy relation. A fuzzy relation is

$$R = \{\big((x_1, \dots, x_m), \mu_R(x_1, \dots, x_m)\big) \big| \, x_1 \in X_1 \dots \, x_m \in X_m\},$$

where $\mu_R(x_1, \dots, x_m)$ is a membership function.

Given two fuzzy relations, $R_1(x, y)$, $R_1 \subseteq X \times Y$, and $R_2(y, z)$, $R_2 \subseteq Y \times Z$, where $x \in \mathrm{X}$, $y \in \mathrm{Y}$, and $z \in \mathrm{Z}$, the composition of R_1 and R_2, $R_1 {}^\circ R_2$, is

$$\mu_{R_1 {}^\circ R_2}(x, z) = supremum_{y \in \mathrm{Y}}(\mu_{R_1}(\mathrm{x}, \mathrm{y}) \wedge \mu_{R_2}(\mathrm{y}, \mathrm{z})),$$

where $\wedge$ is a t-norm (typically $\wedge$ is the minimum).

A few last important concepts are that of cylindrical extension, projection, and cylindrical closure. The cylindrical extension of a fuzzy set $A \subseteq X$ to $\mathrm{X} \times \mathrm{Y}$, denoted as $[\mathrm{A} \uparrow \mathrm{X} \times \mathrm{Y}]$, is the “smearing” of A across the domain of Y. Formally, for the tuple, (x, y), the membership value of $[\mathrm{A} \uparrow \mathrm{X} \times \mathrm{Y}](x, y)$ is

$$\mu_{[\mathrm{A}\uparrow\mathrm{X}\times\mathrm{Y}]}(x,y) = [\mathrm{A} \uparrow \mathrm{X} \times \mathrm{Y}](\mathrm{x},\mathrm{y}) = \mu_A(x).$$

In the general case of the cylindrical extension of a fuzzy relation R, where $R \subseteq X_i \times \ldots \times X_j$, to $X_1 \times \ldots \times X_m$, where $j \geq i$ and $1 \leq i, j \leq m$, is

$$\mu_{[\mathrm{R}\uparrow X_1\times\ldots\times X_m]}(x_1, \ldots, x_m) = [\mathrm{R} \uparrow X_1 \times \ldots \times X_m](x_1, \ldots, x_m) = \mu_R\left(x_i, \ldots, x_j\right).$$

The projection of a relation R on $\mathrm{X} \times \mathrm{Y}$ to X, $[\mathrm{R} \downarrow \mathrm{X}]$, is a lossy transformation. For each element in X, the membership value is a single number that summarizes all tuples, (x, y), in $\mathrm{X} \times \mathrm{Y}$, where x is fixed but y varies. Formally, $[\mathrm{R} \downarrow \mathrm{X}](x)$ is

$$\mu_{[\mathrm{R}\downarrow\mathrm{X}]}(x) = [\mathrm{R} \downarrow \mathrm{X}](\mathrm{x}) = \ supremum_{y\in\mathrm{Y}}\mu_R(\mathrm{x},\mathrm{y}).$$

The last concept is that of cylindrical closure. For $R_1 \subseteq X_1, \ldots, R_k \subseteq X_k$, $Cyl(R_1, \ldots, R_k)$ is the intersection of all the cylindrical extensions, specified according to

$$Cyl(R_1, \ldots, R_k) = \cap_{i=1}^{k}[R_i \uparrow X_1 \times \ldots \times X_k],$$

The value at $(x_1, \ldots, x_k)$ in $Cyl(R_1, \ldots, R_k)$ is

$$Cyl(R_1, \ldots, R_k)(x_1, \ldots, x_k) = \wedge_{i=1}^{k}\{\,\mu_{R_i}(x_i)\}.$$

2.3.1 Fuzzy Logic - Introduction

The subject of higher-level analytical thought has been investigated by many great philosophical and mathematical thinkers and it is a classical topic in Artificial Intelligence (AI). The development of classical AI has very much been inspired by crisp logic, which might help explain the historical amount of symbolic manipulation and brittleness often found. Logic systems have seen great advancement from the early days of Aristotle's formal logic to more general and expressive forms of symbolic and mathematical logic. Fuzzy logic is a generalization of classical binary valued logic.

A summary of important contributors to logic include the Greek philosopher Aristotle and his syllogistic logic, George Boole and Boolean algebra, Gottlob Frege and predicate logic, Lukasiewicz and multi-valued logic, and more recently, Lotfi Zadeh for fuzzy logic. Fuzzy logic, founded by Zadeh in 1973 [34], is a powerful framework for approximate reasoning. Much of human reasoning is not precise. A classical example is the problem of stopping a car at an intersection. No one stops perfectly three feet before the intersection line exactly when the light turns red. One typically applies the breaks and slows down when the light turns yellow and stops close but before the intersection when the light is red. More to the point here, there is not a crisp instance of time where someone is standing upright or lying on the ground. Regardless of whether the problem is a simple day to day human task, a higher level cognitive task, or a problem from the physical sciences, uncertainty is unavoidable. The selection of a logic that can operate in these situations is desirable. An overview of fuzzy logic and a fuzzy inference engine are detailed next. The mathematical aspects of fuzzy logic are explained in the following sub-section.

Fuzzy logic inference system (FIS) operates on knowledge structured in an IF-THEN rule format. The IF part of a rule is called the antecedent, while the THEN part of a rule is called the consequent. Rules are constructed using linguistic variables. Once the knowledge, rule set, has been specified or learned, the general structure for applying all the knowledge to new input in order to draw a conclusion is done by a fuzzy inference engine, which is illustrated in figure 2.5.

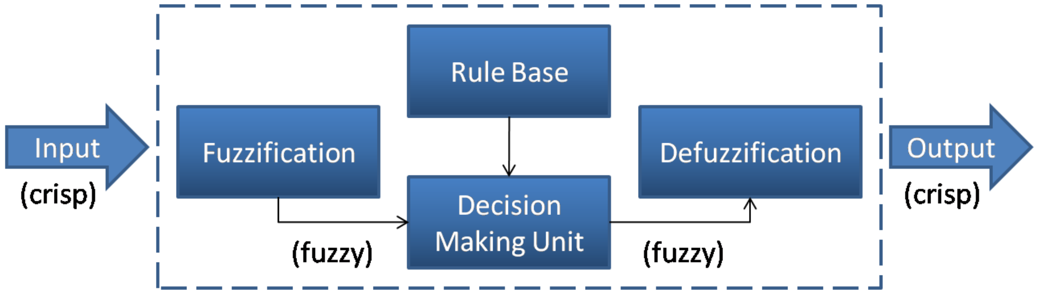


Fig. 2.5. Illustration of a fuzzy inference engine. Inputs, typically crisp, are fed into the system, crisp values are fuzzified, rules are fired using the rule base and the inference engine, and outputs are typically, but not always, defuzzified to obtain a crisp value that can be used by a system.

If the input to the system is crisp, then the first step is fuzzification. Fuzzification takes the inputs and converts them to fuzzy sets. However, in practice, most just determine the degree to which the input belongs to each of the appropriate fuzzy sets via their respective membership functions. Next, the decision making unit uses the fuzzy memberships and the rules from the rule base to draw its conclusions, which is typically followed by defuzzification. Defuzzification is the process of converting a fuzzy set into a single scalar, such as a single value to be used by a control system. This process is repeated for each input presented to the system. There are many variations on how these stages are calculated, such as how the antecedents are combined, the operators and process in the decision making unit, the selection of a defuzzification operator, etc. In this dissertation, the standard Mamdani-Assilion fuzzy inference system is used (illustrated in Fig 2.6) [34][44].

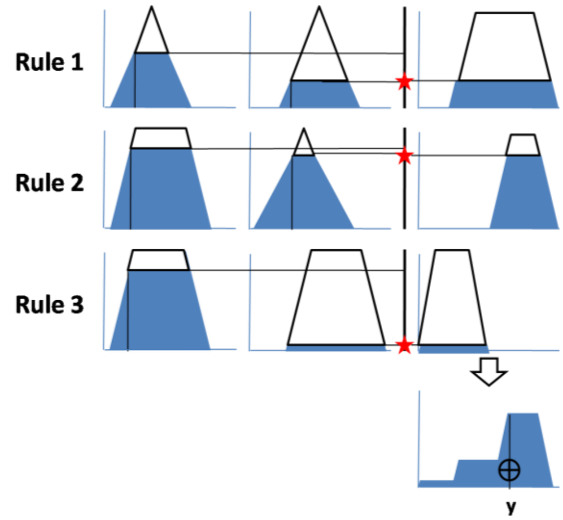


Fig. 2.6. Computational stages in a Mamdani-Assilion fuzzy inference system (FIS). Rules are fired by calculating the Min of all the fuzzified inputs, the Min of the respective consequent fuzzy set and the rule firing strength is computed, the consequents are aggregated, using a Max operation at each discrete output element, and the centroid, e.g. defuzzification, is calculated.

2.3.2 Fuzzy Logic – Implication and Rules of Inference

A rule of inference is a relationship detailing how conclusions (consequents) are drawn from a set of premises (antecedents). For example, the well known rule of inference modus tollens, $((P \Rightarrow Q) \wedge \neg Q) \Rightarrow \neg P$, Latin for "the way that denies by denying", is

if P then Q

$\neg\, Q$

Therefore, $\neg\, P$,

where ¬ is the not operator and P and Q are propositions, which in the classical non-fuzzy case is a statement about the affirmation or denial of something. Another more commonly used rule of inference, modus ponens, $((P \Rightarrow Q) \wedge \mathrm{P}) \Rightarrow Q$, Latin for “mode that affirms by affirming”, states

$$\text{if } P \text{ then } Q$$

$$P$$

$$\text{Therefore, } Q.$$

Fuzzy reasoning is based on the generalized modus ponens,

$$\text{if } (V\ is\ A) \text{ then } (W\ is\ B)$$

$$V\ is\ A',$$

$$\text{Therefore, } W\ is\ B',$$

where V and W are variables (in fuzzy logic, these are linguistic variables), which respectively have the constraints A and B (in fuzzy logic, these are terms), where $A \subseteq \mathrm{X}$ and $B \subseteq \mathrm{Y}$. In fuzzy logic, rules are modeled using fuzzy relations, $R \subseteq \mathrm{X} \times \mathrm{Y}$, and an implication operator is used to build R. Zadeh used the (Lukasiewicz) operator

$$\mu_{A \Rightarrow B}(x, y) = 1 \wedge (1 - \mu(x) + \mu(y)),$$

the Mamdani operator is

$$\mu_{A \Rightarrow B}(x, y) = \mu(x) \wedge \mu(y),$$

the product operator is

$$\mu_{A \Rightarrow B}(x, y) = \mu(x) * \mu(y).$$

The implication operator used in this dissertation is the Mamdani operator.

2.3.3 Fuzzy Logic – Compositional Rule of Inference

In fuzzy logic, inference is performed using Zadeh's compositional rule of inference (COI) [34]

$$[\,[A' \uparrow \mathrm{X} \times \mathrm{Y}] \circ R \downarrow \mathrm{Y}\,].$$

The value $\mu_{B'}(y)$ is calculated according to

$$\mu_{B'}(y) = supremum_{x \in \mathrm{X}}(\mu_R(\mathrm{x}, \mathrm{y}) \wedge \mu_{A'}(\mathrm{x})).$$

In the case of compound propositions, such as

if (V is A) AND (U is B) then (W is C)

(V is A') AND (U is B'),

Therefore, W is C',

the standard translation of "(V is A) AND (U is B)" is "$< V, U >$ is P", where

$$\mu_P \colon \mathrm{X}_1 \times \mathrm{X}_2 \to [0,1],$$

$$V \in \mathrm{X}_1, U \in \mathrm{X}_2, W \in \mathrm{Y},$$

$$\mu_P(x_1, x_2) = ([A \uparrow \mathrm{X}_1 \times \mathrm{X}_2] \wedge [B \uparrow \mathrm{X}_1 \times \mathrm{X}_2])(x_1, x_2) = \mu_A(x_1) \wedge \mu_B(x_2),$$

and the new relation is "$<< V, U >, W >$ is R", where, for the Lukasiewicz implication operator,

$$\mu_R(x_1, x_2, y) = \mu_{A \Rightarrow B}\left([A \times B](x_1, x_2), \mu_C(y)\right),$$

$$= 1 \wedge [1 - [A \times B](x_1, x_2) + \mu_C(y)],$$

$$= 1 \wedge \left[1 - \left(\mu_A(x_1) \wedge \mu_B(x_2)\right) + \mu_C(y)\right],$$

and the membership for each consequent domain element is now

$$\mu_{C'}(y) = supremum_{x_1 \in X_1}\left(supremum_{x_2 \in X_2}\left(\mu_R(x_1, x_2, y) \wedge \mu_{A'}(x_1) \wedge \mu_{B'}(x_2)\right)\right).$$

Figure 2.7 is a visualization of stages in the compositional rule of inference.

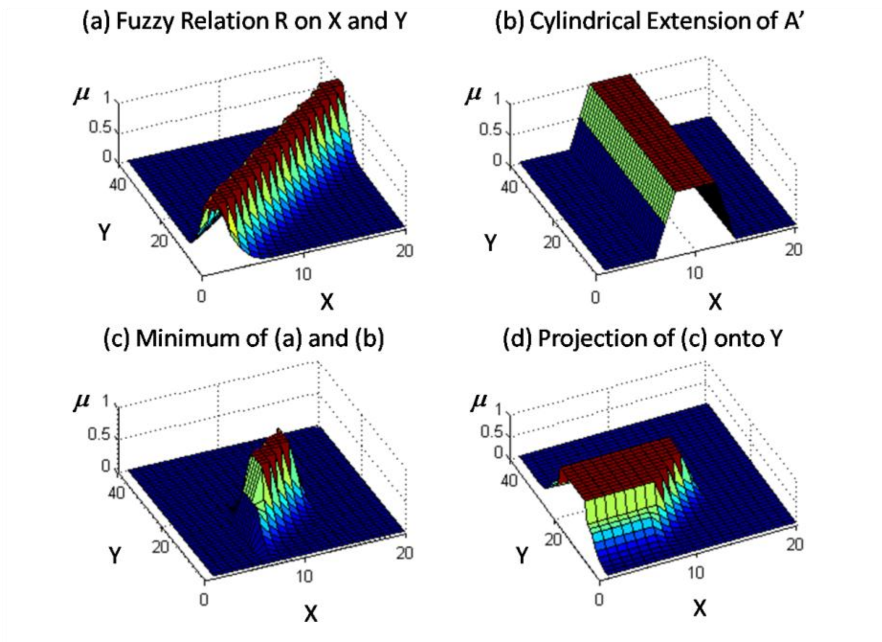


Fig. 2.7. Visualization of stages in the compositional rule of inference.

2.4.1 Change Detection and Human Segmentation from Video - Introduction

Some people, such as Martin, use non-video-based sensing technologies, such as toggle switches, vibration sensors, temperature sensors, pressure sensors, etc [45], others use active sensing modalities such as RFIDs [46], while others build wearable sensors, e.g. sensors placed in the soles of shoes [47]. Factors, such as how practical it is to outfit environments with certain types of sensors, assumptions that people might continuously wear active sensors, combined with the amount of information that can be extracted from such simple non-video-based technologies is limiting. While the context of this thesis is computer vision, non-video sensors are not considered to be of no value. It is this researcher's belief that long-term success will depend on engaging multiple types of sensors and fusing their results.

Video sensors are a rich source of information that can be used to monitor a scene. However, there are many challenges in implementing a computer vision system. The simple idea of "just subtract two images" to find the human does not work. Emphasis is placed here on algorithms for monitoring single people in indoor environments. This line of research has utility for the monitoring of independent single elders in their living quarters. If a goal is adverse event detection, e.g. fall recognition, a reasonable assumption is that if there are multiple individuals present then someone is there to assist if a fall occurs.

The algorithms reviewed below assume that if there are multiple cameras monitoring the same scene, each camera is carrying out human segmentation independently. Typically, multi-person tracking is based on the idea of apply the same initial change detection procedures presented below along with post-processing algorithms for the identification, labeling and tracking of multiple people in light of matters such as occlusion [48][49]. The ongoing stereovision-based human segmentation research of Luke and Anderson [14] is an extension of the low and medium level computer vision components contained in this thesis. That work specifically represents one possible way to extend this thesis to address multiple person tracking. Lastly, the work of Han et al. provides an alternative way to address people finding by bypassing change detection and using histogram of gradients (HOG) and local binary pattern (LBP) features to directly detect multiple people in single camera systems [50].

The standard approach to change detection and person recognition is the theory of background models [27][14][23][51][24]. A background model is everything non-human. Change detection is the process of detecting regions of sufficient difference from the background model. However, due to factors such as illumination changes, shadows, moving objects, not all "changed" regions are human. Person recognition is the process of detecting which change detection regions are human.

It is worth mentioning here that most current background-based approaches are just change detection algorithms, meaning there is nothing specific regarding the identification of humans. Some have worked to extend the initial change detection results to consider distinguishing features of humans, such as skin tone and head tracking in volume element (voxel) space [52], as well as human face detection and height fields for differentiating objects based on their height profile at various known or modeled locations using stereo vision [53].

2.4.2 Change Detection and Human Segmentation from Video – Image Space Change Detection

A raster image at time t, I_t, is a two dimensional collection of picture elements (pixels). Typical images are derived from the visible band of the electromagnetic spectrum and are represented as grayscale (monochromatic), $I_t(x, y) \in [0,1]$, where (x, y) are row and column indices respectively, or multi-chromatic, e.g. red, green, and blue (RGB), $\vec{I}_t(x, y) =< r, g, b >^T$, YCbCr, HSV, or CIELab.

The first stage of human segmentation is change detection. That is, the discovery of pixels that have *sufficient* difference from the current background model B_t (the background model is the subject of analysis in following sections),

$$C_t = \{((x, y), \vec{I}_t(x, y)) \big| \, \vec{I}_t(x, y) \in I_t, d\left(\vec{I}_t(x, y), \vec{B}_t(x, y)\right) > \tau\},$$

where $\vec{B}_t(x, y)$ is a background model pixel, $d\left(\vec{I}_t(x, y), \vec{B}_t(x, y)\right)$ is a distance function, and τ is a user or algorithm identified difference threshold. There is not a single change detection algorithm for all problems. While the method of background-based differencing is common, the features used and criteria for determining change vary per domain and problem.

A significant concern is always the matter of privacy. Very few people are comfortable with "big brother" watching. Acceptance of vision-based technologies depends in part on who has what viewing privileges. To preserve the privacy of subjects in this work, segmentation of images results in silhouettes, where S_t is the silhouette at time t. In operation, silhouettes are stored, never the original images. A silhouette is a binary image, $S_t(x, y) \in \{0,1\}$, in which the majority of distinguishing characteristics of people is removed. Of particular interest to human change detection is the (foreground) set S_t^F,

$$S_t^F = \{((x, y), \vec{I}_t(x, y)) \big| \, \vec{I}_t(x, y) \in I_t, S_t(x, y) = 1\}.$$

2.4.3 Change Detection and Human Segmentation from Video – Mixture of Gaussians

Stauffer and Grimson introduced an adaptive method for background modeling and subtraction that utilizes a mixture of Gaussians (GMM) per pixel with a real-time, online approximation to the model update [24]. Each pixel is modeled using K Gaussian distributions. For each pixel, the probability of observing the current pixel value is

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

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

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

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

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

The user specifies a threshold T, $0 \leq T \leq 1$, that describes the portion of the data that should be accounted for by the background. Therefore, $B_{(x,y)}$, where $1 \leq B_{(x,y)} \leq K$, mixtures account for the background. The authors, for computational reasons, assume that $\Sigma_{j,(x,y)} = \sigma_{j,(x,y)}^2 I$, where I is the identity matrix. Before selecting the set of mixtures that make up the background, the models are sorted according to $\omega_{j,(x,y)}/\sigma_{j,(x,y)}$, or in the case of a multidimensional $\vec{\sigma}_{j,(x,y)}^2$, $\omega_{j,(x,y)}/\left\|\vec{\sigma}_{j,(x,y)}\right\|$ [54]. This value increases both as the variance decreases and the mixtures weight increases. The number of background mixtures, $B_{(x,y)}$, is therefore

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

where $\omega_{(k),(x,y)}$ are the sorted, in increasing order, weights. If the current pixel lies in 2.5 standard deviations of any background mean, then a matching model is found and the location is assumed to be background. In this case, the mean, standard deviation, and mixtures are updated as follows. The value t, i.e. $\omega_{j,(x,y),t}$, indicates the current time step and t-1, i.e. $\omega_{j,(x,y),t-1}$, indicates the prior time step.

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

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

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

$$\rho = \alpha\eta(\vec{I}_t(x,y) | \vec{\mu}_{j,(x,y),t-1}, \Sigma_{j,(x,y),t-1}),$$

where α, $0 \leq \alpha \leq 1$, is the learning rate (user specified), $\omega_{j,(x,y),t}$ is the jth mixture at time t, $\omega_{j,(x,y),t-1}$ is the jth mixture at the previous time step, $M_{j,(x,y),t}$ is a function that is 1 for the matched mixture, and 0 otherwise. If the current pixel is not within 2.5 standard deviations of a distribution, then the weakest model, according to $\omega_{j,(x,y)}/\|\vec{\sigma}_{j,(x,y)}\|$, is identified and replaced. The weakest model is replaced with $\vec{\mu}_{j,(x,y),t} = \vec{I}_t(x,y)$, and $\vec{\sigma}^2_{j,(x,y),t}$ and $\omega_{j,(x,y),t}$ are set to user defined initial values. In order to keep the constraint on $\omega_{j,(x,y),t}$ valid, the sets of weights need to be normalized by the sum of the current set.

This algorithm has difficulty with abrupt changes in lighting and shadows. Also, separate techniques must be adopted in order to not adapt humans into the background.

2.4.4 Change Detection and Human Segmentation from Video – Eigenbackgrounds

Oliver et al. carry out foreground segmentation in eigenspace, where the background is modeled as an eigenbackground [23]. The eigenbackground is built using N background images. Rows from $I_t(x,y)$ are concatenated to build one large feature vector, $\vec{v}_t$, that is $(R * C)$x1 in size, where R is the number of rows and C is the number of columns in I_t, hence

$$\vec{v}_t = \begin{pmatrix} I_t(0,0) \\ \cdots \\ I_t(R-1,0) \\ \cdots \\ I_t(R-1,C-1) \end{pmatrix}.$$

The mean of the N background image vectors is

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

Next, a matrix whose columns are the differences between the background image vectors and the mean, which is $(R * C)$x N in size, is constructed,

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

Eigenbackgrounds use the Karhunen-Loeve Transform, also known as Principal Component Analysis (PCA), a linear feature reduction procedure for generating mutually uncorrelated features for matters such as avoiding information redundancy and addressing the curse of dimensionality. First,

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

is computed (the covariance matrix of A, without the $\frac{1}{n-1}$ scaling factor), whose size is $(R * C)$x$(R * C)$. So, for a 640x480 image this is 307200x307200, which is extremely large. The next step is to compute the eigen information, which is very computationally expensive for a $(R * C)$x$(R * C)$ matrix. A trick is to instead make the matrix of size NxN [55],

$$L = A^T A.$$

The matrix L is subject to eigen analysis, where Φ is the matrix of eigenvectors, whose vectors are arranged in column format, and Λ is the matrix of eigenvalues, where the eigenvalues are aligned along the diagonal and the matrix is zero at off diagonal elements. The covariance of the data set can be decomposed as follows

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

The trick in using the L matrix is that [56]

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

where $\overrightarrow{\Phi}_i$ is the ith eigenvector (column vector) of $A^T A$. Left multiplying A for both sides gives

$$(AA^T)(A\overrightarrow{\Phi}_i) = (\Lambda_{i,i})(A\overrightarrow{\Phi}_i).$$

It is concluded that $\Lambda_{i,i}$ is an eigenvalue of $\mathrm{Cov} = AA^T$, with the corresponding eigenvector $A\overrightarrow{\Phi}_i$. A user picks $N^{'} \ll (R * C)$ eigenbackgrounds and the $(R * C)$x1 size eigenvectors are computed according to $A\overrightarrow{\Phi}_i$. The matrix V, of size $(R * C)$x$N^{'}$, is calculated,

$$V = [A\vec{\Phi}_1, \dots, A\vec{\Phi}_{N'}].$$

Once a new image is presented to the system, $\vec{v}$, it is projected into eigenbackground space by

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

The vector $\vec{\Omega}$ describes the contribution of each eigenbackground in representing the input image, where the eigenbackgrounds comprise a basis set for images. The reconstruction of the background from the eigenbackground, of size $(R * C)$x1, is

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

The distance between the background and its reconstruction is

$$\vec{\zeta}(x, y) = |(\vec{v}(x, y) - \vec{\mu}(x, y)) - \vec{s}(x, y)|.$$

Pixels that have $\vec{\zeta}(x, y) > \tau$, where τ is a user specified threshold, are detected change. In [57], Wang et. al presents an incremental update method for subspace learning. Eigenbackground's are sub-adequate for the same reasons mentioned in the GMM section (rate of background adaptation, change is not guaranteed to be human, i.e. shadows, illumination, moving objects, etc).

2.4.5 Change Detection and Human Segmentation from Video – Graph Cuts

Many difficulties, such as noise from CCD cameras, insufficient contrast between foreground and background, shadows, illumination change, moving objects, etc, force a system to perform post-processing on initial change detection results to reduce noise and obtain region segmentations. General approaches include mathematical morphology (i.e. closing, opening, reconstruction) and region filling. However, these methods generally result in silhouette deformation, typically along the contour. A more intelligent approach is based on the theory of minimum graph cuts [58][59] and domain heuristics. Graph cuts have been used for image clustering [60] and more recently for the correspondence problem in stereo vision [61]. Other general related segmentation procedures include normalized graph cuts [62] and Markov random fields [63]. Greig et al. initially showed how minimum cut/maximum flow can be used to minimize certain energy functions in the area of computer vision [64], particularly those of the general form

$$E(L) = \sum_{p \in P} D_p(L_p) + \sum_{(p,q) \in \mathrm{N}} V_{(p,q)}(L_p, L_q),$$

where $L = \{L_p | p \in P\}$ is a labeling of image P, $D_p(\cdot)$ is a data penalty function, $V_{(p,q)}$ is an interaction potential, and N is a set of all pairs of neighboring pixels.

Specifically, a graph is built based on the image. Each pixel has a corresponding graph vertex. Two additional vertices are included from the two labels, foreground and background. Each pixel vertex is connected to six or ten other vertices. For a four-connected neighborhood, there are four connections plus two to the labels. As mentioned above, the neighborhood connections are created in order to help enforce spatial coherency of the labels assigned to pixels. The label link weights generally depend on the difference between the current image and the background, and/or the foreground. The neighbor link weights between pixel vertices could all be assigned identical values or could vary over an image based on local information, such as neighborhood contrast [65]. The goal is the partitioning of the resulting graph into two disjoint sets of vertices based on the calculated weights [59]. Well known minimum cut/maximum flow algorithms include the Goldberg-Tarjan style “push-relabel” or Ford-Fulkerson style “augmenting paths” [59].

In [65], $D_p(\cdot)$ is defined as the similarity between a pixels background and foreground. Background similarity is based on an alpha blend between global and local color models. The authors rationalize this by mentioning that the mixture model on the whole image is more robust than the local model but the local model can better handle pixel jitter, local illumination changes, and small movements in the background. The global color model, $p(I_r|L_p = 0)$, where I_r is the rth pixel, is a mixture model over the entire image, typically 10 to 15 mixtures, and it is acquired using a single known background frame. The local color model, $p_B(I_r)$, is a single Gaussian acquired per pixel using a known background sequence. The authors learn a foreground color model, $p(I_r|L_p = 1)$, typically 5 mixtures, by analyzing different images using the acquired per-pixel background models, $p_B(I_r)$, and two thresholds (τ_b and τ_f). If $p_B(I_r) > \tau_b$, the pixel is called background. If $p_B(I_r) < \tau_f$ the pixel is called foreground, else it is labeled uncertain. The set of foreground pixels are used to train the foreground mixture model. Their specific $D_p(L_p)$ is

$$D_p(L_p) = \begin{cases} -\log(\alpha * p(I_r|L_p = 0) + (1-\alpha) * p_B(L_r)) & L_r = 0 \\ -\log p(I_r|L_p = 1) & L_r = 1 \end{cases},$$

and $V_{(p,q)}(L_p, L_q)$ is

$$V_{(p,q)}(L_p, L_q) = |L_p - L_q| * \exp(-\beta * d_{(p,q)}),$$

where $d_{(p,q)}$ is the L_2 norm of the color difference. The authors also introduce a method called adaptive background contrast attenuation for improving $V_{(p,q)}(L_p, L_q)$. A method for updating α based on the Kullback-Liebler divergence between the global and local GMMs.

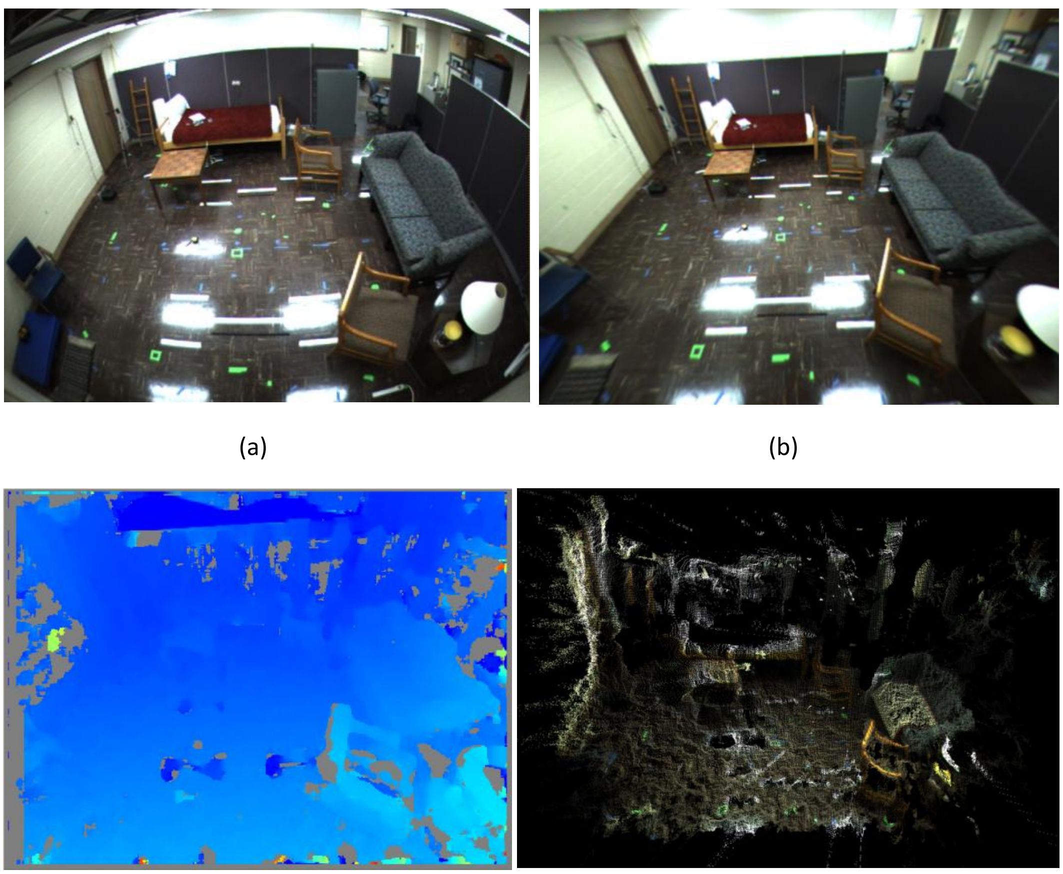
The authors assume a rather constrained environment. Their problem domain is video conferencing. It is assumed that there are subtle illumination changes because of auto gain/white-balance control of a camera and illumination from environment light sources, such as a fluorescent lamp. They use histogram specification to adjust the background image globally. They use the labeled background regions in the current image and the background model. They acquire and use a histogram transformation function to update the entire background image at each frame. If they ever detect a large sudden illumination change, via frame differences and a threshold, then they begin a process of re-acquiring the background model using alpha blending for gradual model parameter update. The background color model is updated as an alpha blend of the old background and the new image. The variance is also just an alpha blend of the old variance and the change between the current image and the background model. This procedure is simple and it does not take into account many of the difficulties in model adaptation due to large global change and objects in the scene during model update.

They address movement in the background in two ways. First, if an object has significant color difference from the background, then in many cases their proposed method will gradually automatically adapt and rely on the global background model, by reducing the effect of the per-pixel background models. If the object is still detected and there is no intersection between the moving object and the foreground, then connected components is used and large islands are kept (they assume they are tracking a single person). Lastly, they attempt to measure 'casual' camera shaking (< 4 pixels), the specifics are not discussed, and if it is a small shake then they blur the local background model and leave the global background model alone. If the translation is large, then they disable the per-pixel color model. Overall, this approach interestingly incorporates multiple low (pixel level differencing from the background), medium (region extraction) and high level (scene event detection and handling) computer vision techniques for indoor people tracking.

2.4.6 Change Detection and Human Segmentation from Video – Stereo Vision

A superior, in comparison to silhouette extraction carried out in image space, but far less researched approach to human change detection involves stereo vision and world space [66][67][68][69][70]. Stereo vision is a method of determining three dimensional depth from multiple two dimensional cameras. It is this researcher's belief that many people avoid stereo vision due to misconceptions about real-time processing, cost of equipment, or one is just unaware of the approach due to lack of advertisement in the area of human-based change detection. Theoretical advances in stereo vision, e.g. [68][61], and modern technological advancements in hardware has turned stereo vision into a reality. For example, affordable commercial solutions such as Point Grey's bumblebee system [71] now exist. Granted, the cost of a stereo vision solution is still considerably greater than that of single camera technology, but the research benefit in relation to price is now justifiable.

Stereo vision is not invariant to changes in illumination or shadows, however it theoretically provides a better foundation to begin to address these topics that plague silhouette extraction in image space. Neither illumination nor shadows change the position of an object in stereo vision, just its visual appearance. Figure 2.8 illustrates the different stages in stereo vision.



(a) (b)

(c) (d)

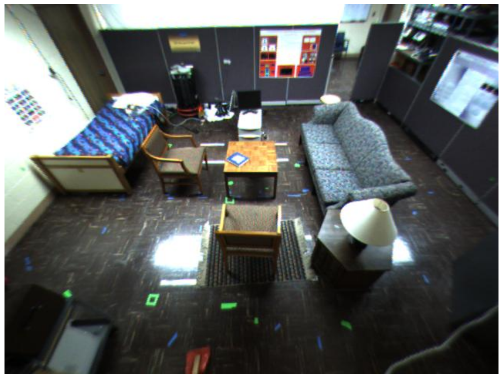
Fig. 2.8. Stereo vision example. Image (a) is the raw camera output, (b) is the rectified image, (c) is the disparity map, and (d) is the three dimensional point cloud textured according to their image colors. Data was collected using Point Grey's bumblebee system.

Given a pair of stereo images, the first step is rectification. Rectification determines a transformation of each image plane such that pairs of epipolar lines become collinear and parallel to one of the image axes. Next, pixel correspondences are found, resulting in disparity, and depth can then be calculated using the disparity and the position of points are found using trigonometry [68]. Refer to [68] for more information regarding occlusion detection in stereo vision.

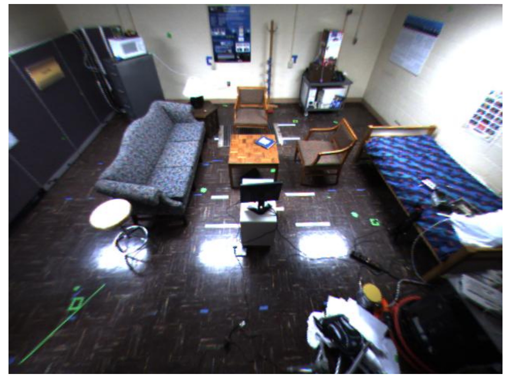
The most direct idea for incorporating stereo vision into change detection is that once the depth map is acquired, GMM, Eigenbackgrounds, or any other change detection and adaptive background update algorithm can be used directly on the depth values [72]. The values acquired from a depth map have a greater meaning as they relate to object position and subsequently movement, as opposed to a value indicating color and/or texture change.

In [73], Muñoz-Salinas et al. describe a slightly more sophisticated change detection procedure based on a single stereo camera pair and point clouds. The authors construct a height map corresponding to a quantization of the x-y ground plane. Essentially, their approach is just a quadtree (in the x-y plane) that stores the maximum observed z (world up direction) height per cell. They have a simple change detection procedure that takes into account height map value changes over time and subsequently they search for clusters of points, i.e. objects. They then use skin color and other information, such as any faces present, to build confidence in an object being a human. This system is not restricted to single person tracking. However, the approach is limited in the following regard. A height map is an overly simplified representation of a complex three dimensional environment. Additionally, their height map-based change detection procedure is unsophisticated. It will not be able to sufficiently address a complex realistic environment with moving non-human objects. The correspondence problem in stereo vision is not solved. It is unclear how well their height map representation will hold up in light of complex environments with many erroneous matches. Lastly, face detection might not always be available given the monitoring conditions and resolution of people's faces and viewing conditions in a scene.

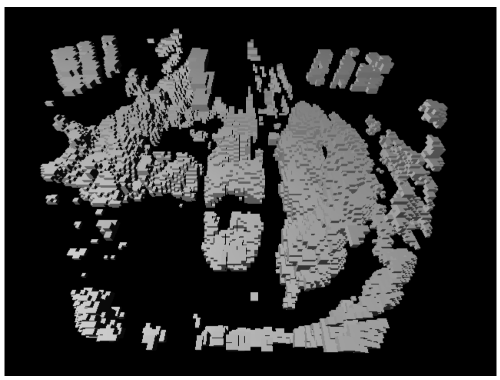
A newer improved approach is that of Luke et al. [14]. In this work, a quantized three-dimensional volume element (voxel) space is used and the output of multiple stereo vision camera pair systems is combined. Each stereo vision system builds a three dimensional voxel environment based on back-projecting the depth information. The individual stereo vision system voxel environments are combined using intersection and the space is post-processed to reduce noise and automatically segment voxel islands. The connected components algorithm is used to find voxel regions attached to each other, which are called islands. Thus, a rich full three dimensional voxel scene is built in real-time using voxel-pixel lists, which are computed offline. A change detection procedure in voxel space is introduced, which results in voxel island identification. Small islands are removed. From the remaining set of islands, a classifier, based on skin color, height, and head shape is used to identify only human voxel islands. In the case that the classifier fails for a given frame, this system resorts to color histogram matching of prior matched humans to current change detection islands with adequate volume to warrant potential classification of a human. The authors demonstrate superiority of this system, over the prior mentioned change detection techniques, in term of foreground and background detection with respect to significant scene change, i.e. major and minor illumination changes, shadows, and moving objects. In summary, this stereo vision approach (1) is a real-time system, (2) is an improved way to address illumination changes and shadows intrinsically using stereo vision, (3) includes a more sophisticated classifier for detecting a human, (4) significantly reduces non-human object identification, (5) has a more mature background model representation and update, and (6) is modular in design such that performance is expected to increase as each component, i.e. correspondence, human skin detection, etc, is individually advanced. Figure 2.9, courtesy of Luke [69], illustrates important steps in this change detection algorithm.



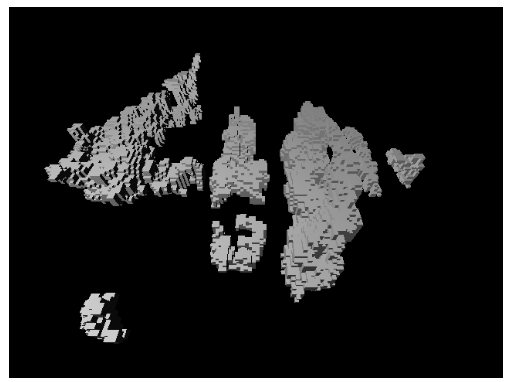
(a)



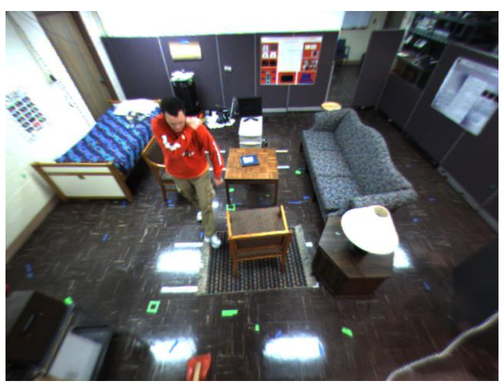
(b)



(c)

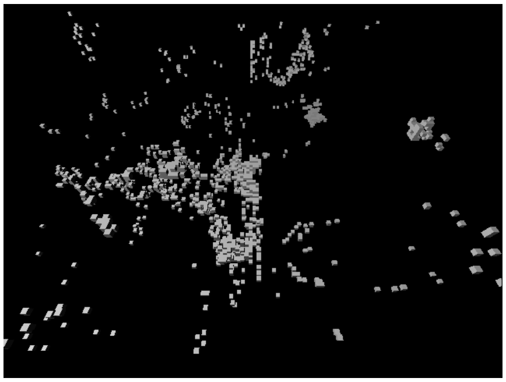


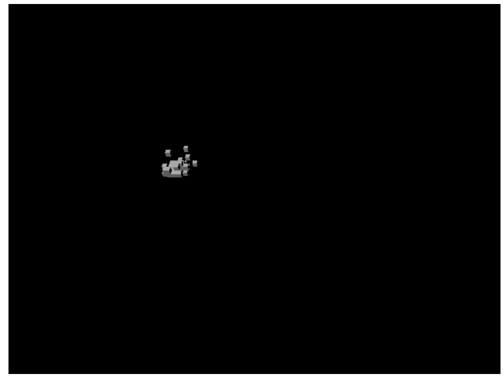
(d)



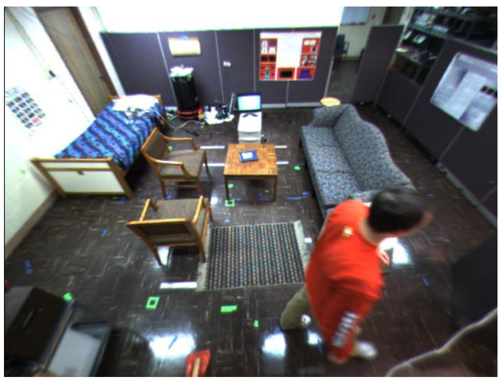
(e)

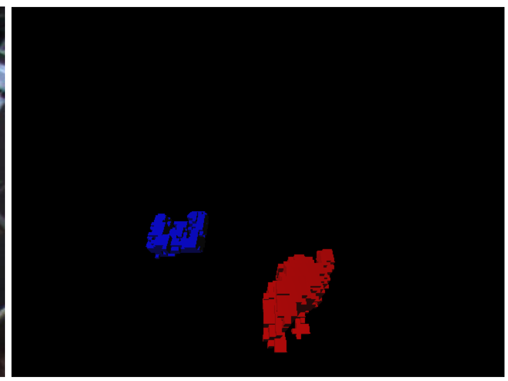
(f)





(g) (h)





(i) (j)

Fig. 2.9. Stereo vision-based change detection in voxel space. Images (a) and (b) show the camera viewing conditions, (c) is the initial stereo-based back-projection results, (d) is the refined voxel space, (e) is a new image containing the human, (f) is detected skin, (g) is skin voxels, (h) is head shape detection intersected with the set of skin voxels, (i) is a frame in which an object, the chair lowest in the image y (height) dimension, was recently moved and multiple initial change detection voxel islands are present, and (j) is the systems classification of the human (red) versus non-human objects (blue).

2.5.1 Graphical Models - Introduction

The most widely accepted and utilized approach to modeling and inferring human activity is probabilistic graphical models[25], which includes dynamic Bayesian networks [74], also known as dynamic graphical models, HMMs [19][23], and HMM variants (hierarchical HMMs [20], Entropic-HMMs [26], coupled-HMMs [23][32], etc). Graphical models, or more specifically, probabilistic graphical models, the probabilistic term will be omitted from this point on, are a union of probability and graph theory. A graphical model is a particular factorization of a joint distribution. In this graph, vertices are random variables and edges (lack of) represent conditional dependencies (independencies).

Probability, statistical independence, and random variables need to be discussed first [75]. Probability theory, which is built on top of classical crisp set theory, is a useful tool for mathematically representing a specific type of uncertainty, specifically, relative frequency. When an experiment is performed, one of several possible outcomes is observed. The set of all possible outcomes is the sample space, Ω. An event, E, is a subset of the sample space, $E \subseteq \Omega$. A probability measure is the mapping of events to $[0,1]$,

$$P: 2^{\Omega} \to [0,1].$$

A probability function P must satisfy the following three axioms.

(1) Non-negativity

$$\forall E, P(E) \geq 0.$$

(2) Boundary conditions

$$\mathrm{P}(\emptyset) = 0 \text{ and } \mathrm{P}(\Omega) = 1.$$

(3) Additivity

$P(E \cup F) = P(E) + P(F)$, for disjoint events E and F

or, more generally

$P(\bigcup_{i=1}^{K} E_i) = \sum_{i=1}^{K} P(E_i)$ for disjoint events $E_i, \ldots, E_K$.

When $P(F) > 0$, the conditional probability, $P(E|F)$, which is the probability that event E occurs, given that F was observed, is

$$P(E|F) = \frac{P(E \cap F)}{P(F)}.$$

The product rule, the probability that both events E and F occur, which is directly derived using $P(E|F)$, is used in graphical models, typically along with independence assumptions, and is given by

$$P(E \cap F) = P(F)P(E|F).$$

The chain rule,

$$P\left(\bigcap_{i=1}^{K} E_i\right) = P(E_1)P(E_2|E_1)P(E_3|E_1 \cap E_2) \ldots P\left(E_K \middle| \bigcap_{i=1}^{K-1} E_i\right),$$

generalizes the product rule to K events. Events E and F are said to be independent iff

$$P(E \cap F) = P(F)P(E).$$

When independent events are analyzed in the context of conditional probabilities,

$$P(E|F) = \frac{P(E \cap F)}{P(F)} = \frac{P(E)P(F)}{P(F)} = P(E).$$

The result illustrates that knowledge about the occurrence of F does nothing to effect the calculation of the probability of occurrence of E. More generally, a collection of events, $E_i, \ldots, E_K$, are mutually independent if

$$P\left(\bigcap_{i=1}^{K} E_i\right) = \prod_{i=1}^{K} P(E_i),$$

while two events E_i and E_j, for $i \neq j$, are pair-wise independent if

$$P(E_i \cap E_j) = P(E_i)P(E_j).$$

Events E and F are said to be conditionally independent given event G if

$$P(E|G) = P(E|F \cap G),$$

or

$$P(E \cap F|G) = P(E|G)P(F|G).$$

If events E and F are conditionally independent, then once G is learned, learning F gives no additional information about E. Conditional independence is a key component in graphical models as it relates to simplification of a model, learning of a model, and efficient inference.

The last important concept needed from probability theory for graphical models is that of a random variable. In many experiments it is easier to deal with a summary variable than with the original probability structure. A random variable X is a function from Ω to some set Ψ,

$$X: \Omega \rightarrow \Psi.$$

Typically, the random variable X is thought of as a measurement of interest in the context of the random experiment. A random variable X is random in the sense that its value depends on the outcome of the experiment, which cannot be predicted with certainty before the experiment is run. Each time the experiment is run, an outcome (s) is observed, and a given random variable X takes on the value $X(s)$. A probability function is defined on the variable X. The probability function on the random variable X is

$$P_X(X = x) = P(\{\, s \in \Omega : \mathrm{X}(s) = x \,\}),$$

which performs the mapping

$$P_X: \Psi \rightarrow [0,1].$$

Depending on the particular problem, the variable can be a discrete or continuous random variable. If the domain of the random variable is finite or countably infinite, as in the set of integers, I, then it is called discrete. Otherwise, if the domain is uncountably infinite, such as the case of $\mathbb{R}$, then it is called a continuous random variable.

The second component in graphical models is graph theory. A good overview of important properties and theorems in graph theory can be found in [76]. Only concepts vital to understanding the aspects of graph theory as it relates to graphical models will be presented in this sub-section. A graph, G, is specified according to

$$G = (V, A),$$

where V is the set of vertices, $V = \{v_0, \ldots, v_N\}$, and A is the set of edges, $A = \{a_1, \ldots, a_K\}$. In a graphical model, a vertex is a random variable. In a directed graph, the edge a_k denotes that v_i is connected to v_j, i.e. $v_i \rightarrow v_j$. In an undirected graph, the edge a_k denotes that v_i is connected to v_j and v_j is connected to v_i, i.e. $v_i \leftrightarrow v_j$. Figure 2.10 illustrates directed and undirected graphs.

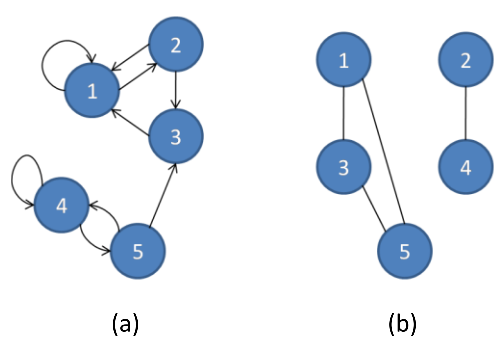


Fig 2.10. (a) Directed versus (b) undirected graphs.

A complete graph is one such that, for every possible pair of vertices, an edge exists in G,

$$\forall(v_i, v_j), \text{ where } 1 \le i \le N \text{ and } 1 \le j \le N, \exists\, a_u \text{ in } A \text{ such that } a_u = v_i \to v_j,$$

and the number of edges in a complete graph is

$$\frac{N(N-1)}{2},$$

undirected edges, or $N(N-1)$ directed edges. The size of a graph is important, as it relates to computational complexity of various graph algorithms and storage of the graph, typically as an adjacency matrix or adjacency link list. A path of length M is an ordered set of edges, e.g. $\{a_c, \dots, a_z\}$, whose cardinality is $|\{a_c, \dots, a_z\}| = M$, or alternatively, it is represented as an ordered set of vertices, $\{v_i, \dots, v_j\}$, where $\left|\{v_i, \dots, v_j\}\right| = M + 1$ with the following property: the bth path, p_b, starts at vertex v_i and end in vertex v_j. A graph has a cycle if there exists a path such that the path begins in one vertex and revisits the same vertex through a set of edge transitions,

$$\exists\, p_b \text{ in } G \text{ such that } v_i = v_j.$$

A directed acyclic graph (DAG) is a directed graph G that does not contain a cycle. The last necessary concept is that of a clique. A clique is a set of vertices, V', such that, for every pair of vertices, (v_i', v_j'), where $i \neq j$, there is an edge, i.e. $v_i' \leftrightarrow v_j'$. A maximal clique is a clique that is not a subset of another clique. Figure 2.11 illustrates the concept of a clique.

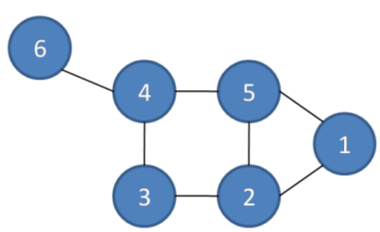


Fig. 2.11. Illustration of cliques in a graph. Vertices 1, 2, and 5 make up the largest clique, which is of size 3. All other pairs of vertices make up cliques of size 2.

The matter of representation in graphical models primarily concerns how a joint distribution is factored into either a directed or undirected graph. The first family of graphical models to be discussed is directed graphs, which includes Bayesian networks and Hidden Markov Models (HMMs). Directed graphs are useful for expressing causal relationships between random variables. Each vertex in a graph is a random variable and an edge $v_i \rightarrow v_j$ expresses probabilistic relationships between variables. The absence of edges implies conditional independence relations. In a directed graph, the edge $v_i \rightarrow v_j$ is typically interpreted to mean v_i causes v_j, or in the following generative sub-section, $v_i \rightarrow v_j$ is only interpreted to mean that v_i generates v_j (a weaker assumption). For example, consider the joint distribution $P(E, F, G)$, which can be factorized using the chain rule into

$$P(E, F, G) = P(G|E, F)P(E, F) = P(G|E, F)P(F|E)P(E).$$

This joint distribution is represented as the directed graphical model shown in Figure 2.12.

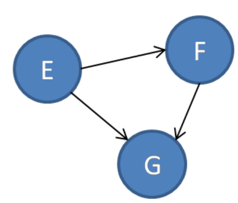


Fig. 2.12. Factorization of the joint distribution $P(E, F, G)$ into a directed graph.

Given this particular factorization of the joint distribution, the graphical models analyzed here are DAGs. There are many ways to factorize a joint distribution, and consequently there are multiple Bayesian networks consistent with a particular joint distribution. For example, when applying the chain rule to $P(E, F, G) = P(G|E, F)P(F|E)P(E)$, the left hand side of the equation is symmetrical with respect to the three variables, where as the right hand side is not. That is, for a single joint distribution, depending on the ordering of the variables, different directed graphical models can be obtained.

The joint distribution defined by a graph is given by the product, over all the nodes in the graph, of a conditional distribution for each node conditioned on the variables corresponding to the parents of that node in the graph. Therefore, for a graph with K vertices,

$$P(E_i, \ldots, E_K) = \prod_{i=1}^{K} P(E_i | pa_i),$$

where pa_i is the set of parents of E_i. For an edge $v_i \rightarrow v_j$, v_i is called the parent and v_j is the child. DAGs specify a factorization of the joint distribution over a set of variables into a product of local conditional distributions. A vertex is independent of its ancestors given its parents, where the ancestor/parent relationship is typically with respect to some fixed topological ordering of vertices. This means that if given the joint distribution $P(E, F, G, H)$, the chain rule is used to obtain

$$P(E, F, G, H) = P(H|E, F, G)P(G|E, F)P(F|E)P(E),$$

however, conditional independence assumptions allows it to be reformulated as

$$P(E, F, G, H) = P(H|F, G)P(G|E)P(F|E)P(E).$$

Conditional independence allows for a more compact representation of the joint distribution. In general, for discrete random variables, if there are N vertices, the full joint would require $\mathrm{O}(T^N)$ parameters, where T is the maximum number of discrete values for the variables, while the factored form only requires $\mathrm{O}(NT^k)$, where k is the maximum number of parents of a vertex [77]. The inference task in a graphical model is typically addressed using message passing algorithms such as belief propagation or the junction tree algorithm, while learning is generally addressed through parameter estimation using maximum likelihood or Expectation Maximization [74].

The next topic involves conditional probability distributions (CPDs). For each vertex, if the problem is one of discrete random variables, then a conditional probability table (CPT) needs to be specified. A CPT lists the probability that the child node takes on each of its different values for each combination of values of its parents. If a vertex has no parents, then the table is a set of probabilities for the elements of that particular random variable, i.e. $E(e_1), \ldots, E(e_P)$. If a vertex has a parent, then the CPT is a table whose rows are tuples of elements from the parents and the columns are conditional probabilities, where each entry is an element from the specific vertices random variable, conditioned on the parents, e.g., for a vertex G with parents E and F, the entries are $G(g_i|e_j, f_k)$. If the random variables are continuous, then there are no CPTs but rather CPDs, with the same respective organization as just discussed.

In addition to the type of graph, such as a directed graph, there is the topic of generative and discriminative models. For example, an HMM is a generative directed graphical model and a Support Vector Machines is a discriminate model. In generative models, one typically discusses a partitioning of the random variables into the sets X (observed), variables for which the value is known (also known as the evidence), and Y (unobserved, latent, or hidden), i.e. $X \cup Y = V$. Therefore, a model is the joint distribution $P(V) = P(Y, X)$. Given the set of observed variables, these graphical models answer queries about hidden variables through probabilistic inference. In other words, it computes $P(Y|X)$. Generative models include a model of $P(X)$. Discriminate models are based directly on the conditional probability $P(Y|X)$ and there is no need for modeling $P(X)$. In the generative approach, the joint probability is modeled using $P_g(Y, X; \theta)$, where θ is a set of parameters to typically be learned, and the assumption is that

$$P_g(Y, X; \theta) = P_g(X; \theta) P_g(Y|X; \theta),$$

while in the discriminative approach, one directly models

$$P_c(Y|X; \theta),$$

and the assumption is that

$$P_c(Y, X) = P_c(X; \theta')P_c(Y|X; \theta),$$

where θ' is different from θ. In discriminative modeling there is more freedom to fit the data because the goal is only to maximize the conditional likelihood with no assumption on the parameters of $P(X)$.

In addition, an HMM is a particular type of Bayesian network called a dynamic Bayesian network (DBN) [74][77]. DBNs are useful tools for time series or dynamic system analysis. A DBN is essentially the "unrolling" of a Bayesian network for T time steps (or T "slices"), where T is the length of the observation sequence. HMMs, as originally presented by Rabiner [78], uses the Baum-Welch algorithm for the task of learning (an instance of Expectation-Maximization), and the model likelihood is calculated using the forward-backward procedure (a dynamic programming algorithm). In this sub-section, the DBN format is adopted in order to maintain consistency with the related discussion on graphical models. Differences between classical HMMs (Rabiner) and a DBN representation include: classical HMMs represent the state of the world using as single discrete random variable while DBNs represent the state of the world using a set of random variables, and a DBN represents the state transitions, $P(X_t|X_{t-1})$, in a compact way using a parameterized graph. Murphy showed that a DBN representation for an HMM helps reduce computational complexity of inference and better addresses the general need for less training data to learn the model, related to the number of free model parameters [74]. Specifically, the complexity comparison is based on the analysis of problems that involve the simultaneous tracking of multiple (D) objects, each of which can assume K states each. The DBN structure for representing HMMs easily extends to HMM variants (hierarchical HMMs, factorial HMMs, coupled HMMs, semi-Markov HMMs, etc) without any modification to the learning or inference algorithms. Assumptions on the behalf of an HMM, which are discussed in further depth below, include: Markov independence, stationary model, and output independence.

For a sequence of length T there are T random state variables, $X = (X_1, \ldots, X_T)$, and T random observation variables, $Y = (Y_1, \ldots, Y_T)$. The joint distribution for an HMM represented as a DBN is

$$P(X_{1:T}, Y_{1:T}) = P(X_1)P(Y_1|X_1)\prod_{t=2}^{T}[P(X_t|X_{t-1})P(Y_t|X_t)].$$

The parameters are assumed to be the same over time (i.e. not a time varying model, rather stationary). The graphical model factorization for $T = 3$ is shown in figure 2.13.

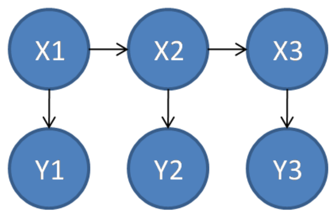


Fig. 2.13. DBN representation for an HMM with $T = 3$. Vertices whose names are prefixed with a Y are observed variables while vertices whose names are prefixed with an X are hidden variables.

The Markov property states that the future is independent of the past given the present, i.e.

$$X_{t+1} \perp X_{t-1}|X_t,$$

where $X_{t+1} \perp X_{t-1}|X_t$ means X_{t+1} and X_{t-1} are independent once we know X_t, while independence (not Markov) at the observation level is assumed to be

$$Y_t \perp Y_{t'}|X_t,$$

for $t \neq t'$. For each vertex, $P(X_1)$, $P(Y_t|X_t)$, and $P(X_t|X_{t-1})$, the CPDs must be specified. The CPD for $P(X_1)$ is typically specified as a vector, $\pi(i) = P(X_1 = i)$, where

$$0 \leq \pi(i) \leq 1,$$

and

$$\sum_i \pi(i) = 1.$$

The CPD for $P(X_t|X_{t-1})$ is typically represented as a stochastic matrix,

$$A(j,i) = P(X_t = i|X_{t-1} = j),$$

where each column sums to 1, i.e. for column i,

$$\sum_j A(j,i) = 1.$$

The CPD for $P(Y_t|X_t)$ is different for discrete and continuous cases. If Y_t is discrete, then

$$P\left(Y_t = y_j \middle| X_t = i\right) = B(i,j),$$

where B is a stochastic matrix, and for state i,

$$\sum_j B(i,j) = 1.$$

If Y_t is continuous, then $P\left(Y_t = y_j \middle| X_t = i\right) = \mathrm{N}(y_j; \mu_i, \Sigma_i)$, hence, for each state, a single normal distribution is used. Figure 2.14 shows the resulting DBN when the CPD, for the discrete case, is specified.

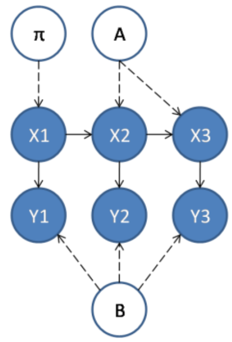


Fig. 2.14. An HMM represented as a DBN, in which all parameters are explicitly represented (nodes with outgoing dotted arcs). Notice that the parameters are stationary over time.

A useful extension involves modeling the observed variables as a mixture of Gaussians. In this case, $P(Y_t = y_j | X_t = i)$ is now explicitly modeled using vertices Y_t and M_t, shown in Figure 2.15, where

$$P\left(Y_t = y_j \middle| X_t = i, M_t = m\right) = \mathrm{N}(y_j; \mu_{i,m}, \Sigma_{i,m}),$$

$$P(M_t = m | X_t = i) = C(i, m).$$

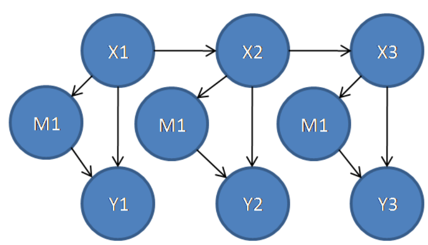


Fig. 2.15. DBN with a mixture of Gaussians for the observed variables.

Murphy presented solutions to multiple inference tasks, including: filtering (tracking the state over time), the most likely state sequence (Viterbi decoding), and classification (used to compute the likelihood of a sequence for different models) [74]. Classification is

$$P(y_{1:T}) = \sum_{x_{1:T}} P(x_{1:T}, y_{1:T}),$$

which is the summation of the probabilities for each possible state sequence ($x_{1:T}$). Exact inference algorithms include: convert the DBN to an HMM and use the forward-backward algorithm, use an unrolled junction tree, use the frontier algorithm, or the interface algorithm [74]. Approximate algorithms include assuming and solving for: a discrete distribution and the beam search, a single Gaussian using the unscented Kalman filter, a mixture of Gaussians and the GPB2 algorithm, set of samples and particle filtering, or variable sized and non-linear regression with online model selection [74]. Murphy proposes using the Expectation Maximization algorithm for learning a DBN [74].

The works reviewed are examples of general or generative directed graphical models. A brief overview of undirected graphical models is provided for completeness. Before presenting the model, Murphy provided a nice tree that illustrates the different organization of graphical models. This figure is shown in Figure 2.16.

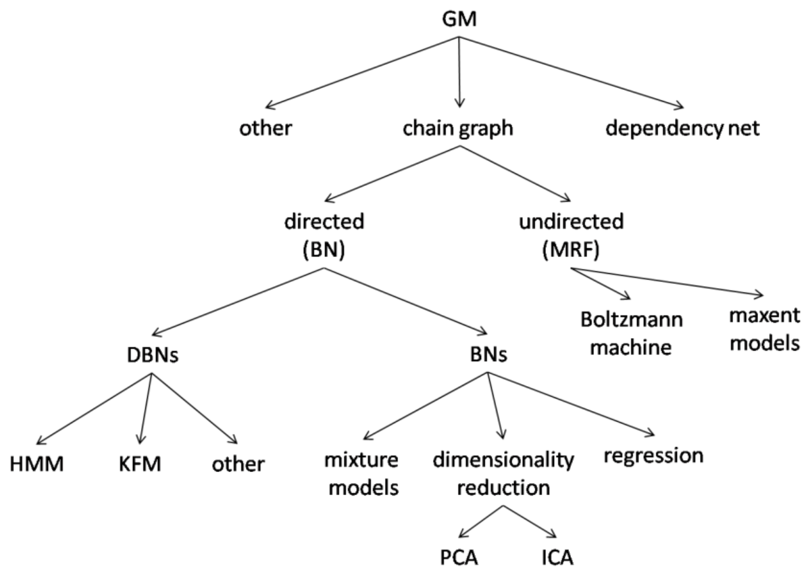


Fig. 2.16. Hierarchy of different kinds of graphical models.

An undirected graphical model, also known as a Markov random field (MRF) or a Markov network (MN), is an undirected graph. This category of graphical models is used extensively in domains such as statistics, robotics, and low level computer vision algorithms where the pixels are observable but the labels are hidden. The conditional independence properties encoded in a MRF are

$$V_A \perp V_B | V_C,$$

where V_A, V_B, and V_C are sets of vertices from V. This property states that iff all paths between all vertices in V_A and all vertices in V_B are blocked by some node in V_C, then there is some intervening $v_c \in V_C$ on every path between every $v_a \in V_A$ to every $v_b \in V_B$. This global Markov property implies that a single vertex v_i is independent of all other vertices in the graph given its neighbors, which are called the Markov blanket of v_i [74]. This property is illustrated in Figure 2.17.

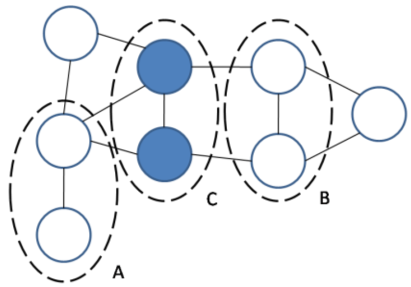


Fig. 2.17. Global Markov property on sets V_A and V_B which are blocked by V_c.

The probability distribution $P(V)$ is now a product of functions defined on maximal cliques. For every clique, C, where the set of vertices in the c$^{\text{th}}$ clique is V_C, there is an associated compatibility function $\psi_C(V_C)$, discussed below. The set of all maximal cliques is $\mathbb{C}$. The product is now represented as the joint distribution

$$P(V) = \frac{1}{Z}\prod_{C\in\mathbb{C}} \psi_C(V_C),$$

where Z is a constant ensuring normalization of $P(V)$. An important note is that the functions $\psi_C(V_C)$ are not required to be proper probability distributions, just non-negative [77]. The following factorization,

$$P(V_{1:5}) = \frac{1}{Z}\psi(V_1, V_2, V_3)\psi(V_3, V_4)\psi(V_4, V_5),$$

is represented in figure 2.18.

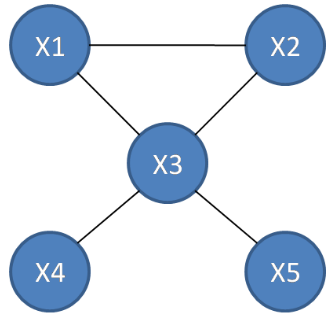


Fig. 2.18. Example MRF.

The typical low level pixel computer vision task assumes the following Markov field [79], shown in figure 2.19.

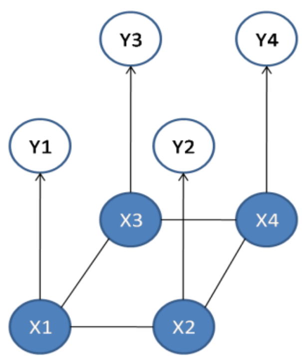


Fig. 2.19. Standard MRF for low level computer vision tasks. The Y's are observations and the X's are hidden.

As stated in Figure 2.19, this type of MRF is composed of hidden and observed random variables, where the gray vertices labeled with Y's are observations and the X's are hidden. In this type of MRF, clique sizes are fixed at 2. The factorization for this graph is

$$P(X,Y) = \frac{1}{Z}\psi(X_1,X_2)\psi(X_1,X_3)\psi(X_2,X_4)\psi(X_3,X_4)\prod_{i=1}^{4}\psi(X_i,Y_i),$$

and the potentials $\psi(X_i, X_j)$ are typically chosen to enforce some local consistency constraints (such as smooth segment edges), while the potentials $\psi(X_i, Y_i)$ are typically a conditional distribution, $P(Y_i|X_i)$, such as a Gaussian, $N(\mu_k, \Sigma_k)$, where k is an index for a segment (such as $k = X_i$). More about Markov random fields can be found in [74].

2.5.2 Graphical Models – Limitations

The most commonly selected type of graphical model for human activity recognition, as well as general gesture recognition, symbol recognition, etc, is the HMM. The Baum-Welch EM procedure for HMMs [78] is a well known technique for learning parameters in a hidden variable model. A fully connected model with S states has $S * S$ transition probabilities, S initial state probabilities, M mixtures for each state, and for a Gaussian with dimension D, there are D components in each mean and $D * D$ components in each covariance matrix. There are many parameters to estimate, even if the model is represented more compactly in the dynamic graphical model form. It is not commonly advertised and acknowledged that there is a severe problem related to the fact that most of the model parameters are only supported by a relatively small subset of the data. For HHMMs, even more parameters exist that need to be estimated. While research is being conducted in the area of learning model structure, such as entropy minimization [26], the majority of researchers still manually specify the model and its sparcity structure or conduct ad-hoc heuristic searches, making this procedure not as automated as most lead it on to be. All of this limits the predicative power of these learned models and their generalizability to a large set of sequences not included in the training set.

These approaches provide a way to compute the likelihood of a model given an observation sequence, the inference task. These likelihoods are useful for comparing which model is the most likely from a set, typically small, of trained models; pick one of K. However, this value is not something that can be easily interpreted as a confidence that the activity occurred, making the reliable rejection of activity not possible for most non-trivial real-world problems. This is especially the case in long observation sequences, even if scaling is applied [78].

Some people elect to train a small number of models to encompass all activities that the system should ignore, i.e. not one of the K to be recognized, called garbage or filter models [80]. The learning and representation of a large number of unknown activities using a small number of models is not a theoretically sound and a computationally tractable approach. In the case of speech and symbol recognition, tasks that can in many situations assume that an observation sequence was generated from one of K known models, likelihood values can be compared and the most likely model identified. However, human activity analysis does not have the structure to assume that an observation sequence was drawn from one of K known models. Threshold selection for model acceptance is the most unreliable approach. An equally sub adequate approach is acceptance based on the formulation of a ratio formed between the top two models, still requires the selection of a ratio threshold.

With respect to human activity analysis, start and end time points need to be identified, i.e. the silhouette sequence time window to analyze, $\{S_1, \dots, S_T\}$. This is not trivial, given the fact that the ultimate goal of the system is determining if an activity actually occurred, and if so, which activity? Hence, how does the system know where an activity started and ended if it does not know if an activity occurred? For tasks such as symbol recognition [81] or handwritten word recognition [82][83], reasonable assumptions can be made regarding an observation sequence as a series of features collected during a stroke, indicated by the pen touching down, dragging around, and then the pen being brought up, or in the case of word recognition, a vertical line scan and feature extraction across an image of pre-processed or assumed correctly oriented symbols. These are not assumptions that are reasonable to make for human behavior. A separate process needs to be responsible for recommending important relevant moments. Thus, sequences can be formed between these markers or a sliding fixed time window can be used. However, a sliding fixed time window assumes different activities occur over the same general time period and that a method exists for the reliable rejection of activity, i.e. an ability to say no activity was recognized at this time step.

What is really needed in the area of human activity analysis is not another non-interpretable likelihood value or the ad hoc training of garbage models, but a confidence value that can be understood and reliably used to reject unknown activities.

2.6.1 Related Work – Fall Recognition using a Single Camera and Silhouettes

The standard approach to silhouette-based human activity analysis involves a single camera and a combination of view dependent spatial and temporal features. Spatial features have the distinction that they are computable directly from S_t^F, while temporal features are computable from a window (set) of silhouettes, i.e. $\{S_{t-i}^F, \dots, S_t^F, \dots, S_{t+j}^F\}$.

A useful feature is the screen space axis aligned bounding box (AABB) of a silhouette. An AABB is a rectangle, specified according to its lower left $(x_{t,min}, y_{t,min})$ and upper right coordinates $(x_{t,max}, y_{t,max})$. The y dimension size of the AABB, $(y_{t,max} - y_{t,min})$, is useful for analyzing height related activities. An example is standing upright versus kneeling over or lying on the ground. However, a more discriminative feature identified for fall detection is the bounding box height to width ratio. When the subject is standing the width to height ratio is small, and when the subject is on the ground, the width to height ratio becomes much larger. The bounding box width to height ratio for the tth silhouette is

$$AABB_{t,ratio} = \frac{x_{t,max} - x_{t,min}}{y_{t,max} - y_{t,min}}.$$

When this value is near one, it is assumed that the individual is kneeling. Depending on the viewing position of the camera and the performance of activity as observed in the image plane, standing should have a value less than one, and lying down should generally be greater than one. In [84], Miaou et al. use an omni-camera, a camera able to view an entire hemisphere, to capture images from a top-down perspective. They used the width to height ratio of the silhouette and a single threshold to classify falls, resulting in a 79.8% fall detection accuracy.

The next feature is of utility as it relates to analyzing the orientation of an object. The off-diagonal covariance term for the t^{th} silhouette, $Cov_{t,xy}$, is found via

$$Cov_t = \frac{1}{(|S_t^F| - 1)} \sum_{i=1}^{|S_t^F|} (\vec{s}_{t,i} - \vec{\mu}_t)(\vec{s}_{t,i} - \vec{\mu}_t)^T,$$

$$\vec{\mu}_t = \left(\frac{1}{|S_t^F|}\right) \sum_{i=1}^{|S_t^F|} \vec{s}_{t,i},$$

which, in the two dimensional case is

$$Cov_t = \begin{bmatrix} {\sigma_x}^2 & \sigma_{xy} \\ \sigma_{yx} & {\sigma_y}^2 \end{bmatrix}.$$

In this work, $Cov_{t,xy} = \sigma_{xy}$ is used.

The next feature is an indication of an object's speed. The motion vector is the difference between the centroid of the silhouette at time t versus time t-1, $\vec{m}_{t-1\to t} = \vec{\mu}_t - \vec{\mu}_{t-1}$. The magnitude of the motion vector, $\|\vec{m}_{t-1\to t}\|$, is interpreted as a person's speed. This calculation is the speed of an object as observed in the image plane, which is not guaranteed to be the same as the speed of the object in the three-dimensional world. Someone walking towards a camera might have little to no speed, and a person walking horizontal in the image plane near the camera will have a different speed than a person walking horizontal in the image plane far away from the camera. Regardless, it is a temporal feature that can be used to distinguish different activities. Another commonly encountered feature is object acceleration, i.e. $(\vec{m}_{t\rightarrow t+1} - \vec{m}_{t-1\rightarrow t})$. Figures 2.20 and 2.21 illustrate the proposed spatial and temporal features.

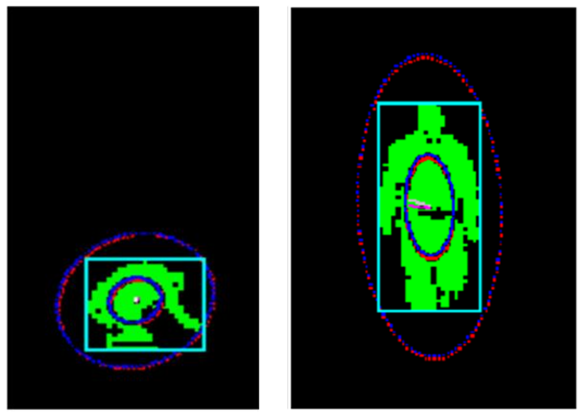


Fig. 2.20. Example spatial and temporal silhouette features for kneeling (left) and walking (right) activities. Green pixels are the silhouette, light blue is the bounding box, purple and white lines are the motion vector (difference of means and medians respectively), and the red and blue ellipses are one and three standard deviation for the covariance matrix (again, with respect to the mean and median respectively).

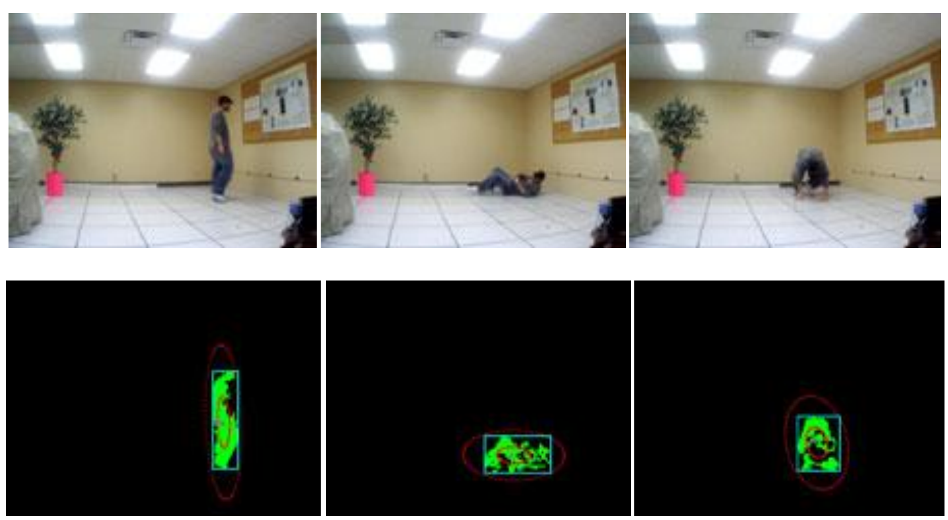


Fig. 2.21. Raw images, silhouettes, and silhouette features for standing, fallen, and bent over. Green regions are the foreground, red ellipses are the covariance tracked at one and three standard deviations, the light blue rectangle is the bounding box, and the purple lines are the mean-based motion vectors.

In [19], Anderson et al. demonstrates preliminary results for the task of classifying the activities walking, falling, and kneeling for single person indoor tracking using a single camera. These results are preliminary because only two sequences are used and the test data is the training data. The results are encouraging, showing separable clusters for $\vec{f}_{c,t} = < AABB_{t,ratio}, Cov_{t,xy} >^T$, and that the proposed activities can be recognized using an HMM. However, the above mentioned problems for a single camera resulted in the eventual shift towards a multiple camera back-projection voxel approach. Also, this work is fundamentally limited because of the probabilistic graphical model shortcomings discussed earlier.

2.6.2 Related Work - Recognition of Human Activity through Hierarchical Stochastic Learning

Lühr et al. presents a method for learning and recognizing human activity, particularly for, but not limited to, the elderly, using hierarchical hidden Markov models (HMMMs) [21]. HHMMs are a generalization of classical HMMs, in which the hierarchical structure of a problem is exploited for abstraction and modularity. It should be noted that all HHMMs are convertible into a "flat" or classical HMM. Another relevant and interesting observation stated in this paper is that HHMMs are a special case of stochastic context free grammars (SCFG), an observation originally noted in [22]. Details regarding the abstraction of an HMM to a HHMM, as well as the task of learning, still casted as an EM task, and inference, casted as a modified forward-backward procedure, can be found in [22]. Again, this work, due to the fact that it is a generalization of an HMM, is subject to the same limitations of probabilistic graphical models discussed earlier.

The authors use a GMM for change detection and a Kalman filter for tracking objects. Their tracker returns the world coordinates of a person's feet, which is mapped into discrete observations (an undefined label is assigned when the person is not in close proximity to any defined area) based on the person's Euclidean distance to predefined areas of interest, e.g. the door, fridge, food preparation area, sink, stove, or dining table. They use four different ways of preparing dinner and five typical lounge room activities. Each sequence is captured at 25 fps and over a period of 60 to 70 seconds. The tracker extracts an observation sequence for each recording through sampling the person's location every 25 frames. They learn four cooking and five lounge room models. Their data set consists of 36 training sequences, four per model, consisting of 16 cooking and 20 lounge room sequences.

Two analytical experiments are performed to view the results of learning the parameters under different conditions. The structure of the model is not learned. No classification results were generated for these two cases. They just perform analysis of the leaned model and its use for generating a description for new observation sequences. They first build two separate three layer models for the first two meal preparation scenarios, where there is one node at the top layer, one end node and three internal nodes in the middle layer. Each internal middle node was comprised of six terminal nodes and an end node. Upon analysis, they discovered that they could manually assign labels to each node in the middle level through analyzing the activity, context, and the way in which productions, and their associated pdf's, are used. The following labels were assigned to the first model: "cooking", "fridge and food preparation", and "enter/exit".

In the second experiment, the middle level nodes in the learned model were labeled: "cooking", "wash dishes", "enter/exit". By tracing the activity of the observation sequences through the model, i.e.

Viterbi algorithm for the most likely state sequence, they build labeled descriptions for the observation sequences, such as “enter/exit”->”fridge and food preparation”->”cooking”->”enter/exit”.

In their second experiment, they train a four layer model using all observations across all meal preparation performances. They pick a single node for the top fourth layer, two internal and one end node for the third level, three internal and one end node respectively for the second layers, and each bottom layer consisted again of six terminal and one end node respectively. Again, they manually label the learned hierarchy and they discovered that the results allowed for a further hierarchical subdivision of a task, into the manually labeled third level sequence “making dinner”->”finish and level”, which can be further decomposed, by going down another level, into “enter”->”cleaning and preparation”->”cooking”->”exit”. Again, labels for internal nodes are not automatically determined. The terms assigned to the labels were assigned manually.

For classification they use 27 held back sequences. The kitchen and lounge room models are tested on their respective sequences, given that spatial location can be used to automatically infer which models to use. They perform the same activity with different durations and variations in the transition order between the areas of interest, still with the assumption that the sequence was one of the four, i.e. not an unknown activity. For the cooking models, they only have two false alarms out of 12. One “prepare then cook” was mistaken for a “wash, prepare, and cook”, and one “prepare and cook” was mistaken for a “round robin”, where a subject transitions between each area of interest, starting and ending at the stove before leaving the room. For the lounge room sequences, three out of 15 sequences are misidentified. In two situations, dinner is mistaken for television dinner, and in one situation television dinner is mistaken for the dinner model.

2.6.3 Related Work - Recognition of Composite Human Activities through Context-Free Grammar-based Representation

Ryoo and Aggarwal use a context-free grammar (CFG) to recognize composite actions and simple interactions between people [85]. Aggarwal previously demonstrated a computer vision-based system for segmenting the human body into blobs and then into human body parts, specifically into hair, face, torso, hands, legs, and feet regions [86]. The blobs are used to estimate the human's pose using graphical models (Bayesian networks). From this, gesture information is inferred, again using graphical models (hidden Markov models). The authors, like many, adopted the use of the training of noise, garbage, or filter models for the recognition of unknown activity. Again, this is not a theoretically sound or reliable procedure. Their work with a CFG begins with the classification of human activity into three event categories: atomic actions, composite actions, and interaction. Atomic actions are the simplest component of human activity, meaning they are not divisible into smaller components (gestures). Composite actions contain two or more atomic actions. Each event is associated with a time interval. The object of inquiry for events is the sequencing, hierarchical nature (composite), and possibly the interaction of events between multiple people. Their events are assigned to an English vocabulary. Temporal predicates, such as 'before', 'meets', 'overlaps', 'starts', 'during', and 'finishes' are defined using equations relating the interval endpoints. Spatial predicates used were 'near' and 'touch'. Logical predicates included 'and', 'or', and 'not'.

Atomic action intervals use the time interval of gestures. The authors used a hand designed CFG for the recognition of composite and interaction events. Composite actions (non-terminals) are specified using actions, thus, both atomic and composite, where atomic actions are terminals. A composite action production is specified using a parameterized rule based on a sequence of required actions, including specific properties about their performances, and temporal relationships, using interval properties of the events, between the events. Their complicated production format is just a general way of compactly and abstractly specifying rules, taking into consideration desired temporal, spatial, and logical properties. The non-terminals in this CFG are assigned parameters based on the elements that matched their rule format. Because interaction events involve multiple people, these productions utilize constraints about the spatial information of the participating people.

The syntax of the CFG is rather simple and fixed. The authors used a parameterized system for specifying and recognizing productions, which took into consideration the element's temporal, spatial, and logical properties. As a result, they correctly stress that this work is ultimately one of semantics versus syntax of the language. Little crucial syntactic investigation was performed. It is also somewhat hard to determine given their description, but it appears that they manually derive a syntax tree in a bottom-up fashion. It appears that the recognition of composite and interaction events are based on their presence as a node in the syntax tree.

The results focused on two-person interactions. The interactions tested included: approach, depart, point, shake-hands, hug, punch, kick and push. The extent of their results are very preliminary, however the recognition rates are impressive (lowest of 83% up to 100%). They say that the system can handle noisy inputs from the HMMs. However, it cannot handle "large scale" errors, such as insertions or deletions of sub-events. They state that in the future they plan to extend this work to large data sets and to take into account the probabilistic nature of the domain.

2.6.4 Related Work - Recognizing Multitasked Activities using Stochastic Context-Free Grammar

Moore and Essa use a stochastic CFG (SCFG) for activity recognition in blackjack [87]. Their goal is the recognition of separable, multi-tasked activities, where multi-tasked activities are assumed to be composed of several single-tasked activities. They define multitasked activities as "the intermittent cooccurrence of events involving multiple people, objects, and actions, typically over an extended period of time." [87]. They note that it is not clear whether events occur independently, interactively, or in some combination of the two. They are interested in separable activities, which they define to be characterized by "wholly independent interactive relationships between two or more agents, i.e., multiple speakers, but separate, independent conversations" [87]. A CFG is used to explain dependencies between interactions occurring within separable groups. Their specific contributions are new parsing strategies that enable error detection and recovery in SCFG.

A SCFG is an extension of a CFG, where each production is assigned a probability. The authors estimate the rule probabilities through calculating the average production likelihood. In most SCFG's, the probability of a particular derivation is the product of the rule probabilities,

$$\rho_{\psi_i} = \rho_1 * \ldots * \rho_{m_i},$$

where ψ_i is the ith string derivation and ρ_i is the probability of a production. The common algorithm for parameter estimation in a SCFG is the inside-outside algorithm [88], which has cubic time complexity in the length of the observed sequences.

Terminals in their SCFG are events, detected using their computer vision system VARS. They track the hands, cards, and betting chips. Their vision-based system classifies hands to the specific subject, while cards and chips are tracked by analyzing the last subject to touch an object. The environment is highly controlled. A camera is placed above the card table looking downward only at the hand and card region. There are no other skin regions, such as the face, the players are wearing long sleeve shirts, and there appears to be little chance for confusion of objects given they are the only objects in the scene. As events are observed, their symbol (i.e. terminal) is appended to a string. Each event is assigned a probability that specifies the likelihood of the detection of that event. The authors modify the Earley-Stolcke parsing algorithm for SCFGs [89]. They modify the algorithm by multiplying their symbol (input) likelihoods with the originally proposed parsing scanning steps, which did not originally consider symbol likelihoods, but only production probabilities. Because there are uncertainties present in the input, there is no guarantee of a unique parse. As a result, they use a generalization of the Viterbi algorithm for retrieving the most likely parse across all possible string derivations. They use a familiar recursive procedure to recover the maximum likelihood derivation tree associated with the Viterbi parse.

The authors selected the game of blackjack for analysis. They refer to this game as complicated, however it is more likely that it was selected because it has a set of formal well-defined rules. They state that because there is rarely any correlation between player interactions, each player-dealer pair is a separable group. Their detected events include, to name a few, ‘dealer removed card from house’, ‘player added card to house’, ‘player bets chip’, etc. The SCFG is hand designed and each production is assigned a linguistic meaning, such as ‘recover card’, ‘settle bet’, ‘house hits’, etc. As already noted, the probabilities are assigned to their predetermined hand designed productions through calculating the average production likelihood.

The authors provide three techniques for detecting and recovering from failures caused by errors. Using their methods, the parser now generates a syntactically legal interpretation but provides no guarantees of its semantic legitimacy. The first error type is substitution error, which occurs when the wrong terminal symbol is generated because the actual event is not detected as the most likely. Insertion errors occur when erroneous symbols that do not correspond to the actual event are added to the input. Deletion errors are failures to detect events that actually occurred. They note that given their specialized system, substitution and insertion errors are rare for them. The authors note that Ivanov and Bobick [90] modify the grammar to handle substitution and insertion errors. Instead, Moore and Essa modify the parsing algorithm to accommodate the recognition of erroneous strings. For insertion errors, they ignore the scanned terminal and return the state of the parser to the point prior to scanning. If a substitution error occurs, they promote each pending prediction state as if it were actually scanned, creating a new path for each hypothetical terminal. A hypothetical path is terminated if another failure occurs in the next real scanning step. If a failure occurs because of a deletion error, they promote each pending prediction state and create a separate path for each hypothetical symbol. When a deletion error is assumed, there is no detection likelihood to recover for the missing symbol. Instead, they approximate this likelihood using empirical values. To address the computational complexity of the parsing for larger grammars or for long terminal strings, they prune recovered paths that have low overall likelihoods.

The authors show results for three experiments. In experiment 1, they evaluate event detection accuracy. Twenty-eight sequences, where each was a full game of Blackjack with at least one player were used, which resulted in the generation of 700 example events. The reported detection rate for all events was 96.2%, while the error rates (determined manually) for insertion, substitution, and deletion errors were 0.4%, 0.1%, and 3.4% respectively. In experiment 2, they evaluated the error detection and recovery. When presented with no insertion, substitution, or deletion errors, they achieved 100% accuracy. They used two data sets to validate this second experiment. In summary, they found improvements in parsing from approximately 42% (with error recovery disabled) to 85% (with error recovery enabled), where a successful parse is one in which the string could be entirely parsed without terminating in failure. They determined that 22.5%, 17.5%, and 60.0% of errors were caused by insertion, substitution, and deletion errors, respectively. In the third experiment, high-level behavior assessment was performed. Players were grouped into low-risk, high-risk, novice, and expert categories. The detection of low-risk players was 92%, high-risk 76%, novice 100%, and expert was 90%. However, no mention was made to classification rates when their parsing strategies were disabled.

2.6.5 Related Work - Fuzzy Ambient Intelligence for Next Generation Telecare

Martin et al. have related work in the area of soft-computing technologies for monitoring the "well-being" of elderly residents [45]. They did not use video sensing technologies, only passive infrared (PIR) sensors, for detecting movement in rooms, toggle switch sensors for detection of windows, entrances, and kitchen cupboards and appliances, vibration and temperate sensors, for the detection of water usage, and pressure sensors, for indications of occupation in chairs and the bed. The categories they wished to montor include: social interaction, both in and outside of the house, personal goals (hobbies, caring for oneself and others, etc), and activities of daily living (cooking, washing, cleaning, etc). However, using their relatively simple non-intrusive sensors, they were only able to measure: leaving and returning home, receiving visitors, preparing food and eating, sleeping patterns, personal appearance, and leisure activities. They are not explicit on how they exactly monitor each category, such as personal appearance and leisure activities. Details are only provided for sleeping and the detection of visitors.

The time stamp of the raw sensor readings were replaced by their fuzzy membership values in 12 fuzzy sets: dawn, morning, afternoon, evening, each of which was described using three modifiers: early, late, and just the term itself. In addition, the duration of various sensor readings, such as bed and kitchen occupancy, were translated from their raw format into fuzzy membership values. The duration concepts very short, short, medium, long, and very long, were used, where the specific fuzzy sets are defined over each individual domain. Thus, even though the terms are the same, their meanings take on different interpretations over different domains, such as long bed occupancy versus long kitchen occupancy. The fuzzy sets for time were manually designed by domain experts, while the fuzzy sets for duration were initially manually designed, but augmented using statistical information from a training data set.

Their analysis consisted of fuzzy logic, statistics, association analysis, and trend analysis. For association analysis, they used an extended version of the Apriori algorithm, known in the data mining community for learning data associations, which discovers associations and confidences among concurrently occurring symbols, daily activities in their case. They mention that they analyze changes in patterns and associations, as well as learn the models, but no real details are provided on exactly how they do this. They insufficiently state that monitoring sleep and predicting restlessness is performed by using “simple rules”. It is unknown if they used crisp or fuzzy rules for the detection. In order to detect visitors they partially explain a few measures that are calculated from the sensor data. The first measure is the activity level. It is defined as the proportion of time in which motion or object use, across all monitored categories, is detected over an interval defined by entrance and exit events. Their assumption is that as more people are in the room, the number of activities detected will rise. They track the change in activity level and use fuzzy sets for describing the amount of change, such as big fall, fall, steady, rise, big rise, and use training data to refine the parameters of their initially manually specified fuzzy sets for a particular resident. They also calculate features such as the number of times a person moved between rooms, amount of time spent in the hallway, and door open time. They use fuzzy sets to characterize these features. Ultimately, three fuzzy logic rules are used for visitor detection, and antecedents are formed from the measures and their fuzzy sets just discussed.

Their system was tested in the home of two elders, one for 6 months and the other for 18 months. Validation was performed by keeping a diary of the resident's activities by calling them occasionally over a period of several weeks. They also captured a week's worth of video data for validation, where video cameras were said to be installed in non-intrusive areas, which are not in the bathroom and bedroom. However, there was no attempt to use the video for a computer vision system. A plot of measured sleep activity is displayed, along with simple linear regression and a moving average, however no quantitative results are presented to show how well their system performed. They reported a confusion matrix for 50 front door opening events. They correctly detected nine visitations, they missed one visitation, they called four cases visits when there was not a visitor, and 36 no visitor events were correctly detected.

2.6.6 Related Work - Video Summarization and Scene Detection by Graph Modeling

Video summarization is a crucial concept in this dissertation, as well as in the following reviewed work. Therefore, a quick overview is necessary. The majority of video summarization work can be categorized into static storyboarding (keyframes, reprehensive frames, etc) or video skimming (moving storyboarding, summary sequence, etc) [91]. The former consists of a collection of salient images extracted from the underlying video source. The latter is composed of a collection of video segments, and possibly audio, linguistic information, etc, extracted from the original video. While these are two different categories, much work has gone into converting one representation into the other [92][93][94]. Methods for storyboarding range from uniform sampling, clustering based on Gaussian mixture models [95] or normalized graph cuts [96], to measures of sufficient content change [97]. Video skim generation is newer and fewer approaches have been proposed to address this category. The work discussed below is one of this category, and a good review of work in redundancy elimination, event/highlight detection, and more can be found in [91].

Ngo et al. presented a method for video summarization based on scene modeling, using a normalized graph cut algorithm [62], temporal graph analysis, and highlight detection, which used motion attention modeling [96]. The assumption is that video is composed of scenes, which are composed of one or more shots. Clusters contain one or more shots with similar visual content. The authors begin by partitioning the video into shots using a proposed spatio-temporal slice model, in which three temporal slices, horizontal, vertical, and diagonal, are extracted from an image volume. The slices are two-dimensional images with one dimension comprising space and the other time. A complete undirected weighted graph is then constructed, where shots are nodes, and edges indicate shot similarities. Normalized graph cutting is then used to decompose the graph, in a hierarchical binary tree fashion, into clusters. In parallel with this stage, they utilize a motion attention model based on the manipulation of motion vector fields extracted from MPEG video streams. They use intensity, spatial coherence, and temporal attention induction formulas to calculate the degree of attention on behalf of an observer to a specific individual frame. The attention values and the clusters form a directed temporal graph. The authors suggest the use of the shortest cost path to eliminate edges in the graph and induce sub graphs that represent scenes.

In this video skim work, summarization is the process of selecting video segments. The user provides a skim ratio, $0 \leq R \leq 1$, which specifies the percentage of the video to keep, i.e. approximately

$(1 - R)$ percent of the video is discarded. Using the acquired scenes, clusters, and shots, the authors provide a few metrics to determine the amount of contribution of each to the video. The basic idea is the elimination of weakly expressed material. They test their results by using hand segmented video sequences, in which the scene border points are provided, and are used to compare against their findings. The recall of their procedure is 90% and the precision is 74%. In addition, they performed human subject experiments, where the role of the subject was the identification of a number that expressed the amount of informativeness and enjoyability of the summarized video sequences. Informativeness addresses the capability of maintaining content coverage and the reduction of redundancy. Enjoyability was the performance of the motion attention model in selecting perceptually enjoyable video segments for summaries. They used 20 students, and found that when retaining only 10% of the video, the average enjoyability score was 70.44%, when retaining 25% of the video the score was 80.93%, 100% retention was considered to be 100%. The informativeness of 10% retention was 70.38%, while the 25% retention average score was 82.50%. Again, 100% retention was considered to be 100%.

2.7 Summary

In summary, the state of the art for human activity analysis is a combination of probabilistic graphical models [74], specifically variations on HMMs [78], and single camera, image space, background model-based change detection, specifically variations on the GMM [24]. Probabilistic graphical models are attractive because they take into account one type of uncertainty, chance, and well-known machine learning algorithms exist [78][74], e.g. Expectation Maximization. However, as discussed above, probability theory and statistical inference have many shortcomings when it comes to the task of activity modeling and, namely, inference. Additionally, single camera silhouette segmentation is restrictive because objects and subsequent features are too view dependent. These background sections review existing noteworthy approaches to activity analysis and include the theoretical basis, namely fuzzy sets and approximate reasoning, necessary to understand the following linguistic summarization work.

Human Segmentation in Image Space

3.1 Introduction

The silhouette extraction system developed for this dissertation is that of Luke, Anderson and et al. [27]. The system is described here in detail because it is critical in the regard that it is the input to the higher level behavior analysis component of this work. Additionally, it has only appeared as a technical report thus far and it might not be easily or always accessible. The proposed single person indoor human change detection system is adaptive, incorporates both texture and color information, and performs shadow removal. Figures 3.1 and 3.2 show the proposed system.

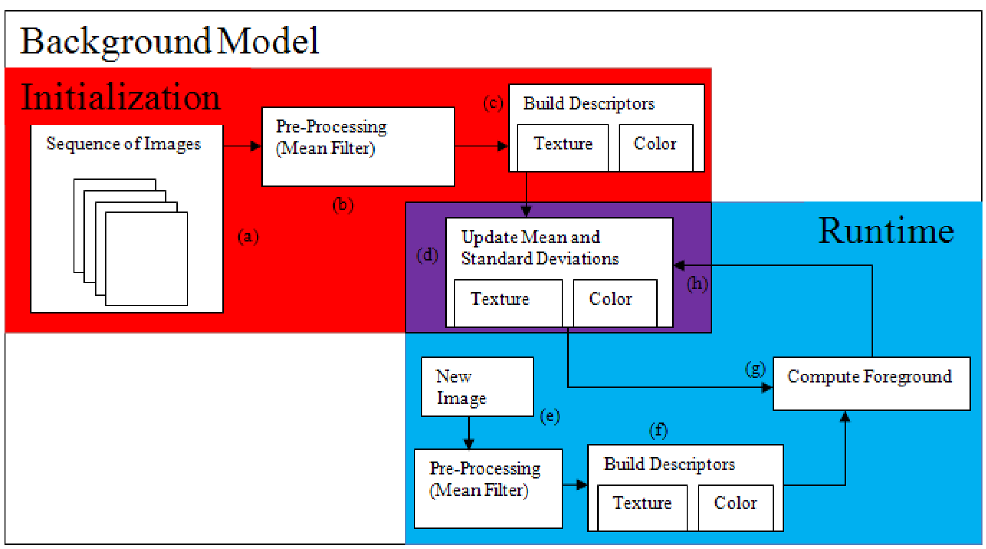


Fig. 3.1. A graphical representation of the background model. (a) A sequence of ten images is input to the system. (b) A 3x3 mean filter is passed over the red, green and blue color planes of each input image. (c) Color and texture descriptors are then extracted from these images. (d) Finally, the means and standard

deviations of the descriptors at each pixel are found over the input sequence. (e) During runtime, a 3x3 mean filter is passed over each new image. (f) Color and texture descriptors are then extracted from the image. (g) The foreground is then found for the new image using the background model. (h) The background model is then updated.

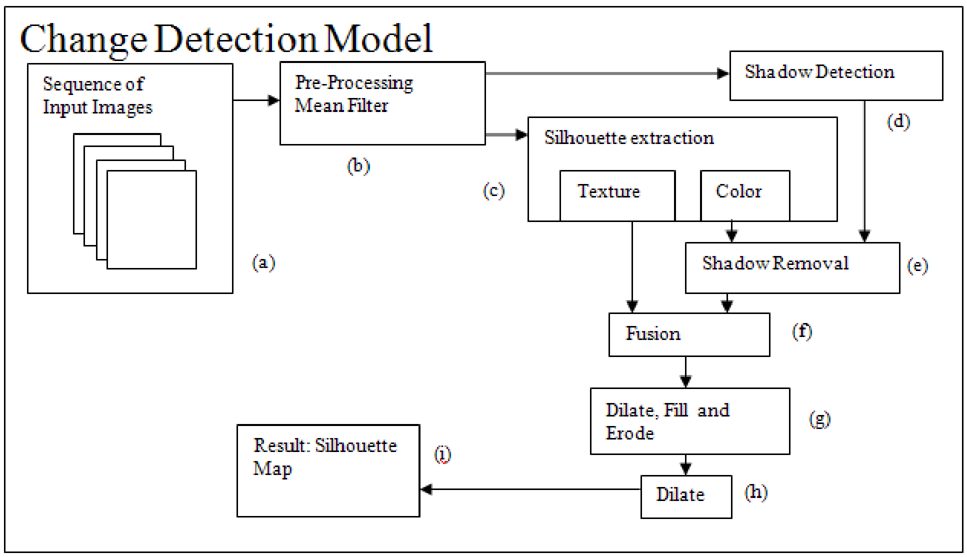


Fig. 3.2. Silhouette extraction procedure. (a) Images are captured from a video camera in the sensor network. (b) A 3x3 mean filter is passed over the red, green and blue color planes of each input image. (c) Silhouettes are found in color and texture features. (d) Shadow regions are identified. (e) Shadows are removed from the output of silhouettes found from color features. (f) The results of the texture and color silhouettes are fused. (g) Morphological and logical operators are applied to the output of silhouettes to remove false alarms and fill-in silhouettes. (h) The fused data is then dilated to correspond to the size of the subjects in the image. (i) The final silhouette output image.

3.2 Texture Features

Texture is a reliable source of information for scene description and change detection. Previous research in texture features includes [98][99][100][101][102]. A smaller amount of research has been focused on texture in color spaces [103][104][105][106]. It is the belief here that color texture features are more robust than their monochromatic counterparts and are necessary for the modeling of a complex background.

Numerous factors go into the choice of color space. Because lighting is an uncontrolled variable in the scene, the color space chosen to extract textural information must be robust to lighting changes. The red, green and blue gamma corrected channels output from the camera make up the $R'G'B'$ color space. Though $R'G'B'$ color space is easily produced, features in this space are susceptible to changes in brightness. The HSV color space; hue, saturation, value; has advantages over $R'G'B'$ space. Most importantly, it separates chroma, (color), and luma, (brightness), components. However, there are still some shortcomings. First, the luma component is unreliable for texture because changes in lighting or hard shadows cause drastic changes in the signals received. Also, when luma is low, the hue is unreliable. In addition, though the chroma and luma are separated, experimentation shows that the saturation is correlated with light intensity.

The YC_bC_r [107] color space was selected for this system due to its reliability in a wide range of lighting situations. This color space again has a luma component; defined using the constants K_b, K_r, and K_g which are based on perceived brightness of human vision to blue, red and green [107]; but it separates the blue and red chroma components, C_b and C_r. We have extended the YC_bC_r space to also include a green component. This C_g component is extracted in a fashion similar to the C_b and C_r components. These three chroma quantities describe how much the red, green and blue components contribute to luma. After experimentation, it was determined that better results were achieved after normalization by dividing the C_b, C_r and C_g components by the luma Y. The new C_b', C_r' and C_g' components used to extract feature descriptors are defined as

$$R', G', B' \in [0,1]$$

$$K_b = .114$$

$$K_r = .299$$

$$K_g = 1 - (K_b + K_r) = .587$$

$$Y = K_r R' + K_g G' + K_b B'$$

$$C_b' = \frac{\left(\frac{B' - G' - \frac{K_r}{1 - K_b}(R' - G')}{2}\right) + .5}{Y + 1}$$

$$C_r' = \frac{\left(\frac{R' - G' - \frac{K_b}{1 - K_r}(B' - G')}{2}\right) + .5}{Y + 1}$$

$$C_g' = \frac{\left(\frac{\left(-\frac{K_r}{K_b + K_r}R'\right) + G' - \left(\frac{K_b}{K_b + K_r}B'\right)}{2}\right) + .5}{Y + 1}.$$

The silhouette change detection features calculated using the $C_b'C_r'C_g'$ space are based on the Edelman descriptor, [108]. The practical use and robustness of the Edelman descriptor have been displayed in the Scale Invariant Feature Transform, (SIFT) [109]. First, gradients are computed for each pixel in each of the C_b', C_r' and C_g' components. For each pixel in each gradient image, an eight dimensional histogram, h, is built using a five by five window of gradients. The orientation of each pixel's gradient, g_o, determines into which bin, b_r, the gradient resides. The bin index b_r is the floor of g_o divided by 45^o and the second nearest bin is b_n. The gradient's magnitude, g_m, is spread into bins b_r and

b_n based on the value $\propto$, which represents how close the gradient's direction is to the center of the nearest bin. The value $\propto$ is found by first computing the modulus of the gradient direction by 45^o. The absolute difference of this value with 22.5^o represents how many degrees this gradient is off from the center of the bin. Finally, this value is divided by 45^o. Hence formally, for each color component,

$$g_o \in [0,360)$$

$$g_m \in [0,\sqrt{2}]$$

$$g_o = \begin{cases} mod\left(\left(\tan^{-1}\frac{d_y}{d_x} + 2\pi\right)\left(\frac{180}{\pi}\right), 360\right) & if\ d_x > 0 \\ \left(\tan^{-1}\frac{d_y}{d_x} + \pi\right)\left(\frac{180}{\pi}\right) & if\ d_x > 0 \\ 90 & if\ d_x = 0\ and\ d_y > 0 \\ 180 & if\ d_x = 0\ and\ d_y < 0 \\ 0 & else \end{cases}$$

$$g_m = \sqrt{{d_x}^2 + {d_y}^2}.$$

For each gradient in a window for a given pixel, the associated histogram h is updated as

$$b_r \in [0,7]$$

$$b_n \in [0,7]$$

$$\propto \in [0,.5]$$

$$b_r = \left\lfloor\frac{g_o}{45}\right\rfloor$$

$$b_n = \begin{cases}(b_r + 1)\ mod\ 8 & if\ g_o mod\ 45\ \geq 22.5 \\ (b_r + 7)\ mod\ 8 & else\end{cases}$$

$$\propto = \frac{|g_o mod\ 45 - 22.5|}{45}$$

$$h(b_r) = h(b_r) + (1{-}\propto)g_m$$

$$h(b_n) = h(b_n) + \propto g_m.$$

The gradient magnitude is the Euclidean distance of the change in the horizontal and vertical direction and therefore has a maximum value when the horizontal and vertical change are both 1, such that

$$g_{m_{max}} = \sqrt{1^2 + 1^2} = \sqrt{2}$$

The results from the process outlined above are three images where each pixel is associated with an eight bin histogram descriptor. Each eight bin histogram represents the accumulation of gradient magnitudes in eight directions for a window of texture. The strength of this descriptor is its matching ability and robustness to pixel jitter. The pixel jittering phenomenon refers to how consistently a camera registers pixel values through time. This jitter is due to noise in the perceived intensity of red, green and blue elements in the CCD of the camera. The cameras used in this system are low cost web cameras that more accurately capture intensity than color information. Human vision is more sensitive to intensity changes than color changes, making these cameras suitable for communication over the web. Higher quality cameras with less pixel jitter could be used, but because this system is to be deployed in an assisted living community, the most economically viable option is to use cheaper web cameras.

3.3 Color Histogram Features

The use of texture by itself is not enough in many situations. It is possible that change has occurred and while texture remains similar color is different, or vice versa. This system uses the HSV color model to build a color histogram at each pixel. Using hue, H, and saturation, S, is preferred over the $R'G'B'$ color space as explained earlier, because this representation separates chroma from luma. The HSV model [107] is defined as

$$\vee - Max$$

$$\wedge -Min$$

$$H \in [0,360),$$

$$S, V, R', G', B' \in [0,1],$$

$$B_{max} = R' \vee G' \vee B',$$

$$B_{min} = R' \wedge G' \wedge B',$$

$$H = \begin{cases} 0, & if\ B_{max} = B_{min} \\ 60\left(\frac{g-b}{b_{max}-b_{min}}\right), & if\ B_{max} = R'\ and\ G' \geq B' \\ 60\left(\frac{g-b}{b_{max}-b_{min}}\right) + 360, & if\ B_{max} = R'\ and\ G' < B' \\ 60\left(\frac{b-r}{b_{max}-b_{min}}\right) + 120, & if\ B_{max} = G' \\ 60\left(\frac{r-g}{b_{max}-b_{min}}\right) + 240, & if\ B_{max} = B' \end{cases},$$

$$S = \begin{cases} 0 & if\ B_{max} = 0 \\ 1 - \frac{B_{min}}{B_{max}} & else \end{cases},$$

$$V = B_{max}.$$

Similar to the texture descriptor, for each pixel in the image, a histogram is built using the local color information. The 360^o hue component is discretized into eight bins, similar to the gradient orientation in the texture feature, resulting in a feature vector of length eight at each pixel.

As mentioned earlier, if the light intensity is low, then hue is unreliable. In an extreme example, the $R'G'B'$ value of (.01,0,0) has a hue of 0^o and a saturation of 1, while the $R'G'B'$ value of (0,.01,0) has a hue of 120^o and a saturation of 1. So, although these colors are nearly identical to a human observer, they are represented as very different values in HSV color space. Due to saturation being a measure of color purity, both of these values make sense in HSV space, but cause great difficulties when trying to compute similarity. We therefore define a new term, “brightness saturation”, S_v, as

$$S_v = S * V = \left(1 - \frac{B_{min}}{B_{max}}\right) B_{max} = B_{max} - B_{min}.$$

This brightness saturation value is spread between the two nearest bins from the hue discretization, similar to the gradient magnitude in the texture feature.

3.4 Change Detection

Before calculating the occurrence of change, a background model must be built and maintained. For this system, the background is modeled with a single Gaussian distribution, with standard deviation modeling the pixel jitter from each pixel's mean. The first T images, (we use 10), of a sequence are used to initialize the background model. The texture and color feature vectors described above are built for each of these images. The means of the texture and color vectors; $\mu_{h_{C'_b}}$, $\mu_{h_{C'_r}}$, $\mu_{h_{C'_g}}$ and $\mu_{h_{HS_v}}$; represent the average background while the standard deviations; $\sigma_{h_{C'_b}}$, $\sigma_{h_{C'_r}}$, $\sigma_{h_{C'_g}}$ and $\sigma_{h_{HS_v}}$; represent pixel jitter.

To detect change at each pixel, the absolute values of the difference between the current frame and the mean vectors are calculated. It is assumed that any values of change less than two standard deviations from the mean are noise and are therefore ignored. Beyond two deviations, the new observation is assumed to be a significant change from the background. It is this value beyond the noise range that we want to keep for change detection. Therefore, two standard deviations are subtracted from the amount of change at each bin. Subtracting the standard deviation, instead of the more common operation of dividing the change value by the standard deviation, has the added benefit of not possibly causing a possible divide by zero error.

The differencing method is performed on the C'_b, C'_r, C'_g and HS_v histogram images to compute

$$C = 2$$

$$\Delta_{h_{C_b'}}(x,y,i) = 0 \vee \left(\left| \mu_{h_{C_b'}}(x,y,i) - h_{C_b'}(x,y,i) \right| - C * \sigma_{h_{C_b'}}(x,y,i) \right),$$

$$\Delta_{h_{C_r'}}(x,y,i) = 0 \vee \left(\left| \mu_{h_{C_r'}}(x,y,i) - h_{C_r'}(x,y,i) \right| - C * \sigma_{h_{C_r'}}(x,y,i) \right),$$

$$\Delta_{h_{C_g'}}(x,y,i) = 0 \vee \left(\left| \mu_{h_{C_g'}}(x,y,i) - h_{C_g'}(x,y,i) \right| - C * \sigma_{h_{C_g'}}(x,y,i) \right),$$

$$\Delta_{h_{HS_v}}(x,y,i) = 0 \vee \left(\left| \mu_{h_{HS_v}}(x,y,i) - h_{HS_v}(x,y,i) \right| - C * \sigma_{HS_v}(x,y,i) \right).$$

These differences are summed across the eight histogram bins to find the total change at each pixel of the image

$$\Delta'_{h_{C_b'}}(x,y) = \sum_{i=0}^{7} \Delta_{h_{C_b'}}(x,y,i),$$

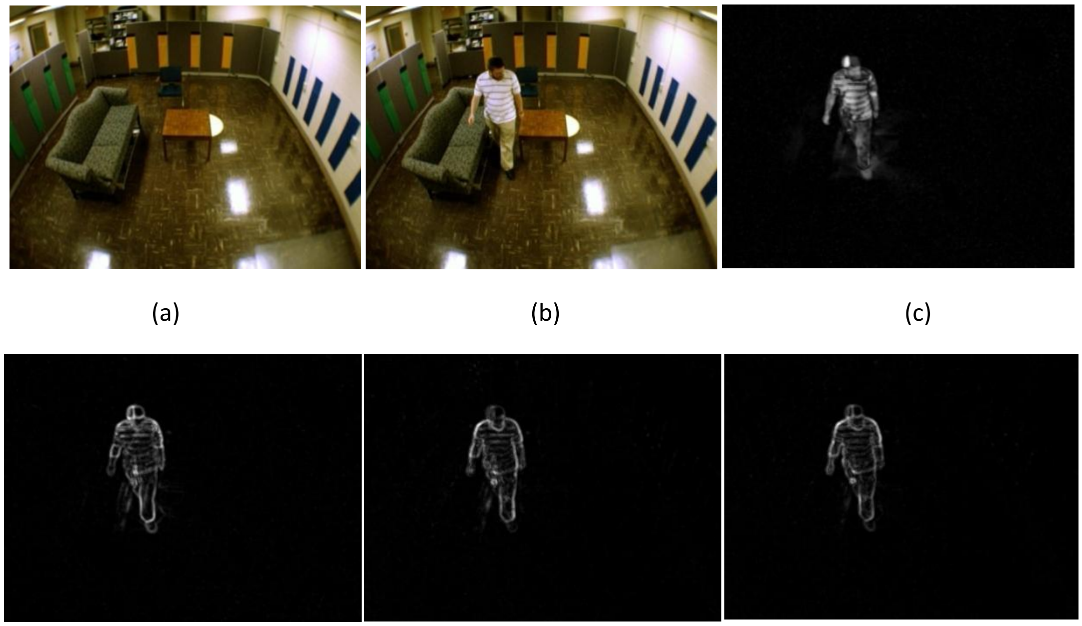
$$\Delta'_{h_{C_r'}}(x,y) = \sum_{i=0}^{7} \Delta_{h_{C_r'}}(x,y,i),$$

$$\Delta'_{h_{C_g'}}(x,y) = \sum_{i=0}^{7} \Delta_{h_{C_g'}}(x,y,i),$$

$$\Delta'_{h_{HS_v}}(x,y) = \sum_{i=0}^{7} \Delta_{h_{HS_v}}(x,y,i),$$

where (x,y) is the pixel location, i is the i^{th} histogram bin, and h_{CS} is the histogram of color space CS.

At every pixel in each color space, the change is a single scalar and the resulting picture can be thought of as a difference image. The four difference images for a single frame are shown in figure 3.3.



(a) (b) (c) (d) (e) (f)

Fig. 3.3. Images representing the amount of change according to each texture and color descriptors. (a) Background image (b) Test image (c) HS_v features Diff Image (d) C_b' features Diff Image (e) C_r' features Diff Image (f) C_g' features Diff Image

The fusion of the changes of the C_b', C_r' and C_g' components require special care because in this color space, changes to secondary colors; cyan, magenta and yellow; register in multiple color planes. Hence, their responses to change are smaller in each plane than for primary colors. A Yager union [110] is used to fuse the texture changes. The operator is defined as

$$\Delta'_{h_{C'_{brg}}}(x,y) = 1 \wedge \left(\Delta'_{h_{C'_b}}(x,y)^w + \Delta'_{h_{C'_r}}(x,y)^w + \Delta'_{h_{C'_g}}(x,y)^w\right)^{\frac{1}{w}}.$$

The variable w is a tunable parameter, which can be manually assigned or learned from training data, which set to 2 here. Values of $\Delta'_{h_{C'_{brg}}}(x,y)$ above a given threshold, 0.4, represent pixels that differ from the background.

$$\varphi_{h_{C'_{brg}}}(x,y) = \begin{cases} 1 & \Delta'_{h_{C'_{brg}}}(x,y) > 0.4 \\ 0 & else \end{cases}$$

The result of this fusion is a binary image, where the pixels along the contour of the person and areas of large textural change are classified as foreground.

Values of the color descriptor change, $\Delta'_{h_{HS_v}}$, above a given threshold, 2, represent pixels of color change. These pixels are defined as

$$\varphi_{h_{HS_v}}(x,y) = \begin{cases} 1 & \Delta'_{h_{HS_v}}(x,y) > 2 \\ 0 & else \end{cases}$$

Figure 3.4 shows the change detected from the HS_v features. Unfortunately, color change detection in HS_v is vulnerable to shadows cast by moving objects, as is shown in Figure 3.4. This problem with shadows is again due to the correlation between saturation and luma. It is therefore necessary to detect and remove shadows.

The shadow detection algorithm here is an extension to an earlier model proposed by Blauensteiner et al., [111]. In [104], circular statistics are built from hue and saturation mapped into a two dimensional space. If the luma, Y', drops by a reasonable amount from the mean luma, $\mu_{Y'}$, while the hue and saturation change very little, a shadow is detected. For this dissertation, an extension to this algorithm was made using our HS_vV space.

The first shadow condition is based on change in luma. When a background pixel goes into shadow, the luma of that pixel is expected to decrease. To be considered shadow, the luma of a pixel must drop below 95% of its average value.

The second condition is based on the chroma of the pixel. When a surface is shadowed, the colors of the pixels in that area are nearly unchanged. Therefore, to determine pixel color change, the mean location, μ_{HS_vV}, is determined for each pixel in HS_vV space and updated as part of the background model. As brightness decreases, the color of each pixel typically moves along an approximately linear path toward the origin in HS_vV space. Let $\vec{f}_{HS_vV}\,(x,y)$ be a pixel's current location and $\vec{\mu}_{HS_vV}\,(x,y)$ be its mean location. Similarity is computed herein as

$$d(x,y) = \frac{\vec{f}_{HS_vV}\,(x,y) \cdot \vec{\mu}_{HS_vV}\,(x,y)}{\left\|\vec{f}_{HS_vV}\,(x,y)\right\|\left\|\vec{\mu}_{HS_vV}\,(x,y)\right\|}.$$

If $d(x,y)$ is greater than .99, the pixel's color is assumed to be unchanged. The threshold of .99 was found empirically to work well in most lighting conditions and coincides with an eight degree angle in HS_vV space.

As mentioned previously, when luma is low, color information becomes unreliable. Therefore, the final condition is that the luma of each pixel must be above .2 to be considered to be a shadow. This also handles a special case of black areas moving through the scene. Without this condition, all new black areas would be discarded as shadow and never selected as foreground.

If the pixel's color is unchanged, its luma has decreased, and its luma is not too dark, the pixel is classified as being in shadow. The output of shadow detection is an image $L(x,y)$ defined as

$$L(x,y) = \begin{cases} 1 & d(x,y) > .99 \; and \; Y(x,y) < \; .95\mu_Y(x,y) \; and \; Y(x,y) > .2 \\ 0 & else \end{cases}$$

The color change detection image is

$$\varphi'_{h_{HS_v}}(x,y) = \varphi_{h_{HS_v}}(x,y) \; \wedge \left(1 - L(x,y)\right).$$

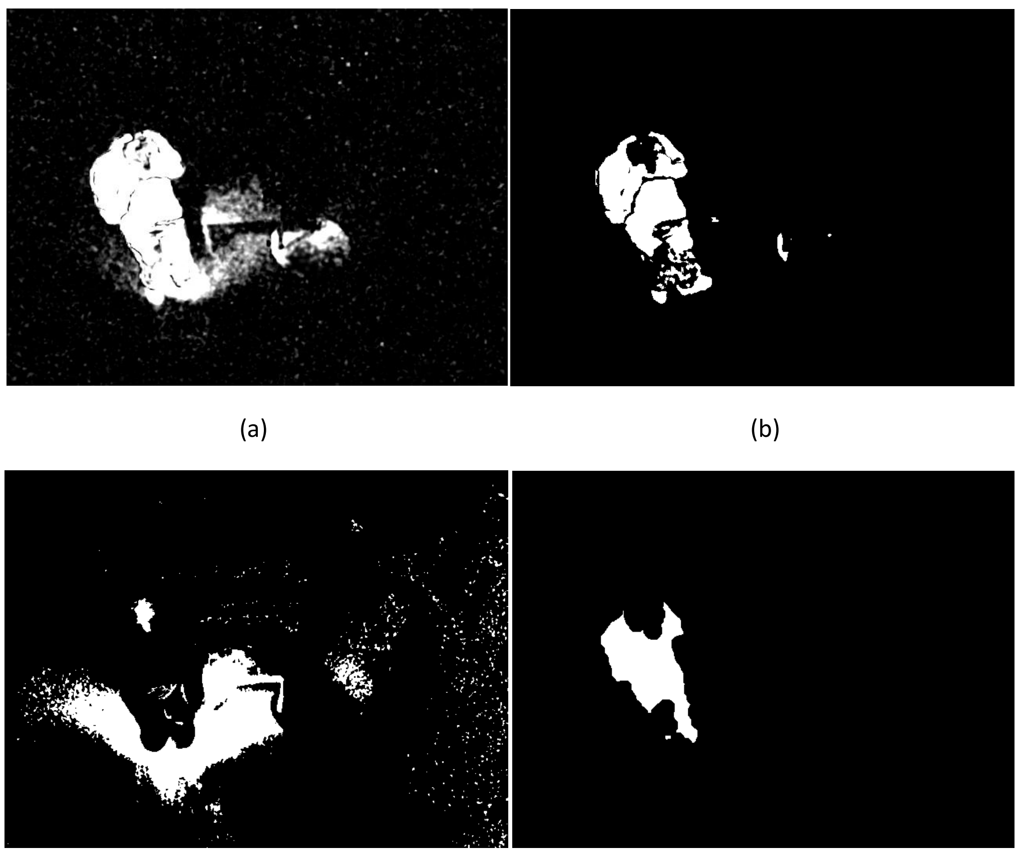
The final step is the fusion of texture and color difference which is performed using a union operator. This image, $F(x,y)$, represents the change detected from both texture and color,

$$F(x,y) = \left(\varphi_{C'_{brg}}(x,y) \vee \varphi_{HS_v}(x,y)\right).$$

Because the contour of the person often has segments missing from change detection, the fused image $F(x,y)$ is morphologically dilated by a circular kernel of radius $k_3 = 3$. Regions of pixels with value zero surrounded by pixels of value one are then filled with ones. The image is then morphologically eroded with a circular kernel of radius $k_6 = 6$, to eliminate noise points. One final morphological dilation with the k_3 kernel is performed to return the silhouettes to their proper size. The operation is defined as

$$O(x,y) = \left(\left(fill(F(x,y) \oplus k_3)\right) \ominus k_6\right) \oplus k_3.$$

The shadow removal process is shown visually in figure 3.4.



(a) (b)

(c) (d)

Fig. 3.4. Multiple stages of change detection using color descriptors. (a) The confidence of change detection in color space. Notice that shadows are detected as change. (b) Pixels that have registered a change above the threshold of 2. (c) Shadows in the scene. (d) The change with respect to color descriptors after removing shadows and performing morphological operations. Although the output does not match the silhouette closely, it fills in parts of the silhouette missed by the texture descriptor.

3.5 Background Update

The background, even in a constrained indoor environment, is not constant. Changes in lighting or manipulation of objects in the scene must be taken into account for a robust system. As mentioned earlier, well known algorithms such as Mixtures of Gaussians and Wallflower have been developed to handle background adaptation. Because this tracking system, designed to monitor elders, is a conglomeration of many smaller systems, algorithms with greater complexity are too computationally expensive to run in real time. It was therefore decided to update just a single mean and standard deviation for each feature dimension at each background pixel.

It is assumed that regions of change correspond to moved objects, or a person. Because the living quarters of the eldercare environment house only a single person, it is assumed that there will be at most only one person in the scene at any given time. Furthermore, our supposition is that the person is larger than any object moved in the scene. Therefore, the largest foreground region is recognized as the person. That area is then dilated by six pixels and is not used in the background update. All other pixels are used to update the background model.

An alpha update similar to that used in [24] is used to update the background model. It is too expensive to store and recompute the 32 means and standard deviations otherwise. The mean values are updated using a linear interpolation of the old value and new value.

$$\mu_{h_{C'_b}}(x,y,i) = (1-\alpha)\mu_{h_{C'_b}}(x,y,i) + \propto h_{C'_b}(x,y,i)$$

$$\mu_{h_{C'_r}}(x,y,i) = (1-\alpha)\mu_{h_{C'_r}}(x,y,i) + \propto h_{C'_r}(x,y,i)$$

$$\mu_{h_{C'_g}}(x,y,i) = (1-\alpha)\mu_{h_{C'_g}}(x,y,i) + \propto h_{C'_g}(x,y,i)$$

$$\mu_{h_{HS_v}}(x,y,i) = (1-\alpha)\mu_{h_{HS_v}}(x,y,i) + \propto h_{HS_v}(x,y,i)$$

Standard deviations are updated in a similar fashion using the absolute difference between the current value and the mean at each dimension.

$$\sigma_{h_{C'_b}}(x,y,i) = (1-\alpha)\sigma_{h_{C'_b}}(x,y,i) + \propto \left| h_{C'_b}(x,y,i) - \mu_{h_{C'_b}}(x,y,i) \right|$$

$$\sigma_{h_{C'_r}}(x,y,i) = (1-\alpha)\sigma_{h_{C'_r}}(x,y,i) + \propto \left| h_{C'_r}(x,y,i) - \mu_{h_{C'_r}}(x,y,i) \right|$$

$$\sigma_{h_{C'_g}}(x,y,i) = (1-\alpha)\sigma_{h_{C'_g}}(x,y,i) + \propto \left| h_{C'_g}(x,y,i) - \mu_{h_{C'_g}}(x,y,i) \right|$$

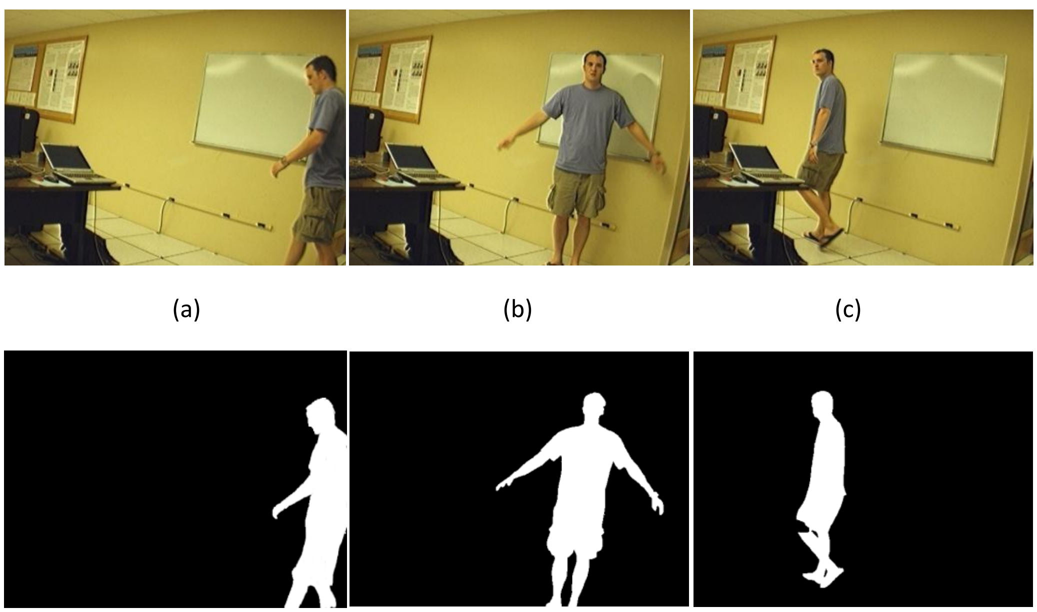
$$\sigma_{h_{HS_v}}(x,y,i) = (1-\alpha)\sigma_{h_{HS_v}}(x,y,i) + \propto \left| h_{HS_v}(x,y,i) - \mu_{h_{HS_v}}(x,y,i) \right|$$

Alpha determines the rate at which the system updates the background model. This parameter is associated with the frame rate of the camera and a user desired update rate for the system. We use an alpha of .01, for a system that captures images at a rate of 5 frames per second.

3.6 Results

In order to calculate the accuracy of the proposed foreground detection system, the extracted silhouettes are compared to three hand-segmented ground truth sequences. The backgrounds in these sequences include a range of colors and intensities that remain static throughout each sequence. The clothing worn by the subjects demonstrate the need to use both texture and color features for change detection. The beginning of each sequence contains only the static background with no humans. The subject then walks to a specified location in the room and performs a random action. The subject then begins walking again.

Test sequence one consists of 148 images of a primarily white and yellow colored background. The subject is wearing a solid blue shirt and khaki shorts. Figure 3.5 illustrates three frames from this sequence with the original image, the hand segmented silhouettes and the extracted silhouettes. In this sequence, the system correctly classified 99% of the hand segmented foreground. Also, only 2% of the pixels classified as foreground by the system were considered background by hand segmentation.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 3.5. A test sequence of images. Frames (a), (b) and (c) are three unprocessed images from the test sequence; (d), (e) and (f) are the hand-segmented silhouettes of the person; (g), (h) and (i) are the three silhouette images output from this system.

The second test sequence has the same background as the first, but with a different subject. This sequence contains 192 frames. The subject in this sequence wears a striped shirt and blue jeans. Figure 3.6 again shows three frames of the sequence. The system correctly classifies 98% of the hand segmented foreground, while only incorrectly classifying 1% of the background.

(a) (b) (c)

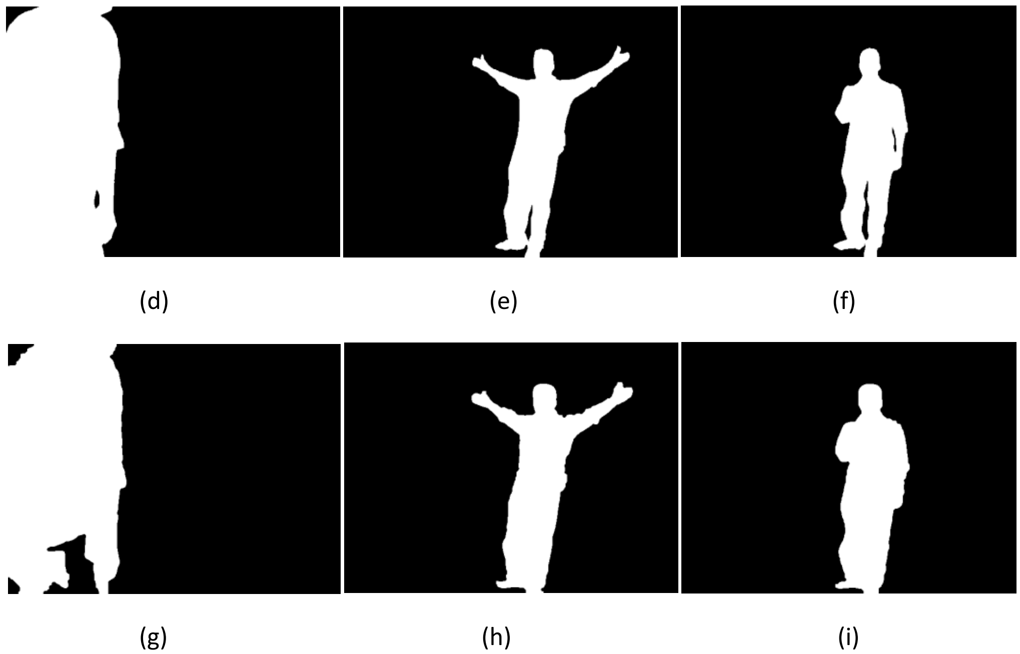


Fig. 3.6. A test sequence of images. Frames (a), (b) and (c) are three unprocessed images from the test sequence; (d), (e) and (f) are the hand-segmented silhouettes of the person; (g), (h) and (i) are the three silhouette images output from this system.

The final sequence has the same flat colored wall, but hard edges on the floor. The subject wears a white shirt and blue jeans. Figure 3.7 shows two representative frames from this sequence. Again, an impressive 99% of the hand segmented foreground is found, while only 1% of the foreground classification is incorrect.

(a) (b) (c)

(d) (e) (f)

Fig. 3.7. A test sequence of images. Frames (a) and (d) are two unprocessed images from the test sequence; (b) and (e) are the hand-segmented silhouettes of the person; (c) and (f) are the two silhouette images output from this system.

Over these three sequences, this system correctly classified 99% of the hand segmented foreground pixels. In addition, 98% of the areas classified as foreground by this system were segmented as foreground by a human. This level of accuracy makes the system suitable for many higher level intelligence processes such as human activity analysis.

The same three sequences were run through a Gaussian Mixture Model [24]. This system models the background as a mixture of Gaussians at each pixel and classifies change, i.e. foreground, when a new pixel does not reside within a user specified number of standard deviations from one of the K Gaussians. A new Gaussian distribution is built for each new pixel that is classified as foreground.

For testing, four models were used per pixel. Pixel values outside of three standard deviations from all Gaussians are classified as change. The GMM was tested separately for R'G'B' and Cb'Cr'Cg' color spaces. The best results were found using the Cb'Cr'Cg' values. As displayed in table 3.1, significantly higher accuracies were found using the system defined here to the GMM system. Figure 3.8 displays the results of the GMM and the system defined here.

Table 3.1. Accuracy of the system described in this dissertation and a Gaussian Mixture Model.

	Current System		GMM System (Cb'Cr'Cg')		GMM System (R'G'B')	
	Percentage of Foreground Pixels Found	Percentage of Pixels Incorrectly Classified as Foreground	Percentage of Foreground Pixels Found	Percentage of Pixels Incorrectly Classified as Foreground	Percentage of Foreground Pixels Found	Percentage of Pixels Incorrectly Classified as Foreground
Sequence 1	99%	2%	77%	8%	76%	9%
Sequence 2	98%	1%	85%	3%	80%	5%
Sequence 3	99%	1%	63%	1%	70%	2%
Total	99%	2%	81%	5%	78%	6%

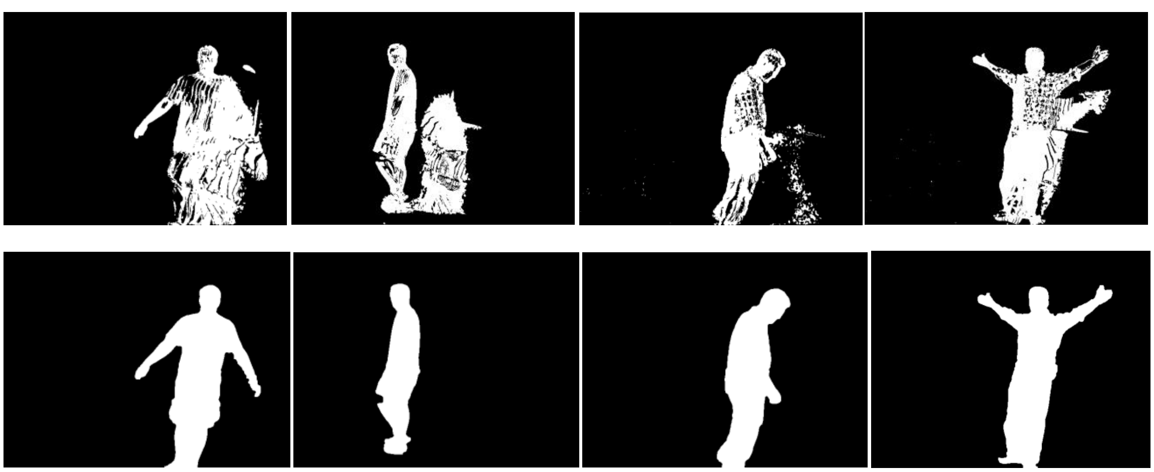


Fig. 3.8. Several images comparing the output of the Gaussian Mixture Model using Cb'Cr'Cg' color space, (top images), and those from the system defined in this paper, (bottom images).

3.7 Summary

In summary, an image space-based silhouette segmentation algorithm is presented above for the identification of single humans in indoor environments. The algorithm is real-time and the major contribution is the identification and fusion of color and texture features. It is also demonstrated that change detection in image space requires the algorithmic identification and removal of shadows. This, combined with post-processing, has resulted in a system shown to have better performance than the standard research approach, i.e. GMM, eigenbackgrounds, etc. The resulting image silhouettes make the next step in human activity analysis, voxel person construction, possible.

Volume Element Space

4.1 Introduction

This chapter outlines the modeling, construction, and refinement of objects in a discretized volume element (voxel) space. A voxel space is the quantization of three-dimensional space into a finite set of rectangular volumes, much like the notion of a picture element (pixel) in two-dimensional space. Voxels are specified according to their center location, $\vec{v}_j = \langle x_j, y_j, z_j \rangle^T$, and width set $\{\omega_1, \omega_2, \omega_3\}$. Generally, the width is the same for all dimensions, i.e. $\omega_1 = \omega_2 = \omega_3$, resulting in a cube shape. Figure 4.1 shows an example 8x4x7 voxel space of width 1x1x1.

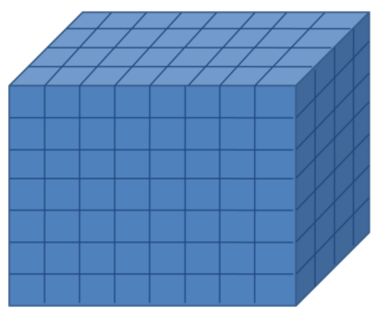


Fig. 4.1. Three-dimensional 8x4x7 volume element (voxel) space.

4.2.1 Construction – Intersection of Back-Projected Image Silhouettes

Reconstruction of three-dimensional objects, both solid representations and hulls, from two-dimensional images through back-projection is not a new concept. Object reconstruction has been studied within computer graphics, computer vision, biomedicine, and even in a variety of forms in the activity analysis domain [112][113][114][115][116]. What separates this work from most, besides the use of silhouettes for back-projection, is the way in which voxel person is used and how its shape is refined using knowledge about the environment. A relatively low object resolution is typically used, for computational efficiency, and the object is only used to obtain features for activity analysis. The object is not explicitly tracked, detailed segmentation of object regions (hands, head, torso, etc…) is not attempted, and the goal is not to build a highly detailed surface or solid representation. Advances have been made in each respective area, but no approach to date is either fast enough or mature enough to be included in a real-world system that runs unsupervised for long time periods. The resident's privacy is also further protected by not building a detailed object representation, which is difficult to obtain using silhouettes produced by a robust change detection system. A wide range of activities, especially in an eldercare domain, and for fall detection in particular, do not require high detail for reasoning about activity that involves the movement of the entire body.

After silhouettes are individually extracted from each camera in a scene, a three-dimensional representation of the human is constructed in voxel space, called voxel person. A voxel is defined here as a non-overlapping cube. The set of voxels belonging to voxel person at time t is

$$V_t = \{\vec{v}_{t,1}, \vec{v}_{t,2}, \dots, \vec{v}_{t,P}\},$$

where the center of the jth voxel at time t is $\vec{v}_{t,j} = \langle x_j, y_j, z_j \rangle^T$. The capture time for each camera is recorded and the silhouettes, one from each camera, that are the closest in time are used to build V_t. The construction of voxel person from a single camera is the planar extension of the silhouette along the direction of the camera viewing angle. Voxels in the monitored space that are intersected by this planar extension are identified. The projection procedure used in this work, illustrated in figures 4.2.a and 4.2.b, involves using the camera's intrinsic parameters to estimate pixel rays, and these rays are tested for intersection with voxels. An alternative projection procedure, the use of view volumes, truncated pyramids, is illustrated in figure 4.2.c. This alternative procedure results in more voxel intersections, which in return increases the storage requirements and the computational time related to performing the intersection tests, a standard graphics clipping operation. The selection of projection procedure depends on factors such as the image resolution, camera horizontal and vertical viewing angles, voxel size, size of the monitored space, and the desired accuracy.

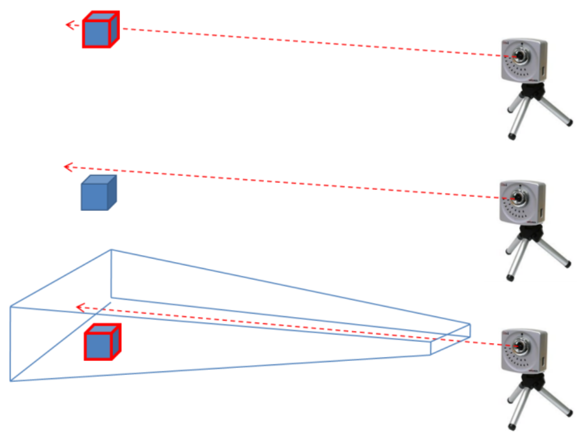


Fig. 4.2. Illustration of camera pixel view ray testing (a)(b) versus pixel view volume-based testing (c). The red dotted lines are the camera pixel view vectors and the blue truncated pyramid is the pixel view volume. Pixel ray testing results in fewer overall voxels.

Bouguet made accessible a freely downloadable Matlab package for the calibration of intrinsic and extrinsic camera parameters [117]. His work is based on that of Zhang [118][119]. A checkerboard pattern, where the size of the checker pattern is known, is attached to a rigid body object. The object is moved around a cameras field of view under different rotation conditions. The end product of calibration is a set of pixel rays, specified initially in image space, where each ray is of unit length. These rays are necessary to determine which voxels are viewable according to a particular pixel, the information required for back-projection. The ray for pixel (m, n) is $\vec{r}_{m,n} = \langle x_{m,n}, y_{m,n}, z_{m,n}\rangle^T$. The center position of camera c, $1 \leq c \leq C$, $\overrightarrow{cpos}_c = \langle x_c, y_c, z_c\rangle^T$, is also recorded. Markers are then placed in the environment and their positions are recorded with respect to some fixed location (the assumed origin of the global coordinate system, such as a corner of a room). The extrinsic camera parameters, the unit length vectors $\{\overrightarrow{right}_c, \overrightarrow{center}_c, \overrightarrow{up}_c\}$, are estimated using the rays, the location of the markers in image space, and the position of the markers in the global coordinate system [117]. These vectors respectively represent the right, forward/center, and up directions that form the orthonormal basis

$$ob_c = \begin{bmatrix} right_{c,x} & center_{c,x} & up_{c,x} & 0 \\ right_{c,y} & center_{c,y} & up_{c,y} & 0 \\ right_{c,z} & center_{c,z} & up_{c,z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

This matrix is used to transform between local and global coordinate systems. The ray $\overrightarrow{Ray}_{m,n}$, which is $\vec{r}_{m,n}$ transformed into the global coordinate space, is

$$\overrightarrow{Ray}_{c,m,n} = ob_i * \langle \vec{r}_{m,n}\ 1 \rangle^{T}.$$

In the following, $\vec{R}_{c,m,n}$ is used to represent the first three components, $\langle x\ y\ z \rangle^T$, from $\overrightarrow{Ray}_{c,m,n}$.

In the brute force approach, each voxel in the environment is tested for intersection with each ray. To simplify the intersection tests, voxels are represented as a sphere with radius

$$\sqrt{(3 * (G/2)^2)}.$$

The radius, G, is set so that the entire voxel fits into the sphere. This results in a few more intersections than a ray-cube approach. However, the goal is not to build a perfect representation of voxel person, but rather an approximation of its shape in real-time that is of use for activity analysis. The features extracted for fall detection do not require near perfect voxel person shape and detail.

A vector is then created between $\vec{v}_{t,j}$ and $\overrightarrow{cpos}_c$,

$$\vec{b}_j = (\vec{v}_{t,j} - \overrightarrow{cpos}_c).$$

The ray-sphere intersection test for the ray emitted from pixel (m, n) in camera c, $\vec{R}_{c,m,n}$, and $\vec{b}_j$, which represents voxel j from the standpoint of camera c, is based on the following distance value

$$D = \left\| \left(\left(\frac{\vec{R}_{c,m,n}}{\|\vec{R}_{c,m,n}\|} \right) * \left(\frac{(\vec{R}_{c,m,n} \cdot \vec{b}_j)}{(\|\vec{R}_{c,m,n}\| \|\vec{b}_j\|)} \right) * \|\vec{b}_j\| \right) - \vec{b}_j \right\|$$

$$= \left\| \left(\left(\frac{\vec{R}_{c,m,n}}{\|\vec{R}_{c,m,n}\|} \right) * \left(\frac{(\vec{R}_{c,m,n} \cdot \vec{b}_j)}{\|\vec{R}_{c,m,n}\|} \right) \right) - \vec{b}_j \right\|$$

$$= \|(\vec{R}_{c,m,n} * (\vec{R}_{c,m,n} \cdot \vec{b}_j)) - \vec{b}_j\|$$

since $\|\vec{R}_{c,m,n}\| = 1$.

If $D < G$, then $\vec{v}_{t,j}$ is added to camera c's pixel (m, n) list. This metric is illustrated in figure 4.3 and the back-projection of a single image pixel is shown in figure 4.4.

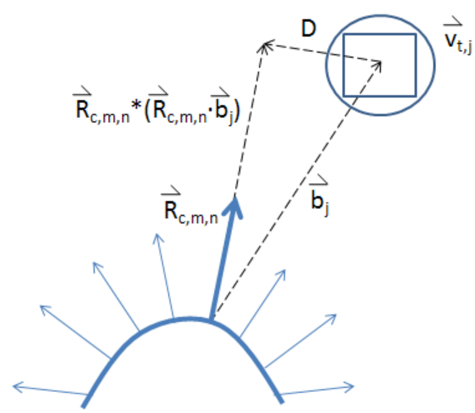


Fig. 4.3. Illustration of the ray-sphere intersection test. The $\vec{v}_{t,j}$ voxel would be rejected by $\vec{R}_{c,m,n}$.

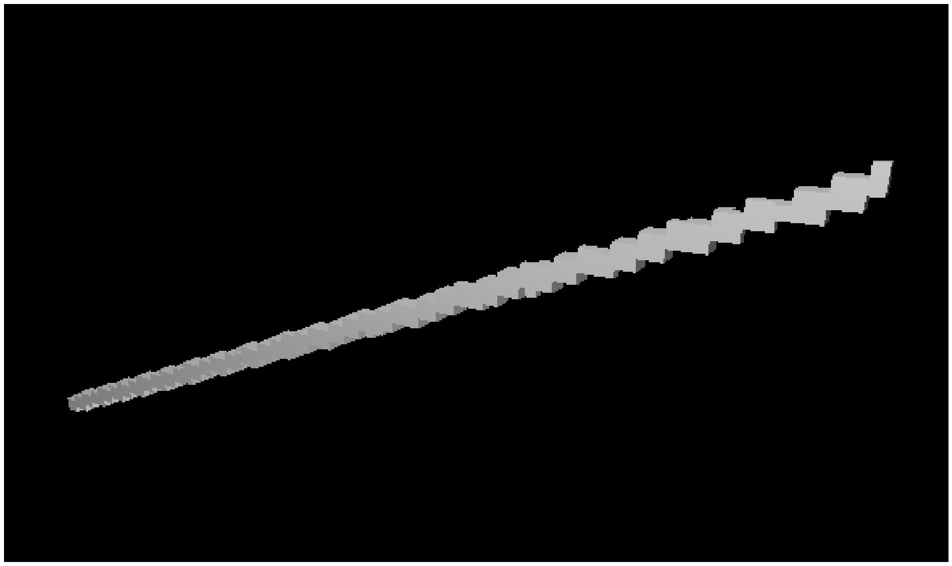


Fig. 4.4. Illustration of the back-projection of a single pixel ray vector into voxel space.

Voxel person, according to camera c at time t is V_t^i, and its cardinality, $|V_t^i|$, is P_i. The planar extensions of voxel person from multiple cameras, $\{V_t^1, \dots, V_t^C\}$, are combined using an operation, such as intersection,

$$V_t' = \bigwedge_{i=1}^{C} V_t^i,$$

to assemble a more accurate object representation. Illustration of voxel person construction from two cameras is shown in figures 4.5 and 4.6.

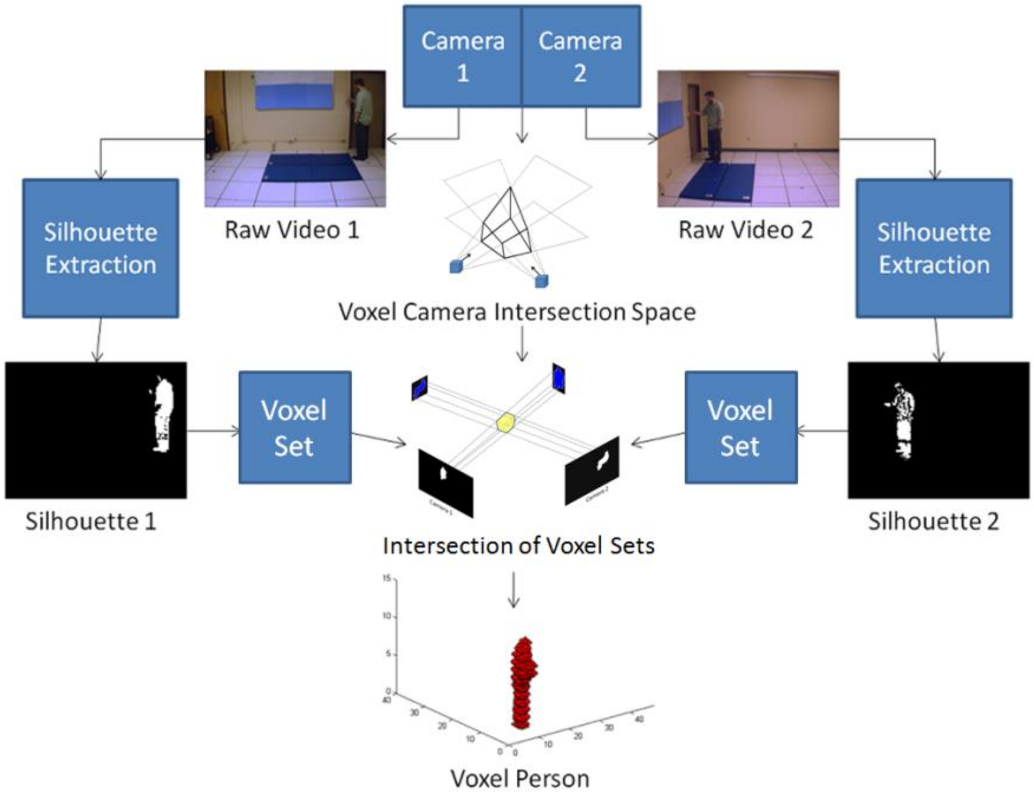
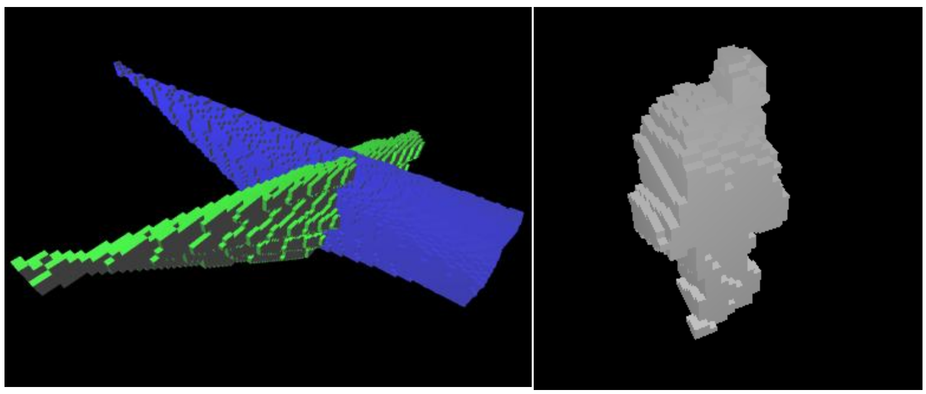


Fig. 4.5. Voxel person construction. Cameras capture the raw video from different viewpoints, silhouette extraction is performed for each camera, voxel sets are calculated from the silhouettes for each camera, and the voxel sets are intersected to calculate voxel person.

(a) (b)

(c) (d)



(e) (f)

Fig. 4.6. Raw images, (b) and (d), silhouettes, (a) and (c). Image (e) is the back-projection of the silhouettes, shown as blue and green voxel sets. Image (f) is voxel person, i.e. the intersection of the two camera silhouette pixel-voxel list sets.

For each pixel, a set of voxels that its viewing ray intersects are identified offline, $L_{c,(n,m),t}$ for pixel (n, m), camera c, and time step t. This is the enabling step that makes voxel person construction real-time. Building voxel person reduces to an indexing procedure. The voxel-pixel test is a procedure that only has to be computed one time when the cameras are positioned in the room. If computational complexity of voxel-pixel correspondence is of concern, possibly because the cameras could be moved frequently or the possible voxel space is very large, spatial partitioning of voxel space, with either an octree or binary spatial partition tree, can be used to speed up voxel-pixel set construction.

4.2.2 Construction – Visible Shell

The intersection of individual camera back-projected voxel sets is one account of the object. It is shown in the next section that this object is prone to back-projection error. There is another way to aggregate the individual camera data to help reduce final object error. The visible shell set for camera c at time t, $S_{t,c}$, is the subset of voxels from V_t' that are directly viewable by camera c, i.e. not occluded by any other voxels in V_t'. In the end, each individual cameras visible shell object is aggregated to generate the complete object visible shell,

$$S_t' = \bigvee_{i=1}^{C} S_{t,i}.$$

Algorithm 4.1 is used to compute S_t', while figures 4.7.a and 4.7.b illustrate the visible shell.

ALGORITHM 4.1. Visible Voxel Shell Construction

---

1. Initialize sets S_t^1, …, S_t^C to empty

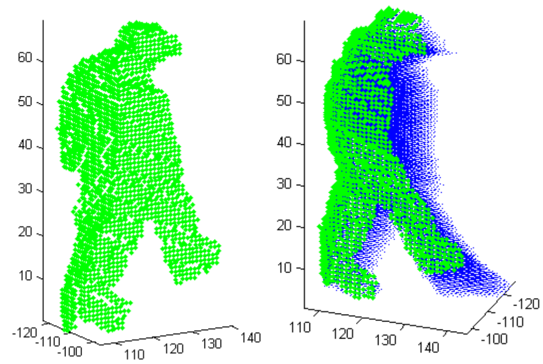
2. Compute the full intersection-based voxel person, i.e. $V_t' = \wedge_{c=1}^{C} V_t^c$

3. For each camera $(1 \leq c \leq C)$

a. For each pixel in the silhouette set

i. Find the closest voxel, $\vec{v}_j'$, from $L_{c,(n,m),t}$ in V_t', and add $\vec{v}_j'$ to S_t^c

4. $S_t' = \vee_{c=1}^{C} S_t^c$.



(a) (b)

Fig. 4.7. (a) Visible shell, shown in green, and (b) visible shell along with the back-projected intersected object, shown in blue.

4.2.3 Construction – Error

Object construction error due to change detection failure is not discussed in this section. Instead, focus is placed on error resulting from the process of back-projection and the intersection of back-projected sets. Different fundamental types of back-projection-based error exist, namely visible and non-visible object error, void space error, and occlusion error (self as well as other environment objects). It is important to know about and study the different error types in order to reduce their effect in post-processing or to factor them somehow into the design of features for a voxel object.

The first topic is visible and non-visible object back-projection error. Informally, this error is incorrectly identified volume attached to an object. This type of error is color coded as yellow (visible error) and orange (non-visible error) in figure 4.8. Visible error is incorrectly inferred object volume located in front, between camera and object, of the visible surface of an object. It is the result of not knowing the true depth of visible surface locations. Stereo vision can be used to minimize this type of error [69]. Non-visible error is incorrectly inferred volume that is occluded by the object.

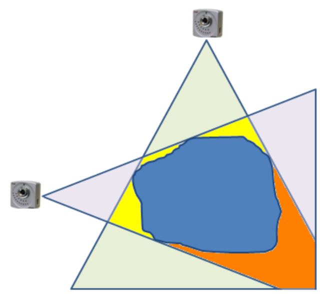


Fig. 4.8. Visible and non-visible object error. Yellow, blue, and orange areas are the silhouette back-projected object. Blue is actual object, yellow is visible error, and orange is non-visible error.

Void space error is the incorrect identification of islands not connected to any true object. An island is a set of pixels, or voxels respectively, that form a connected group [107]. This type of error occurs when there are multiple islands in image space, e.g. multiple objects or change detection failure. In figure 4.9, void space error islands are color coded orange.

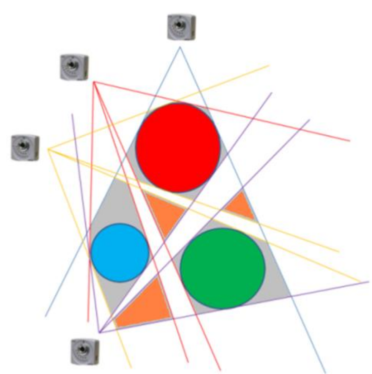


Fig. 4.9. Example scene containing multiple objects. Red, blue, and green circles are true objects. Orange islands are void space error. Gray regions are visible and non-visible object error.

The last error category is due to occlusion. In the case of non-self occlusion, regions are not back-projected and parts of the object are never created. In the extreme case of total occlusion in at least one camera, which can be detected by the absence of the silhouette in at least one camera, the intersected object is never created. With respect to self-occlusion, incorrect volume is filled in, which is shown in figure 4.10. Silhouette-based back-projection operates most ideally on convex objects.

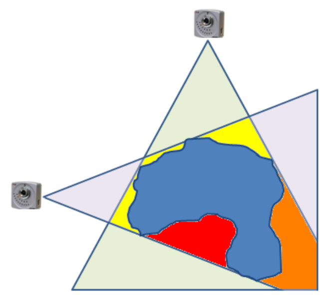
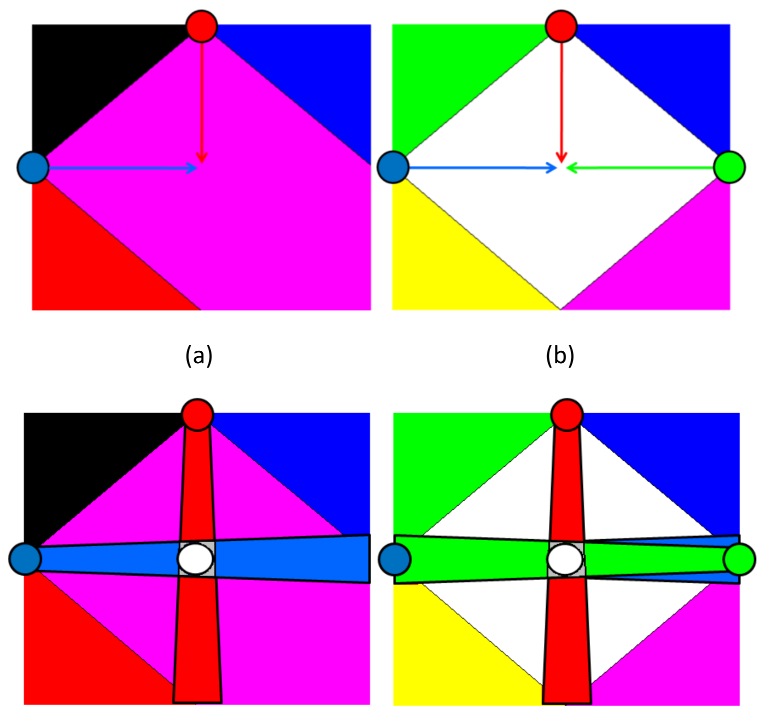


Fig. 4.10. Self-occlusion back-projection error. Yellow area is visible error, orange is non-visible error, blue is the true object, and red is self-occluded error.

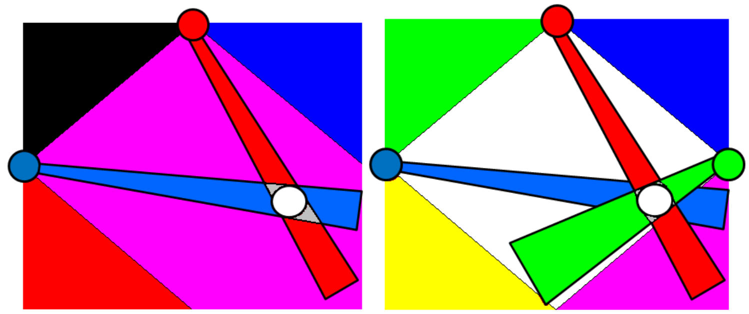
4.2.4 Construction – Viewing Conditions

In some applications it is possible to create high quality back-projected objects using a device such as a turntable, i.e. rotate the object, or camera, and capture a large number of images from different perspectives. However, for the monitoring of humans with cameras located at fixed locations a question arises regarding how many cameras to use and what their viewing conditions should be. Unfortunately, it is generally the case that only a few cameras are available, e.g. two. This is typically because one can only financially justify, physically install, or technologically process the data for a low number of cameras in real-time. Camera placement should be based on maximizing the viewing conditions, i.e. what the activities looks like from the perspective of the cameras, and on viewing volume available. Unfortunately, one must also minimize the resulting potential back-projection error. Ideal construction for this work breaks down to orthogonality of joint camera viewing. Another tradeoff occurs in the fact that the further back the cameras are installed the more viewing region but the larger the error. Ultimately, the configuration depends on the size of the space and of what regions need to be viewed and under what viewing conditions for activity analysis. Figure 4.11 illustrates many of the problems at hand for a common two versus three camera configuration.



(a) (b)

(c) (d)



(e) (f)

Fig. 4.11. Non-white circles are cameras and each is respectively color coded. White circles and gray regions are the back-projected object. In (a), a two camera setup, purple is the joint monitored space. In (b), a three camera setup, white is the joint monitored space. As the number of cameras increase, the joint viewable space decreases. Images (c), (d), (e), and (f) show camera color-coded back-projected view volumes. In the case that an object is in the middle of the viewing region, (c) and (d), the two and three camera setups are very similar. However, when the object is placed at another location in joint viewable space, (f), the superiority of the three camera setup is apparent.

For a two camera setup, viewing can be decomposed and analyzed according to the dot product of camera center pixel view vectors. Figure 4.12 illustrates $\vec{a} \cdot \vec{b} = 0$, $\vec{a} \cdot \vec{b} = -1$, and the case of when $\vec{a} \cdot \vec{b}$ approaches a value of one.

Fig. 4.12. Quality of construction based on orthogonality of camera center pixel view vectors.

4.2.5 Construction – Blanketed Set

Non-visible object error can be essentially eliminated at the expense of a fraction of true object volume for a predominantly downward viewing scene, i.e. cameras installed on the ceilings angled downward. For this type of viewing, the assumption is that the majority of the head and shoulders are observed most of the time. The blanketed set, U_t, is a subset of the intersected voxel person, V_t', *under* the visible shell, S_t'. Algorithm 4.2 is used to compute U_t. In the following algorithm, g_k and v_k are the kth vector components in $\vec{g} \in S_t'$ and $\vec{v} \in V_t'$, where k is the world up direction.

ALGORITHM 4.2. Blanketed Set Construction

---

1. Calculate voxel person, $V_t' = \bigwedge_{c=1}^{C} V_t^c$*, and the visible shell,* $S_t' = \bigvee_{c=1}^{C} S_t^c$

2. Calculate $U_t = \left\{\vec{v}_j' \in V_t' \mid \vec{v}_j' \in \Omega(S_t')\right\}$*, where* $\Omega(S_t') = \{\vec{v} \mid \exists \vec{g} \in S_t' \ \ s.t. \ g_k \geq v_k\}$.

In algorithm 4.2, $g_k \geq v_k$ is a check for existence of a visible shell voxel *above* $\vec{v}$ (with respect to world up direction), and $\Omega(S_t')$ is the Umbra of V_t'. The advantage of U_t is the drastic removal of non-visible error. Downfalls include the continued inclusion of visible error and the loss of some of the true object. Examples are shown in figures 4.13 and 4.14.

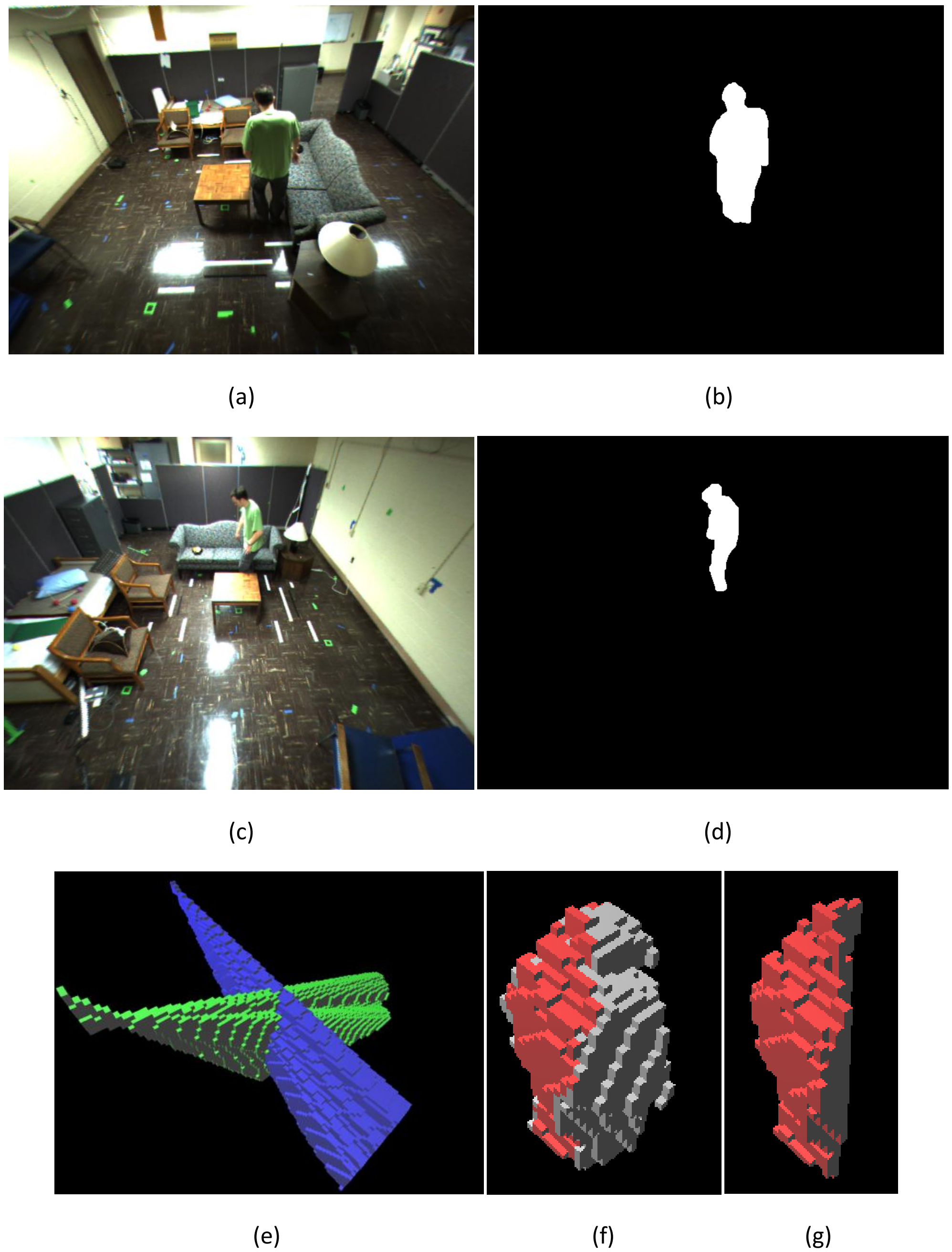
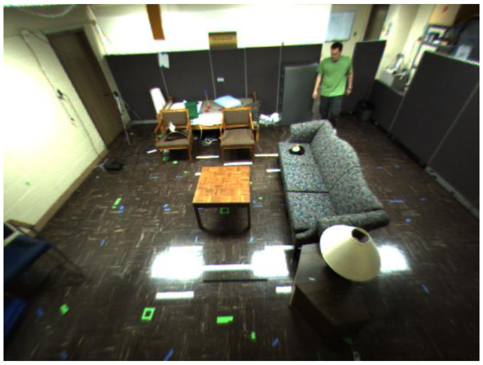
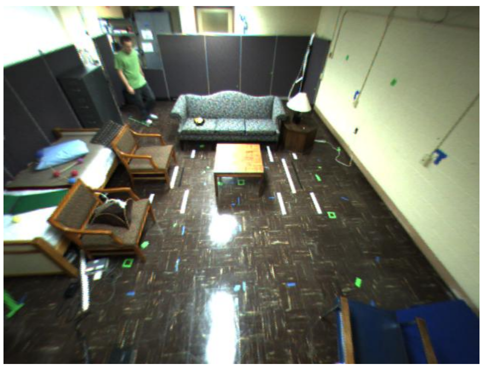


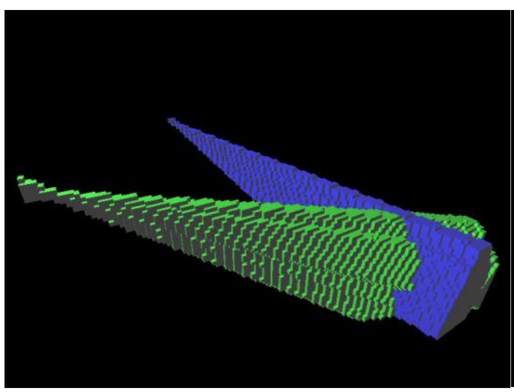
Fig. 4.13. Blanketed set for an ideal case of near orthogonal viewing construction of a human. Images (a) and (c) are the raw images, (b) and (d) are the silhouettes, (e) are the silhouette back-projections, (f) is the visible shell (red) and intersected set (gray), and (g) is the blanketed set. Take note of the loss of some true object volume. However, features, such as orientation, centroid, and height are still valid.

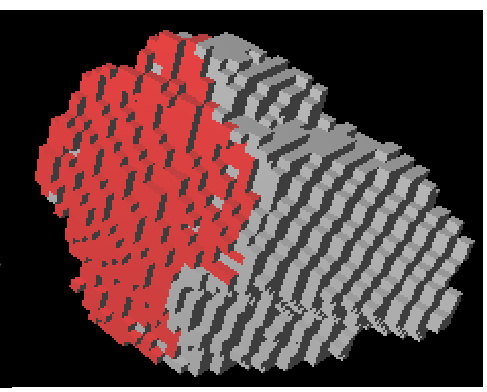


(a) (b)

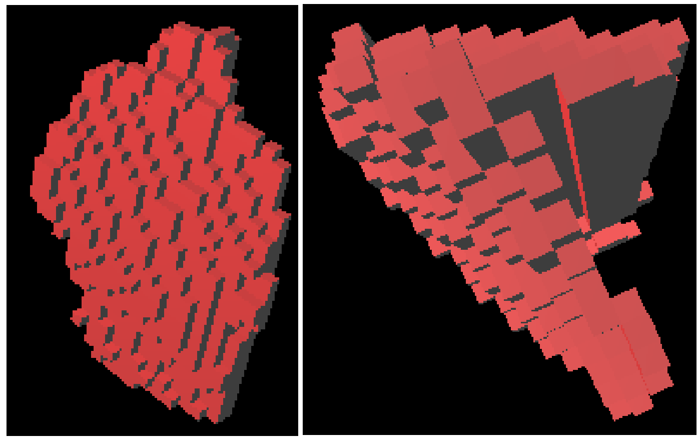


(c) (d)





(e) (f)



(g) (h)

Fig. 4.14. Blanketed set for a non-ideal case of viewing construction of a human. Images (a) and (c) are the raw images, (b) and (d) are the silhouettes, (e) are the silhouette back-projections, (f) is the visible shell (red) and intersected set (gray), and (g) and (h) are the blanketed set. Take note that the non-visible error in (f) is drastic. While the blanketed set, shown from two different angles, (g) and (h), has lost some true object volume, the resulting features, e.g. centroid, orientation, and height, are much more accurate.

4.3 Quality of Construction

The above sections provide a qualitative account and rules of thumb for the quality of construction. The following section is a quantitative account. Quality varies over the space with respect to the object's location, the installation locations of the cameras and their respective orientations. Quantitatively knowing the construction quality is of importance for at least two reasons. First, knowledge about the ability to properly construct the human voxel object affects the features calculated, and subsequently our decisions regarding activity inferred using those features. Secondly, there may be locations in a space for which it is incredibly difficult to properly construct a sufficient voxel object for tracking given a camera configuration. In many instances, suppression of object construction in such areas is of value.

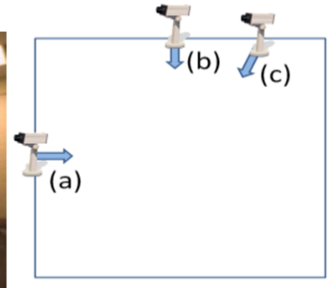
As already stressed, the ideal case of construction occurs when intersecting view vectors are orthogonal. This means that each individual voxel has a different construction quality. The quality of construction for voxel $\vec{v}'_{(i,j,k)}$ is

$$Q_{(i,j,k)} = 1 - \underset{\substack{\vec{C}_g,\vec{C}_h \\ g\neq h}}{\text{maximum}} \left| \left(\vec{v}'_{(i,j,k)} - \vec{C}_g\right)^T \left(\vec{v}'_{(i,j,k)} - \vec{C}_h\right) \right|,$$

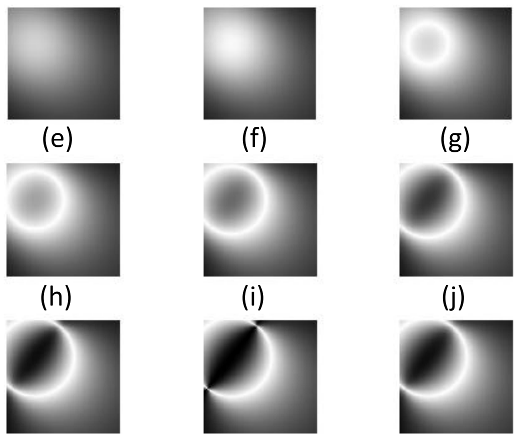
where $\vec{C}_g$ and $\vec{C}_h$ are the center locations of cameras g and h. Thus, the best case per voxel per camera pair with respect to orthogonality is considered. If intrinsic camera parameters have been estimated, as in [117], then the pixel rays that intersect voxel $\vec{v}'_{(i,j,l)}$ may be used in place of $\vec{v}'_{(i,j,k)} - \vec{C}$.

To determine environment construction quality, the entire scene is converted into a voxel space. However, this time the resolution is much lower. In this work, it was empirically determined that a sampling of one voxel every foot was a sufficient resolution for quality assessment. Figure 4.15 shows nine horizontal slices, x-y planes for a varying z, of the sampled voxel space and their respective qualities.

(a) (b)



(c) (d)



(e) (f) (g)

(h) (i) (j)

(k) (l) (m)

Fig. 4.15. (e)-(m) is nine horizontal (x-y plane) slices of voxel object construction quality for the camera configuration {(a),(b)} in (d). Brighter values in (e)-(m) represent higher quality. Map (e) is at a height of 1 foot, and each consecutive map is 1 foot higher in the z dimension (world up direction).

The next matter is determination of the quality of construction for an object. One can compute the mentioned quality measure for each voxel and make a decision based on the set of all memberships, or the object can be summarized, according to its centroid, height, or some other measurement, and that point estimate can be used to make a decision for the entire object. The latter is selected here. It has been decided that if any part of the object has too low of a construction quality then the entire object will be removed. The monitored area is first converted into a low voxel resolution space, i.e. figure 4.15, then the height domain is collapsed and a single voxel plane in the x-y dimension is produced. The voxel qualities are combined in the height domain using a t-norm, e.g.

$$Q_{(i,j)} = \underset{k}{\text{minimum}}\, Q_{(i,j,k)}.$$

This pessimistically selects the worst construction case per element in the final x-y plane voxel set. Figure 4.16 shows the final x-y voxel plane of combined confidences for figure 4.15.

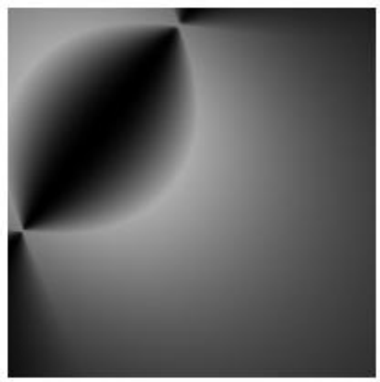
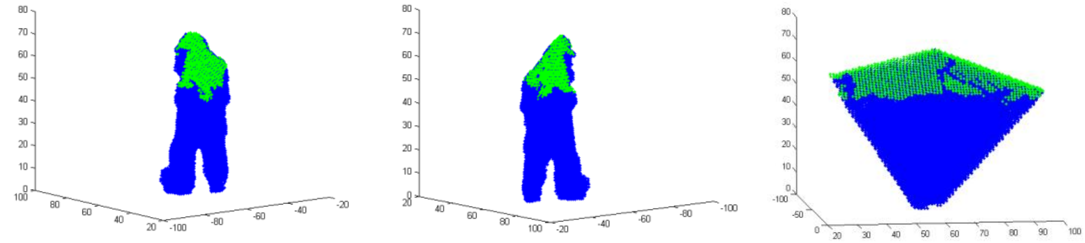


Fig. 4.16. T-norm produced single x-y voxel plane for figure 4.15.

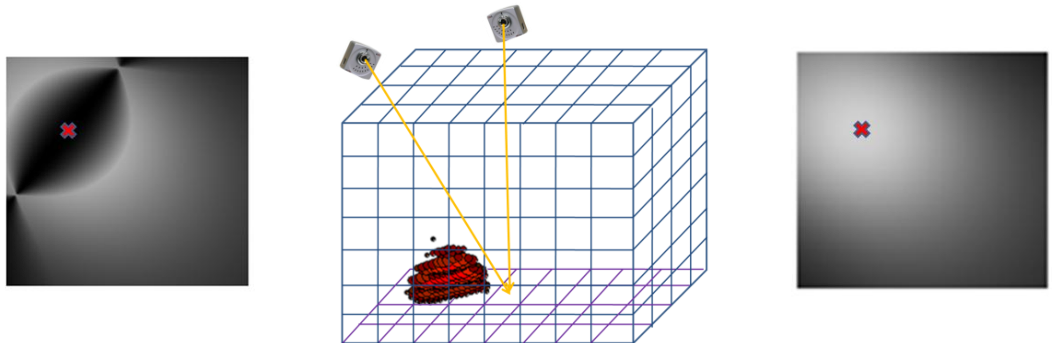
Next, the voxel object's centroid is mapped to the closest (i,j) element in the quality map plane and the respective quality value, $\phi_t = Q_{(i,j)}$, is retrieved. If $\phi_t < \delta_1$, then the object is not tracked at that moment because it could not be adequately constructed. Parameter δ_1 is not specific to any one single environment. Instead, it is a global value relating two viewing rays to back-projection construction quality. Figure 4.17 shows an example construction in which the quality value is a low value of 0.05.



(a) (b) (c)

Fig. 4.17. Example of low object construction quality, value of 0.05, for cameras {(a),(b)} in figure 4.15.d. The object is approximately half way along the line connecting cameras {(a),(b)}. Thus, a good majority of the object is constructed using view ray vectors pointing roughly towards each other. Green is the visible shell and blue is the intersected object. The object appears to be constructed properly when observed with respect to the two individual cameras views, (a) and (b). However, when the object is viewed from a different location in space, (c), it is apparent that the object is poorly constructed.

On a final note, the above procedure can be improved if one does not always just use the entire set of voxel x-y quality planes, i.e. all nine planes in figure 4.15, but instead a subset based on the current height of the object. Specifically, the subset of voxel planes from the ground up to the current maximum height of the voxel object. The rational is as follows. For the case of predominantly downward viewing cameras, the higher the voxel in the world up direction the lower the quality. Thus, for very low quality areas in figure 4.16, when an object is upright there is difficulty in properly constructing the upper parts of the object, i.e. 4.17.c. However, if the human is on the ground, e.g. has fallen, it is possible that the object can be constructed with sufficient quality if one only considers the subset of quality planes from the ground up to the current height of the voxel object. This concept is illustrated in figure 4.18.



(a) (b) (c)

Fig. 4.18. Dynamic selection of construction quality planes based on an objects current height for the case of predominantly downward viewing cameras. (a) is the t-norm produced plane for all nine planes in figure 4.15. (c) is the t-norm produced plane for only the bottom three planes, i.e. {(e),(g),(f)}. Red crosses are the location of the object in (b). (a) shows that the object has a very low quality and should be ignored, i.e. figure 4.17.c, while (c) shows that the object can be constructed with sufficient quality.

4.4 Post-Processing

Error in silhouette extraction translates into error in voxel person construction. However, voxel person construction can actually assist in eliminating some types of silhouette-based error. Artifacts observed in one camera that do not intersect artifacts, or valid projection regions, in the other cameras are automatically removed. For example, specular highlights are camera view dependent. Thus, for two different cameras, many specular highlights are automatically removed through intersection, or result in void space error, which can be eliminated using the visible shell.

One of the motivating factors for moving from a two dimensional to a three dimensional representation involves the ability to model the environment. Knowledge about the world allows for the identification of voxels that correspond to walls, floor, ceiling, or other static objects or surfaces. These voxels are removed because they do not provide any significant contribution to voxel person's shape. By removing these volumes, one effectively removes areas upon which many shadows are projected.

After objects are created, they undergo three dimensional binary morphological reconstruction [107]. Set operations used below are erosion, $\ominus$, dilation, $\oplus$, and reconstruction, $\otimes$. The translation of voxel set E, e.g. U_t, by $\vec{x}$ is

$$M_{E,\vec{x}} = \left\{\vec{v}_j + \vec{x} | \vec{v}_j \in E\right\}.$$

The erosion of E by a kernel K, is

$$E_E = E \ominus K = \left\{\vec{v}_j \in E | M_{\vec{v}_j,\vec{x}} \in E, \forall \vec{x} \in K\right\},$$

and the dilation of E by K, is

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

Algorithm 4.3 is the morphological reconstruction of E by kernels (K_1, K_2), $E_R = E \otimes (K_1, K_2)$.

ALGORITHM 4.3. Morphological reconstruction

---

1. $C = E \ominus K_1$

2. REPEAT

  a. $A = C$

  b. $B = C \oplus K_2$

  c. $C = \{\vec{v}_j | \vec{v}_j \in E \text{ and } \vec{v}_j \in B\}$

3. UNTIL $C = A$ *or Maximum Number of Iterations.*

The final post-processing step used here is small island removal. Connected components technique [107] is used to discover voxel islands, and islands below a user defined threshold are removed.

4.5 Feature Extraction

In order to use voxel person for activity analysis, features must first be extracted at each time step. There are two primary categories of features, global, or total body features (e.g. mean, height, etc), and local, those specific to a region/subset of the body (e.g. the feet, legs, hands, etc). The simplicity of global features are that they are directly derivable from V_t'. Global features also tend to be more robust than local features. They are generally less susceptible to factors such as voxel resolution and back-projection error. Lastly, the difficulty in measuring local features is when and how to identify the appropriate subset of voxels from V_t', e.g. the voxel leg set.

The following spatial and temporal features were extracted with fall recognition in mind. The spatial features include voxel person's centroid, eigen-based maximum height, eigen-based minimum height, and the similarity of voxel person's primary orientation with the ground plane normal. The centroid of voxel person at time t, $\vec{c}_t$, is

$$\vec{c}_t = \left(\frac{1}{P}\right)\sum_{j=1}^{P} \vec{v}'_{t,j}.$$

Calculation of voxel person's height from the maximum, in the world up direction, observed voxel is not robust. Silhouette extraction is often incorrect. Silhouette error in multiple cameras leads to back-projection error, which in return leads to erroneous height values for a trivial maximum calculation. Instead, the features eigen-based maximum height and minimum are calculated. The covariance matrix, used to find the eigen information, is

$$Cov_t = \left(\frac{1}{P-1}\right)\sum_{j=1}^{P}(\vec{v}'_{t,j} - \vec{c}_t)(\vec{v}'_{t,j} - \vec{c}_t)^t.$$

The eigenvectors, $\overrightarrow{eigvec}_{t,k}$, where $k = \{1,2,3\}$, are scaled by their respective eigenvalues, $eigval_{t,k}$, and are added to the voxel person centroid, i.e.

$$\overrightarrow{eigheight}_{t,k} = \vec{c}_t + \left(2\sqrt{eigval_{t,k}}\right)\overrightarrow{eigvec}_{t,k}.$$

It is assumed that the eigenvalues, and their respective eigenvectors, are sorted in decreasing order, i.e. $eigval_{t,1} \geq eigval_{t,2} \geq eigval_{t,3}$. A problem arises in the direct use of the eigenvectors for subsequent calculations. For example, $\overrightarrow{eigvec}_{t,1}$ is one direction of maximum variance while $(-1)\overrightarrow{eigvec}_{t,1}$ is the other. Thus, both $\overrightarrow{eigvec}_{t,k}$ and $(-1)\overrightarrow{eigvec}_{t,k}$ need be considered. For each $k = \{1,2,3\}$ eigenvector, a corresponding $\overrightarrow{eigheight}_{t,k+3}$ is generated,

$$\overrightarrow{eigheight}_{t,k+3} = \vec{c}_t + \left(2\sqrt{eigval_{t,k}}\right)(-1)\overrightarrow{eigvec}_{t,k}.$$

The maximum world up direction value from the $\overrightarrow{eigheight}_{t,k}$ set, a total of six vectors now, is recorded, $maxeigheight_t$. Respectively, the minimum world up direction value, $mineigheight_t$, i.e. how far voxel person is off the ground currently, is also recorded from the $\overrightarrow{eigheight}_{t,k}$ set.

The next feature is the similarity between voxel person's primary orientation, that is, the eigenvector with the largest corresponding eigenvalue, and the ground plane normal,

$$gpsim_t = \text{maximum}\left(\overrightarrow{eigenvec}_{t,1} \cdot \langle 0,0,1\rangle^T, \overrightarrow{eigenvec}_{t,4} \cdot \langle 0,0,1\rangle^T\right).$$

The $gpsim_t$ value helps in determining if the individual is upright, a value near 1, or if he or she is on the ground, a value near 0.

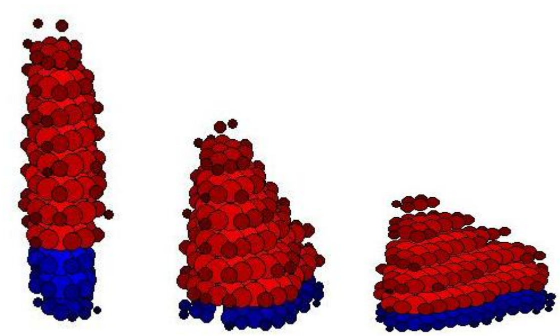
The above features are global. A final local feature is the centroid of the assumed leg set. This feature is helpful for inferring activities such as walking and standing. The reason for using the voxel leg set, instead of just V_t', is that upper body motion should not be considered when inferring walking or standing. Assumptions are made when calculating this feature. One must know when to extract it. The procedure for knowing when to extract and use this feature is discussed in subsequent chapters when state and activity inference is introduced. For example, if the human is upright (state) then it is safe to assume that the following feature is accurate and can be used to infer walking or standing (activities). When the human is upright, the set of voxels assumed to be legs is

$$Legs_t = \{\vec{v}'_{t,j} \in V_t' \,|v'_{t,j,k} < \left(\frac{1}{3}\right) maxeigheight_t\},$$

where $v'_{t,j,k}$ is the k "height" (world up direction) component of $\vec{v}'_{t,j}$. While this is not guaranteed to be the entire set of voxels belonging to the legs, it is a fair approximation for the goal of tracking the movement of the human through an environment. The centroid of the leg voxel set is

$$\vec{c}_t^{Legs} = \left(\frac{1}{|Legs_t|}\right) \sum_{\vec{v}'_{t,j} \in Legs_t}^{|Legs_t|} \vec{v}'_{t,j}.$$

In later chapters, temporal analysis of the motion of $\vec{c}_t^{legs}$ is introduced for recognizing activity. Figure 4.19 shows the leg set for a low resolution voxel person (5x5x5 inches).



(a) (b) (c)

Fig. 4.19. Examples illustrating the approximation of leg voxel sets during different activities. Assumed leg voxels are blue and a resolution of 5x5x5 inches is used. In (a), the human is standing. In (b), the humans legs and hands just touched the ground during a fall. In (c), the human is on the ground after a fall. The accuracy of this local feature depends on a systems ability to correctly infer context, i.e. decide if the person is standing, kneeling over, fallen, etc.

4.6 Environment Partitioning and Object Interaction

The inference of many human actions require interactions with non-human objects, e.g. couches and chairs. Object recognition, such as via the scale invariant feature transform (SIFT) [109], is not performed in this dissertation. Instead, this work relies on the use of assumed static important objects and regions of interest manually identified by a user. However, there is nothing about the following that restricts it from being used directly in combination with an object recognition and tracking system. One such example is the stereo vision and genetic algorithm-based non-human object segmentation work of Anderson et al. [70].

An environment is first broken up, in the ground (x-y) plane, into a set of non-overlapping regions, P_k, $1 \leq k \leq K$. Example locations include the living room, kitchen, and other areas that provide a context for subsequent activity analysis. In addition, objects of interest, where the ith object is O_i, $1 \leq i \leq B$, are manually identified. Assumed static objects of interest include the bed, recliner, couch, table, etc. Objects and regions are specified as polygons. Figure 4.20 illustrates this process.

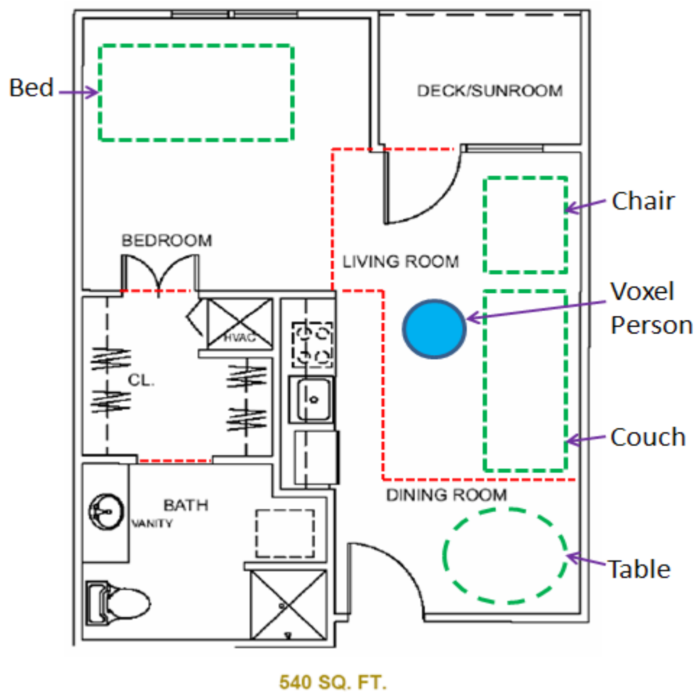


Fig. 4.20. Partitioning of an environment into regions, whose adjacent edges are shown in red, and objects, shown in green, for activity analysis monitoring.

The tracking of human-region interaction, such as the living room, kitchen, etc. is based on the projection of voxel person's centroid onto the ground plane. The region containing the projected centroid is identified. The tracking of human-object interaction, e.g. couch, chair, etc. is based on the amount of overlap between an object and the human on the ground (x-y) plane. This plane is considered to be the most discriminative for determining object overlap in the case of indoor monitoring. At each time step, the entire voxel person object is projected onto the ground plane and its overlap with each individual object, specified as polygons, is calculated. The amount of voxel person-object overlap, for object O_i, is

$$OVERLAP(V'_t, O_i) = \left(\frac{1}{P}\right)\sum_{j=1}^{P} CONTAINED(\vec{v}'_{t,j}, O_i),$$

where CONTAINED is a binary check for containment of voxel $\vec{v}'_{t,j}$ in polygon describing O_i. In both the object and region instances, checking for containment of a single point is a polygon can be achieved using a scan line technique for region filling in computer graphics [107]. A two dimensional scan line fill operation can be performed using an edge table and active edge list.

4.7 Summary

In summary, the back-projection of multiple image silhouettes collected at approximately the same moment in time from different cameras results in an object in voxel space. This object can be refined using knowledge of the scene and viewing conditions. The combination of voxel person and the visible shell gives rise to the blanketed set. The blanketed set helps specifically with the minimization of non-visible back-projection error. An algorithm is also introduced for the monitoring of an objects position in a low resolution voxel reconstruction quality space. This procedure informs a system about the potential construction quality of an object at a given location. The decision to suppress subsequent activity analysis for objects in regions of the environment that do not give rise to adequate construction is left to the particular system. On a final note, features for tracking human activity are outlined.

Human State and Activity Recognition

5.1 Introduction

Two approaches to modeling and inferring human activity from voxel person are presented. The first procedure, which is unique to this dissertation, is a hierarchy of fuzzy inference systems with linguistic summarization. The second approach, which is in line with the current state of art, is based on HMMs. The second approach is included for the purpose of comparative analysis.

5.2 Hierarchy of Fuzzy Inference and Linguistic Summarization

In the novel approach, linguistic variables are used to describe features. An example is the height of voxel person that might have a value of *tall*. Human state is inferred using only features from a single moment in time. State generally describes a human's pose. However, pose is sometimes combined with information about region or object interaction. An example is voxel person is sitting on the couch. Linguistic variables are also used to describe state. Thus, at each moment (frame), the human belongs to every state under consideration to a degree. While states are observed at each moment, activity occurs over a time interval. An l-th level activity, $1 \leq l \leq L$, is a concept inferred using features extracted from voxel person over a time interval along with information extracted from activity concepts inferred at lower levels of activity, i.e. $\{1, \ldots, l-1\}$. For notational simplicity, the first level of activity, $l = 1$, is assumed to be state. The level in the activity hierarchy expresses the activities complexity and number of dependencies required to infer such a concept. Currently, $l = 1$ states include: *upright*, *in between*, *on the ground*, *lying on the couch*, *on the chair*; $l = 2$ activities include: *standing*, *walking*, *relaxing on the* *chair, lounging on the couch*, *fall*; and an example $l = 3$ activity is *walking with an abnormal gait*. The proposed human activity analysis framework is illustrated in figure 5.1.

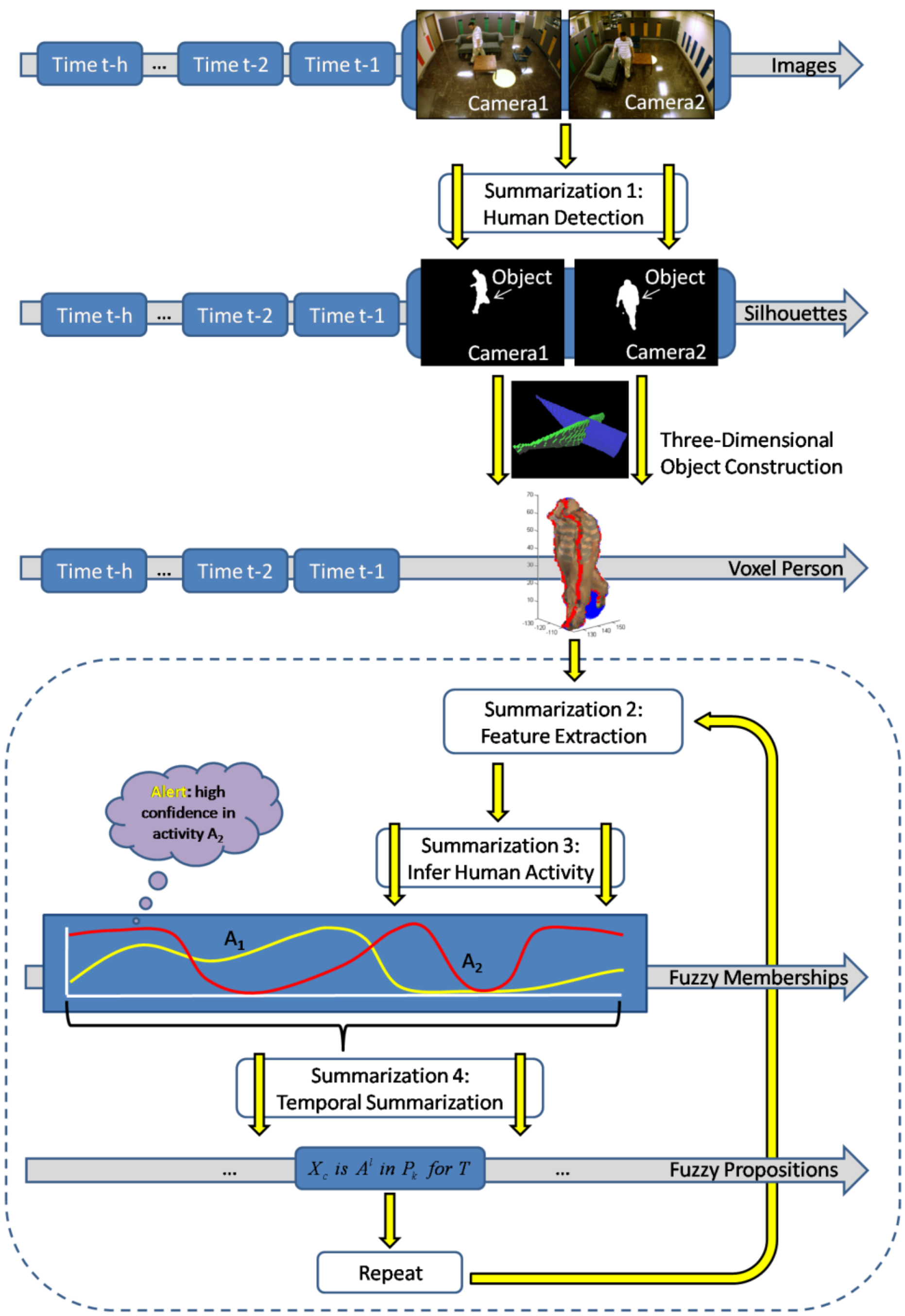


Fig. 5.1. Linguistic summarization framework for human activity analysis. Video is collected from multiple cameras, silhouettes are extracted, and voxel persons are built. The next three steps are repeated for the number of activity levels tracked. For each activity level, features are extracted from voxel person and lower level activity summarizations, activity is inferred, and then it is linguistically summarized.

The first significant departure from related work is the acceptance of linguistic variables for modeling features and activity instead of random variables. Under the frequentist interpretation of probability [75], probabilities represent the chance that a particular event is observed. In comparison, fuzzy memberships express ***compatibility*** with an event [120][121]. For example, a fuzzy membership of 0.8 indicates a *strong* compatibility between a human and an activity, while a probability of 0.8 indicates that 20% of the time it is believed that the human is performing a different activity. Thus, when error, i.e. occlusion, silhouette extraction, voxel object construction, etc, is present, which it always is, probability theory offers one way to account for our uncertainty based on how often a person was performing that activity given a set of features and a large number of previous observations. In contrast, fuzzy membership values express the degree to which a person currently matches an activity definition. This is most clear in the case of near perfect information. When there is little error, the interpretation of what a random variable represents with respect to an activity is unclear. There is no single point in which a human stops belonging to one activity and transitions into another. For example, a person can be somewhat on the ground or slightly upright. Fuzzy set theory gives us a way to address this problem. Hence, fuzzy set theory expresses the degree to which a person is *compatible* with an activity definition.

The major deficiency of using probability theory for human activity analysis resides in the tasks of inference and recognition. In this work, inference is fuzzy instead of probabilistic, rules are used instead of finite state automaton for representing knowledge, and inference is performed using fewer, richer linguistic summarizations instead of the original fully sampled time series. Specifically, fuzzy inference is used to acquire $l = 1$ activity (state) membership degrees. In a graphical model, specifically a single layer HMM, states are modeled as mixtures of probability density functions (pdfs), typically Gaussian or Poisson. Even if a hierarchical HMM (HHMM) is used, the bottom layer models state and all other layers are dedicated to non-state activity. After determining state, in either approach, there is still too much data, redundant and uninformative, contained in the state time series. In our approach, the time series is reduced to fewer numbers of linguistic summarizations. In a graphical model, state sequences are not summarized. This can lead to problems in inferring activity for moderate to long time sequences, even if normalization is used such as that classically proposed by Rabiner [78] for HMMs. HMMs use a finite state automation representation; a small finite set of states, pdfs associated with each state, typically a mixture model, and state transition probabilities, typically of time length one, i.e. $P(S_t = i|S_{t-1} = j)$. There is both the problem of forgetfulness in this approach, mostly related to the Markov and independence assumptions, and there are also numerical issues, because of the number of elements in the sequence and probabilistic inference is the product of many values in $[0,1]$. Probabilistic inference results in a likelihood value, which is useful for determining the most likely activity model, but no interpretable confidence value is generated. In contrast, the linguistic summarization method proposed in this dissertation utilizes explicit summarization, rule-based knowledge, and fuzzy inference is the mechanism used to infer activity. Fuzzy inference yields values in a meaningful interpretable confidence domain.

In addition, the proposed flexible rule-based framework allows for rules to be added, deleted, or modified to fit different types of residents based on knowledge about possible daily activity, physical status, cognitive status, and age. Rules can be modified to deal with elders that should never be on the ground unless they have fallen, versus active younger elders that might be on the ground stretching, exercising, etc. This framework makes it possible to model special cases where there is little training data, which is not the case with HMMs.

5.3 State Inference

After voxel person is constructed and features are extracted, the next task is the inference of the state of the human at each frame. The ith state is S_i and K states being tracked, $S = \{S_1, \dots, S_K\}$. The cth video sequence is $X_c = \{\vec{f}_{c,1}, \dots, \vec{f}_{c,T}\}$, where $\vec{f}_{c,t}$ is a feature vector extracted from voxel person at time t. For each feature, such as voxel person's centroid, a linguistic variable, $LV_i^{feature}$, and associated term set, $\{LT_{i,1}^{feature}, \dots, LT_{i,M_i}^{feature}\}$, is defined, where M_i is the number of terms in the ith linguistic variable. An example $LV_i^{feature}$ is voxel person's *height*, which has the term set $LT_i^{feature} = \{low, medium, high\}$. In addition, each state has a linguistic variable, LV_i^{state}, and term set, $\{LT_{i,1}^{state}, \dots, LT_{i,B_i}^{state}\}$, where B_i is the number of terms in the ith state linguistic variable. Each $LT_{i,j}^{feature}$ and $LT_{i,j}^{state}$ is a fuzzy set, specified by a set of parameters for a given membership function (Gaussian, trapezoidal, z-shaped, s-shaped, etc). Linguistic variables are used by rules to infer state. In this dissertation, domain experts defined the rule set and the set of linguistic variables and terms for features.

The algorithm to infer human state at time t, given $\vec{f}_{c,t} = (f_{c,t,1}, \dots, f_{c,t,D})$, where D is the number of features, is simple. State inference is the firing of the FIS in response to $\vec{f}_{c,t}$, which produces the tuple $\vec{\mu}_t = \langle \mu_{t,1}, \dots, \mu_{t,K} \rangle$, where $\mu_{t,i}$ is the $[0,1]$ membership value for S_i. If no rule fires in support of S_i, then $\mu_{t,i} = 0.5$ (total ambiguity) is assigned.

5.4 Linguistic Summarization of State

The result of reasoning about voxel person's state at time t is $\vec{\mu}_t = \langle \mu_{t,1}, \dots, \mu_{t,K} \rangle$. The objective is to take seconds, minutes, hours, and even days of resident activity and produce succinct linguistic summarizations, such as "the resident was preparing lunch in the kitchen for a moderate amount of time" or "the resident has fallen in the living room". This is a situation in which less information is more useful.

Reporting activity for every frame results in information overload. Linguistic summarization is designed to increase the understanding of the system output and produce a reduced set of salient descriptions that characterize a time interval. The linguistic summarizations help in informing nurses, residents, residents' families, and other approved individuals about the general welfare of the resident, and they are the input for the automatic detection of cognitive or functional decline or abnormal event detection.

State summarizations are built by temporally processing the fuzzy inference results about voxel person's state. The sequence $D = \{\vec{\mu}_1, \dots, \vec{\mu}_N\}$ has N elements, i.e. state decisions for N time steps. The first step in summarization is a median filtering of each individual state sequence from D,

$$\mu_{n,k} = median(\{\mu_{n-\mho,k}, \dots, \mu_{n,k}, \dots, \mu_{n+\mho,k}\}),$$

where $\mho$ is a time window parameter. The time series is median filtered for two reasons. First, due to factors such as silhouette segmentation, object construction, feature extraction, rule specification, term set specification, etc., it is possible to encounter erroneous or wrongly inferred moments. Median filtering helps remove outliers. Additionally, the frame rate of a system is generally a higher sampling rate than the speed at which most human activity occurs, especially for elders. This operator combines recent inference results in order to produce smoother decisions that model the reasonable rate at which change in activity might be expected. An example of the median filter is shown in figure 5.2.

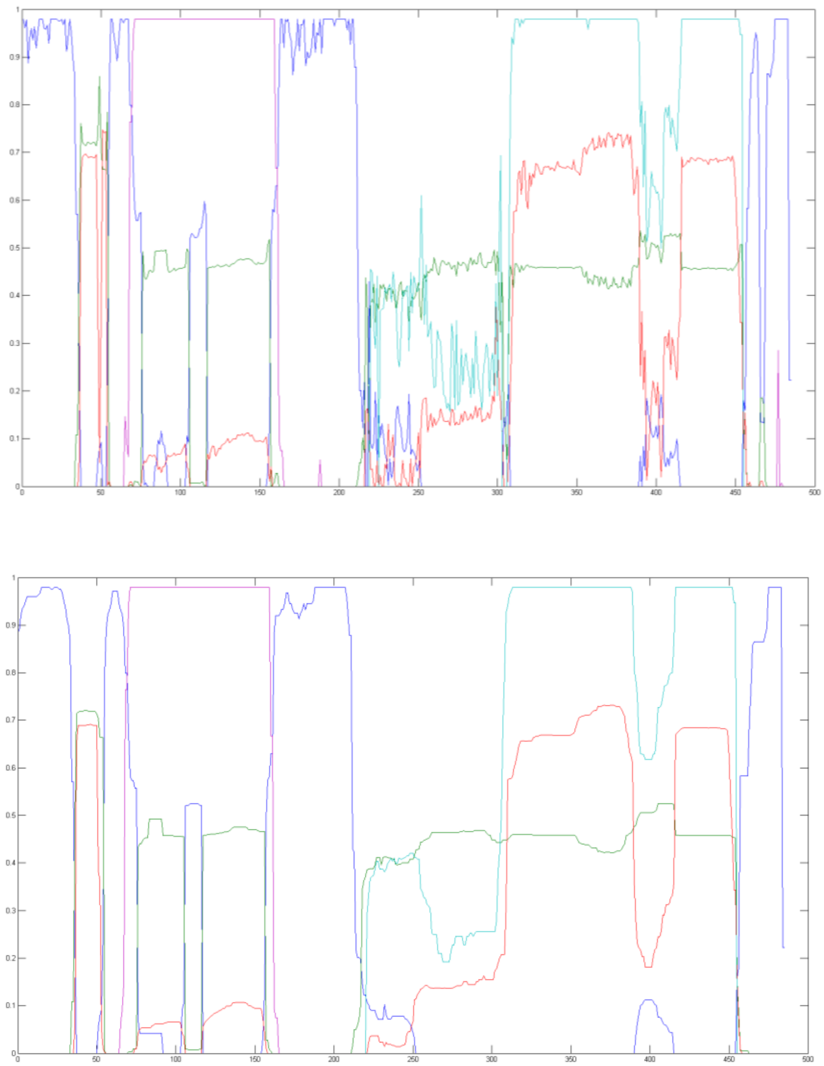


Fig. 5.2. Example state time series before (top) and after (bottom) the median filter. Here, Ʊ was chosen to be 3.

Next, two additional operations are performed on D. First, elements from D for which the maximum membership, $\mu_{t,max} = maximum_k(\mu_{t,k})$, is not clearly distinguishable from the other memberships,

$$\left(\mu_{t,max} - maximum_{j \neq argmax_k(\mu_{t,k})}(\mu_{t,j})\right) < \tau_1,$$

are removed. A parameter, $\tau_1 \in [0,1]$, is used for indeterminate maximum state identification. Specific values used in this work, as well as subsequent parameters introduced below in this section, are discussed in the experiments chapter. Also, samples that have too low of a maximum confidence value, $\mu_{t,max} < \tau_2$, are removed. The new sequence, D', has cardinality $N' = |D'|$, where $N' \leq N$. Each $\vec{\mu}'_t \in D'$ has its original position in D recorded, $I = \{i_1, \dots, i_{N'}\}$. A maximum state index sequence, $\omega = \{s_1, \dots, s_{N'}\}$, is also constructed, where $s_t = argmax_k(\mu'_{t,k})$.

Linguistic summarization of the state membership values is the generation of meaningful human understandable information of the form

X_c is S in P_k for T.

The object of interest, voxel person, is X_c $(1 \leq c \leq C)$. In this work, only a single resident is tracked, hence $C = 1$. The element P_k $(1 \leq k \leq K)$ is the current environment location. For example, the kitchen, living room, etc. Elements X_c and P_k are crisp, while T and S are fuzzy. The time duration of a summarization is the set $T = \{T_1, \dots, T_J\}$, where J is the number of terms defined over the time domain. The system does not discard the exact number of frames, it is just not reported as part of the summary. Even though T and S are recorded, each summary is generally reported to a human as "X_c is S_{max} in P_k for T_{max}", i.e. just the terms with maximum membership value. An example summarization of this form is "voxel person is (on the ground,0.95) in the living room for a (moderate amount of time,0.72)".

The sequence D' initially contains G summarizations. Summarizations are extracted by searching for indices where $s_j \neq s_{j+1}$ or $(i_j + 1) \neq i_{j+1}$, for $1 \leq j \leq (N' - 1)$. Thus, moments where the maximum state index changes or a gap is present are identified. This results in $U = \{u_1, \dots, u_{G-1}\}$, where u_i is an index from D' and $U = \emptyset$ when $G = 1$. Indices 1 and N' are appended to U, resulting in $U' = \{1, U, N'\}$. The gth summary, $1 \leq g \leq G$, Sum_g, is the sequence (interval) from u'_g to u'_{g+1}, where $u'_g \in U'$.

This initial segmentation can be improved based on three additional operations. The first step is to merge consecutive summarizations, Sum_g and Sum_{g+1}, if they are the same state and if there are only a few, τ_4, frames separating them. The reason for performing this operation is that error, e.g. in silhouette extraction, feature extraction, or inference, occurring for only a few frames can result in the separation of a single summary into two or more summaries.

In addition, since the goal is recognizing elderly activity, specifically falls, only summaries representing sufficient time duration are desired. The goal is not the recognition of high frequency activity occurring for a fraction of a second. Such periods are believed to be due to incorrect silhouette segmentation, inaccuracies in fuzzy inference, or high frequency activity not related to fall detection. Elders do not generally perform extremely quick activities, such as being on the ground for only one second. Therefore, any Sum_g, where $\left|Sum_g\right| < \tau_3$ is removed.

Lastly, the merging algorithm is performed one more time. The result of these operations is a new sequence of G' summarizations, $\left\{Sum'_1, \ldots, Sum'_{G'}\right\}$. Algorithm 5.1 is a summary of the above steps.

ALGORITHM 5.1. Temporal Summarization of Human State

---

1. Remove indeterminate and "too low" of confidence decisions

a. Remove elements where $\left(\mu_{t,max} - maximum_{j \neq argmax_k(\mu_{t,k})}(\mu_{t,j})\right) < \tau_1$ *or* $\mu_{t,max} < \tau_2$, *which results in* D' *and the associated sets* $\omega = \{s_1, \ldots, s_{N'}\}$ *and* $I = \{i_1, \ldots, i_{N'}\}$

2. Build the initial observed set of summaries, $\{Sum_1, \ldots, Sum_G\}$

a. Partition the sequence based on changes in values in ω and I, i.e., $s_j \neq s_{j+1}$ or $(i_j + 1) \neq i_{j+1}$

b. Merging step

i. If Sum_i *and* Sum_{i+1} *are the same state and if the number of frames separating them is less than* τ_4, *then merge the summaries*

4. Build the final set of summaries, $\{Sum'_1, \ldots, Sum'_{G'}\}$

a. Remove too brief of time duration summaries, i.e. $|Sum_g| < \tau_3$

b. Merging step

i. If Sum'_i *and* Sum'_{i+1} *are the same state and if the number of frames separating them is less than* τ_4*, then merge the summaries*

5.5 Activity Inference and Summarization

As alluded to above, human activity is assumed here to be hierarchical (illustrated in figure 5.3). If a person has fallen (activity) then they are on the ground (state). Additionally, if someone is walking with a limp (activity) then they are walking (activity), which in return is based on being upright (state). The proposed soft computing system is designed to exploit this hierarchical nature of human activity.

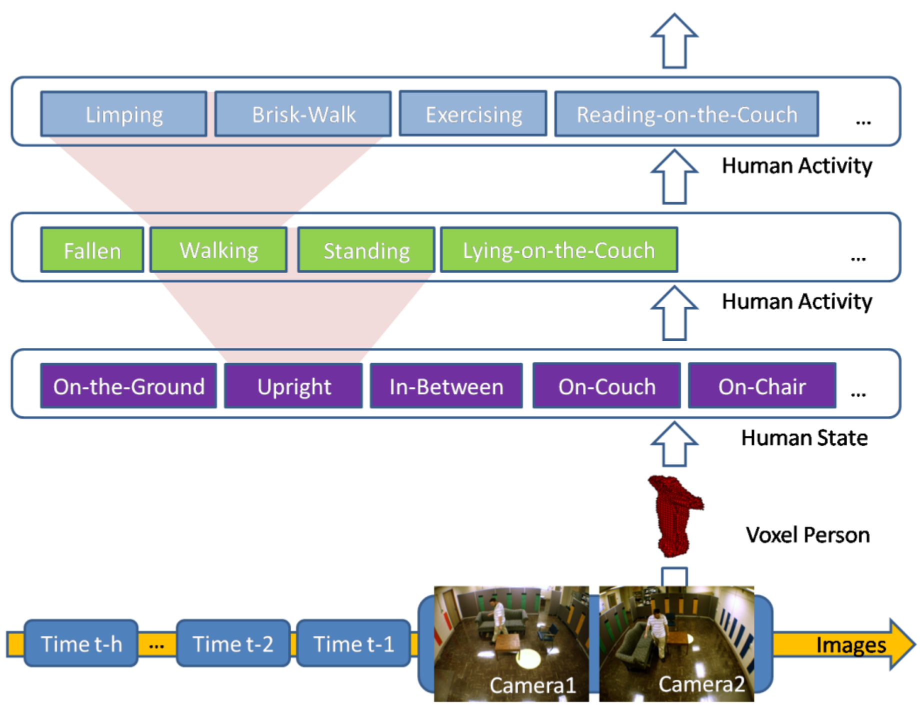


Fig. 5.3. Illustration of the assertion that human activity is naturally hierarchical. Shaded truncated pyramids show nested structure. For example, limping and brisk-walk are types of walking, walking and standing depend on the state upright, and upright depends on voxel person features.

Activity level l is $A^l = \{A_1^l, \ldots, A_{N_l}^l\}$, where A_i^l is the i$^{\text{th}}$ activity at level l. As already discussed, $A^1 = S = \{S_1, \ldots, S_K\}$. Linguistic variables are used to describe activities as well as features. Again, the linguistic variables and terms presented in this dissertation are defined by domain experts (nurses).

Activity inference is performed at each frame, or alternatively in response to particular triggers. For a system designed to just recognize falls, activity inference only needs to be performed in the case that an "on the ground" state summary is encountered. As with state, temporal segmentation algorithm 5.1 is used to acquire linguistic summarizations of level two and above activity, i.e. $Sum'_{l,g}$.

Multiple frames, i.e. voxel person over multiple time steps, are required to recognize activity at levels $\{2, \ldots, L\}$. The first feature is the confidence of summary A_i^l in $Sum'_{l,g}$,

$$\pi_{A_i^l, Sum'_{l,g}} = median\left(\left\{\mu_{t_{start},i}, \ldots, \mu_{t_{start}+\left|Sum'_{l,g}\right|-1,i}\right\}\right),$$

where t_{start} is the first index in $Sum'_{l,g}$. This feature is valuable for expressing how confident the system is that a particular activity was performed.

The next temporal feature is based on voxel person's motion vector between time t and $t+1$,

$$\vec{m}_{t \to t+1} = \vec{c}_{t+1} - \vec{c}_t.$$

The magnitude of the motion vector, $\|\vec{m}_{t \to t+1}\|$, represents the person's speed. The distance traveled per second by voxel person during $Sum'_{l,g}$ is

$$\varphi_{Sum'_{l,g}}^{\vec{c}_t} = \left(\frac{F}{|Sum'_{l,g}| - 1}\right)\left(\sum_{j=0}^{|Sum'_{l,g}|-2} \|\vec{m}_{t_{start}+j \to t_{start}+j+1}\|\right)\left(\frac{R}{M}\right),$$

where t_{start} is the index of the first motion vector magnitude in $Sum'_{l,g}$. Value M is a unit normalizing factor. If world units are specified in inches then $M = 12$ converts the final measurement into feet. Value R is the resolution of voxel space. An example is $R = 5$ for a 5x5x5 inch voxel resolution. The value F is the system frame rate. This value is used to convert the final measurement into seconds. Given the parameter set $\{F = 3, R = 5, M = 12\}$, $\varphi_{Sum'_{l,g}}^{\vec{c}_t}$ is the average distance moved in feet per second per summary. This feature is useful for situations such as analyzing how much motion is present while a person is on the ground during a fall. If someone was rendered unconscious, then no motion should be observed during the summary time period. Additionally, $\vec{m}_{t \to t+1}$ is useful for analyzing moments of quick change in the persons speed before a state, such as on the ground, which might indicate a fall.

The next measurement is based on the idea that a system might not need to know the distance traveled per second for an entire summary, but rather the distance traveled per second recently relative to the present state (or activity) summary. For example, it is helpful to know the distance traveled per second over the last few seconds relative to the human being upright. Any motion outside of this state might be of no concern to the particular decision at hand, such as inferring gait. The feature used in this work to infer standing and walking is

$$\Omega_{t-h,\ldots,t}^{\vec{c}_{t,(x,y)}^{Legs}} = \text{maximum}_{i,j \in \{t-h,\ldots,t\}}\left(d\left(\vec{c}_{i,(x,y)}^{Legs}, \vec{c}_{j,(x,y)}^{Legs}\right)\right),$$

$$g_k = \begin{cases} 1 & if \ \left(\Omega_{t-k,\ldots,t}^{\vec{c}_{t,(x,y)}^{Legs}}\right)\left(\frac{R}{M}\right) > \zeta \\ 0 & else \end{cases},$$

$$\vartheta_{\zeta,\varrho}^{\vec{c}_{t,(x,y)}^{Legs}} = \begin{cases} \left(\dfrac{\underset{\substack{\forall k \in \{1,\ldots,\varrho\} \\ s.t.\ argmax_j(\mu_{t-k,j})=argmax_j(\mu_{t,j})}}{\operatorname{argmin}} (g_k = 1)}{F} \right) & if\ \exists g_k = 1 \\ \left(\dfrac{\varrho}{F}\right) & else \end{cases}.$$

The first formula, $\Omega_{t-h,\ldots,t}^{\vec{c}_{t,(x,y)}^{Legs}}$, is a measure of the maximum distance traveled by the centroid of the leg set over a time window h. The window parameter h is varied in the search for $\vartheta_{\zeta,\varrho}^{\vec{c}_{t,(x,y)}^{Legs}}$. The value g_k, one for each different window length tested, i.e. as h is varied, is a piecewise defined binary function that tests if the human has moved a distance greater than some user specified amount, ζ, for example, has a human moved at least two feet. The value $\vartheta_{\zeta,\varrho}^{\vec{c}_{t,(x,y)}^{Legs}}$, the amount of time that it took voxel person to move a specific distance, is the piecewise defined test for the minimum amount of time one must look back in time in order for the leg set to have traveled a minimum distance ζ, bounded by a user specified ϱ number of time steps and the current summary.

The next measurement is the detection of a large recent change in voxel person's speed before the current $l-1$ level summary. This calculation helps with the detection of impact-based falls (activity). Inferring this human activity involves looking for when a human transitions too fast from an upright pose (state) to an on the ground pose (state). A window of size W of magnitudes of motion vectors is analyzed before the summary,

$$\{\|\vec{m}_{t-W\to t-W+1}\|, \ldots, \|\vec{m}_{t-1\to t}\|\},$$

where t is the start of the current summary. Elements in this window are first smoothed with a median filter of size Ψ, resulting in $\|\vec{m}'_{t\to t+1}\|$. The derivative is then calculated using forward finite difference,

$$\nabla\|\vec{m}'_{t\to t+1}\| = \|\vec{m}'_{t+1\to t+2}\| - \|\vec{m}'_{t\to t+1}\|.$$

The detection of a quick change involves identifying a large change in the sequence $\nabla\|\vec{m}'_{t\to t+1}\|$ in the second half of the window. The maximum $\nabla\|\vec{m}'_{t\to t+1}\|$ from the first half of the window,

$$sd_{half1} = maximum(\{maximum_i(\nabla\|\vec{m}'_{t-W+i\to t-W+i+1}\|), 0\}),$$

where $i = 0, \ldots, \lfloor W/2 \rfloor - 1$, and the maximum in the second half of the window,

$$sd_{half2} = maximum(\{maximum_i(\nabla\|\vec{m}'_{t-W+i\to t-W+i+1}\|), 0\}),$$

where $i = \lfloor W/2 \rfloor, \ldots, W - 1$, are calculated. The relative ratio feature used is sd_{half2}/sd_{half1}.

The last feature, OSC_{h,A_i^l,A_j^l}, is the (discrete) number of times that the system changed back and forth between activities A_i^l and A_j^l over the last h time steps. Counting starts at the current time step t and if any activity outside of A_i^l or A_j^l is encountered during the search backwards in time then counting is terminated. This feature has helped in the detection of falls in which the person is on the ground and keeps unsuccessfully attempting to get back to an upright stance.

Rule specification and the algorithm for performing activity inference are the same as discussed in section 5.3. The result of activity inference at time step t is the membership tuple $\vec{\mu}_t^l = \langle \mu_{t,1}^l, \ldots, \mu_{t,N_l}^l \rangle$, where $\mu_{t,i}^l$ is the $[0,1]$ membership of A_i^l.

5.6 Alert Generation

In the context of surveillance, an important goal is the recognition and subsequent issuing of an alert if a particular summary is ever observed with enough confidence. Here, the primary example is an alert for the detection of a fall. The goal is to make a crisp decision from the fuzzy summary information. The simplest method for making a crisp decision is the identification of a confidence threshold, $\tau_5 \in [0,1]$. An advantage of fuzzy logic for the task of inference is that the value τ_5 can now be interpreted. This approach is more reliable than attempting to pick a threshold for the likelihood of a model occurring in an HMM, which is generally the product of a large number of $[0,1]$ values, or the ad hoc selection of a ratio of the top two most likely models, which does not necessarily tell one if the activity was even performed.

The procedure used to select τ_5 is as follows. First, a user selects an activity consequent domain term and a minimum acceptable membership degree for that term. The minimum of the user specified membership threshold and the term is computed. The value τ_5 is the activity consequent domain location of the centroid for the resultant set. The rationale for this procedure is as follows. First, it is natural for a user to linguistically indicate a minimal acceptable confidence, i.e. the system must generate a confidence of *high* or greater. However, this fuzzy set needs to be converted into a numeric value for crisp alert detection. This is the reason why a user is required to specify a minimal acceptable degree for which the term must be fired. The value used in this work is 0.5 (total ambiguity). This means that at frame t the FIS must yield a decision whose activity consequent domain centroid location is greater than τ_5. The specific set of activity consequent domain terms, their associated membership function types and parameters, and the resulting τ_5 value based on this procedure is provided in the experiments section.

5.7.1 Hidden Markov Models – Most Likely Model

The second approach to activity analysis introduced here for comparative analysis is also based on voxel person. In the literature, this type of approach generally uses image silhouettes [19][20]. However, to draw a fair comparison between HMMs and the proposed hierarchy of fuzzy logic and linguistic summarization, voxel person is used.

The c$^{\text{th}}$ video sequence is $X_c = \{\vec{f}_{c,1}, \ldots, \vec{f}_{c,T}\}$, where $\vec{f}_{c,t}$ is the feature vector extracted from voxel person at time t. A collection of observation sequences, $X = \{X_1, \ldots, X_C\}$, where the length of sequences do not have to be the same, is used to train a set of HMMs. Each activity, such as falling, has a separate

HMM, possibly multiple models, that are trained via the Baum-Welch procedure [78]. Given a new sequence, the most likely model is calculated according to the forward-backward procedure, with appropriate scaling to avoid problems with numerical precision [78]. As already introduced in section 2.5, HMM_d is specified according to (A, B, π). The notation $(q_1, \dots, q_T)$, a specific sequence of states through HMM_d based on the observation sequence X_c, is introduced to simplify and compact the following discussion. Additionally, this is the typical notation introduced by Rabiner [78]. In the case of a flat stationary classical HMM represented as a graphical model, the parameters (A, B, π) are the same, outside of the specific method used to estimate them. The initial state distribution is given by

$$\pi(i) = P(q_1 = i),$$

the finite state transition probabilities are

$$A(i, j) = P(q_t = j | q_{t-1} = i),$$

and B, the observation symbol probabilities, which are either discrete or continuous, is defined differently depending on if a mixture model approach is taken. In this work, a continuous Gaussian mixture model is used. Each mixture, L per state, is specified by a mean and full covariance matrix. The l^{th} mixture in the i^{th} state has a weight of $w_{i,l}$, where

$$\sum_{l=1}^{L} w_{i,l} = 1,$$

and a probability density function of

$$\mathrm{N}(\vec{f}_{c,t}; \vec{\mu}_{i,l}, \Sigma_{i,l}).$$

Therefore, the probability of state i given the observation $\vec{f}_{c,t}$, $B\left(i, \vec{f}_{c,t}\right)$, is

$$\sum_{l=1}^{L} w_{i,l}\, \mathrm{N}\left(\vec{f}_{c,t}; \vec{\mu}_{i,l}, \Sigma_{i,l}\right).$$

The probability of sequence X_c, according to HMM_d is $P(X_c|HMM_d)$. The goal of the forward-backward procedure is an efficient calculation of $P(X_c|HMM_d)$,

$$P(X_c|HMM_d) = \sum_{all\ Q} P(X_c|Q_k, HMM_d)\, P(Q_k|HMM_d),$$

where Q_k is the kth sequence of states, $Q_k = (q_1, \dots, q_T)$, and "$all\ Q$" denotes the iteration over all possible state sequences. The value $P(X_c|Q_k, HMM_d)$ is specified according to

$$\prod_{t=1}^{T} P\left(\vec{f}_{c,t}\middle|q_t, HMM_d\right),$$

and $P(Q_k|HMM_d)$ is

$$\pi(q_1)A(q_1, q_2) \dots A(q_{T-1}, q_T).$$

This brute force method has complexity $2TN^T$, where N is the number of states and T is the observation sequence length. The unrolling of this process into T time steps reveals an underlying trellis structure, in which dynamic programming, specifically the forward-backward procedure, assists in a more efficient calculation, with a manageable complexity order of NT^2. The forward variable is

$$\alpha_t(i) = P\left(\vec{f}_{c,1}, \dots, \vec{f}_{c,t}, q_t = i\middle|HMM_d\right),$$

which is the probability of the partial observation sequence, $(\vec{f}_{c,1}, \dots, \vec{f}_{c,t})$, and state i at time t, given HMM_d. The value of $\alpha_t(i)$ and $P(Q_k|HMM_d)$, where HMM_d has N states, are easily found via

1. $\alpha_1(i) = \pi(i)B\left(i, \vec{f}_{c,1}\right)$
2. $\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)A(i,j)\right]B\left(j, \vec{f}_{c,t+1}\right)$
3. $P(X_c|HMM_d) = \sum_{j=t}^{N} \alpha_T(j)$.

The backward component of the forward-backward algorithm is

1. $\beta_T(i) = 1$

2. $\beta_t(i) = \sum_{j=1}^{N} A(i,j)\, B\big(j, \vec{f}_{c,t+1}\big)\beta_{t+1}(j)$

3. $P(X_c|HMM_d) = \sum_{j=t}^{N} \beta_1(j)\pi(j)\, B\big(j, \vec{f}_{c,1}\big)$.

However, the $\beta_t(i)$ values are not necessary for the calculation of $P(X_c|HMM_d)$.

Algorithm 5.3 is the proposed HMM-based human activity recognition algorithm. It uses a sliding window of size Γ to determine the most likely activity label at time t.

ALGORITHM 5.3. HMM-Based Activity Recognition from Voxel Person

----

1. Set X, the observation sequence, to empty

2. For t=1 to T (each iteration is a frame)

a. For each camera ($1 \leq c \leq C$)

i. Generate the silhouette for camera c, $S_{t,c}^{F}$

ii. Generate camera c's account of voxel person, $V_{t,c}$

b. Intersect all camera voxel persons to obtain intersection-based voxel person, $V_t' = \bigwedge_{c=1}^{C} V_{t,c}$

c. Produce the visible shell using algorithm 4.1, $S_t' = \bigvee_{c=1}^{C} S_t^c$

d. Produce the blanketed set using algorithm 4.2, $U_t = \left\{\vec{v}_j' \in V_t' \,|\, \vec{v}_j' \in \Omega(S_t')\right\}$

e. Extract features from U_t and append to the sequence X

3. Optional step, median filter the individual features in X

4. For t=1 to (T- Γ)

a. For d=1 to D, i.e. for each model

i. Use the forward-backward algorithm and compute $P(X_{t:t+\Gamma}|HMM_d)$

b. Select the model with the largest $P(X_{t:t+\Gamma}|HMM_d)$ likelihood value

i. $HMM_{max} = argmax_{HMM_d} P(X_{t:t+\Gamma}|HMM_d)$

c. Optional step

i. Attempt to reject activity based on the ad hoc criteria of $P(X_{t:t+\Gamma}|HMM_{max})<\delta_1$ *or* $(P(X_{t:t+\Gamma}|HMM_{max})/P(X_{t:t+\Gamma}|HMM_{max2})) < \delta_2$*, where* δ_1 *and* δ_2 *are user defined thresholds, and* $HMM_{max2} = argmax_{HMM_d \neq HMM_{max}} P(X_{t:t+\Gamma}|HMM_d)$

5.7.2 Hidden Markov Models – Model Parameter Estimation

No known analytical solution for solving the HMM parameters exists. The most popular iterative approach for estimating its parameters is the Expectation Maximization (EM) algorithm [75][78][122]. The EM algorithm is a method for finding the maximum-likelihood (ML) estimate of the parameters of an underlying distribution from a given data set when the data is incomplete or has missing or hidden values. Each iteration is guaranteed to increase the log likelihood and the algorithm is guaranteed to converge to a local maximum of the likelihood function. In the standard ML case, the likelihood function is

$$L(\theta|X) = p(X|\theta) = \prod_{i=1}^{N} p(\vec{x}_i|\theta),$$

and the task is one of discovering θ^*,

$$\theta^* = argmax_\theta L(\theta|X),$$

where $log\big(L(\theta|X)\big)$ is generally used because it is analytically easier to solve. As in the case of data that is incomplete, either because of missing or hidden values, a complete data set, $Z = (X, Y)$, is assumed to exist, which gives rise to the joint density function

$$p(\vec{z}|\theta) = p(\vec{x}, \vec{y}|\theta) = p(\vec{y}|\vec{x}, \theta)p(\vec{x}|\theta).$$

For instances such as an HMM with a mixture of densities for each state, $p(\vec{x}|\theta)$ is

$$p(\vec{x}|\theta) = \sum_{i=1}^{M} w_i p_i(\vec{x}|\theta_i),$$

where each p_i is a density, such a Gaussian, characterized by θ_i, such as a mean and covariance matrix, $\theta_i = (\vec{\mu}_i, \Sigma_i)$, and w_i is the mixing coefficient. The log likelihood function, $log\big(L(\theta|X)\big)$, is

$$log\big(L(\theta|X)\big) = log \prod_{i=1}^{N} p(\vec{x}_i|\,\theta) = \sum_{i=1}^{N} log\left(\sum_{j=1}^{M} w_j p_j(\vec{x}_i|\theta_j)\right),$$

which is difficult to optimize because it contains the log of a sum. However, if a new likelihood function, $L(\theta|Z) = L(\theta|X,Y) = p(X,Y|\theta)$, called the complete data-likelihood function, is formulated,

$$log\big(L(\theta|X,Y)\big) = log(P(X,Y|\theta)) = \sum_{i=1}^{N} log(P(\vec{x}_i|\vec{y}_i)P(y)) = \sum_{i=1}^{N} log\big(w_{y_i} p_{y_i}(\vec{x}_i|\theta_{y_i})\big),$$

optimization is simplified. However, the problem switches to the fact that the values of Y are not known. The trick is in assuming that Y is a random vector, and the EM algorithm is used. The EM algorithm consists of two alternating steps, the E-step (expectation) and M-step (maximization). In the E-step, an expectation of the likelihood, using a current estimate of the hidden data, is found, and in the M-step maximum likelihood estimates of the parameters are found through maximization of the expected likelihood found in the E step.

The values $\alpha_t(i)$ and $\beta_t(i)$ were discussed when introducing the forward-backward procedure. These quantities are used to form the variable $\gamma_t(i)$,

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(X_c|HMM_d)},$$

which is the probability of being in state i at time t, given the observation sequence $\{\vec{f}_{c,1}, \ldots, \vec{f}_{c,T}\}$ and the model HMM_d. In addition, the variable $\xi_t(i,j)$,

$$\xi_t(i,j) = \frac{\alpha_t(i)A(i,j)B\left(j,\vec{f}_{c,t+1}\right)\beta_{t+1}(j)}{P(X_c|HMM_d)},$$

is the probability of being in state i at time t, and state j at time t + 1, given the model and the observation sequence. The following estimation formula for the HMM parameters can be derived directly by maximizing Baum's auxiliary function [122][78]. When C observation sequences are present, the expected number of times in state i at time t = 1 is

$$\pi(i) = \frac{\sum_{j=1}^{C} \gamma_t^j(i)}{C},$$

where $\gamma_t^j(i)$ is the $\gamma_t(i)$ value for the jth observation sequence. The mixture coefficient update is

$$w_{i,k} = \frac{\sum_{j=1}^{C} \sum_{t=1}^{T_j} \gamma_t^j(i,k)}{\sum_{j=1}^{C} \sum_{t=1}^{T_j} \gamma_t^j(i)},$$

where T_j is the length of the jth observation sequence, and $\gamma_t^j(i,k)$, the probability that the kth component of the ith mixture generated observation $\vec{f}_{j,t}$ is

$$\gamma_t^j(i,k) = \gamma_t^j(i) \frac{w_{i,k} B\left(i,k,\vec{f}_{j,t}\right)}{B\left(i,\vec{f}_{j,t}\right)},$$

where

$$B\left(i,k,\vec{f}_{j,t}\right) = w_{i,k} \mathrm{N}\left(\vec{f}_{j,t}; \vec{\mu}_{i,k}, \Sigma_{i,k}\right)$$

and

$$B\left(i,\vec{f}_{j,t}\right) = \sum_{k=1}^{K} w_{i,k}\, \mathrm{N}\left(\vec{f}_{j,t}; \vec{\mu}_{i,k}, \Sigma_{i,k}\right).$$

The transition probabilities updates are

$$A(i,j) = \frac{\sum_{j=1}^{C} \sum_{t=1}^{T_j} \xi_t^j(i,j)}{\sum_{j=1}^{C} \sum_{t=1}^{T_j} \gamma_t^j(i)},$$

and the mean and covariance matrix updates are

$$\mu_{i,k} = \frac{\sum_{j=1}^{C} \sum_{t=1}^{T_j} \gamma_t^j(i,k) \vec{f}_{j,t}}{\sum_{j=1}^{C} \sum_{t=1}^{T_j} \gamma_t^j(i,k)},$$

$$\Sigma_{i,k} = \frac{\sum_{j=1}^{C} \sum_{t=1}^{T_j} \gamma_t^j(i,k) \left(\vec{f}_{j,t} - \vec{\mu}_{i,k}\right)\left(\vec{f}_{j,t} - \vec{\mu}_{i,k}\right)^T}{\sum_{j=1}^{C} \sum_{t=1}^{T_j} \gamma_t^j(i,k)}.$$

These update equations are iteratively applied until convergence, typically given by

$$\left| log\left(L(\theta_{t+1}|X,Y)\right) - log\left(L(\theta_t|X,Y)\right) \right| \leq \tau,$$

or a maximum number of iterations.

5.8 Summary

In summary, two different approaches to modeling and inferring human activity from voxel person are outlined. The first novel procedure, unique to this dissertation, is a hierarchy of fuzzy inference and linguistic summarization. The second approach is based on HMMs and is included for the purpose of comparative analysis. Theoretical arguments are made in this chapter for the use of linguistic variables instead of mixture of probability distribution functions and fuzzy inference in place of probabilistic inference for the task of activity analysis. In addition, temporal summarization of a time series is explicitly addressed in the first approach in order to yield succinct human understandable reports as well as reduce the computational complexity for subsequent recognition of activity at various levels of complexity. Lastly, a procedure is discussed for interpreting and selecting an activity confidence threshold to use for alert detection.

Experiments

6.1 Introduction

This chapter is a documentation of the specific states, activities, features, linguistic variables, terms, and other various parameters used in the proposed human activity analysis systems. These systems are designed primarily to recognize falls in an elderly population. Nurses assisted in the design and specification of system parameters. The system has been extended beyond fall recognition to include a few additional common non-fall activities to demonstrate the flexibility of the general framework. In addition, this chapter outlines the experiments performed, data sets analyzed, protocol for data collection, and metrics used to evaluate the success of the system.

6.2.1 Hierarchical Framework for Linguistic Summarization - State

The inference of human state at each image using features extracted from voxel person is detailed in this section. The following states are used in this work.

- **Upright** (S_1 or A_1^1): This state is generally characterized by voxel person having a *high* height, its centroid being at a *medium* height, and a *high* similarity of the ground plane normal with voxel person's primary orientation. Activities that involve this state are, for example, standing, walking, walking with a limp, and meal preparation.

- **On-the-ground** (S_2 or A_2^1): This state is generally characterized by voxel person having a *low* height, a *low* centroid, and a *low* similarity of the ground plane normal with voxel person's primary orientation. Example activities include a fall, stretching, and exercising.
- **In-between** (S_3 or A_3^1): This state is generally characterized by voxel person having a *medium* height, *medium* centroid, and an *indeterminate* (non-identifiable) primary orientation or *high* similarity of the primary orientation with the ground plane normal. With respect to many falls, this state represents the transition between being **upright** to being **on-the-ground**. Other example activities are hunching over, reaching over to pick up/interact with an object, and trying to get back up to a standing stance after a fall.
- **On-the-chair** (S_4 or A_4^1): This state is generally characterized by voxel person being *on* the chair. Examples include sitting on the chair reading or watching television.
- **Lying-on-the-couch** (S_5 or A_5^1): This state is generally characterized by voxel person being *on* the couch, having a *high* eigen-based minimum z value, a *low* similarity of the ground plane normal and voxel person's primary orientation. Example activities include sleeping on the couch and lying on the couch watching television.

It is important to note that being on the ground (state) does not imply a fall (activity). Additionally, none of the features above sufficiently identify voxel person's state all of the time. Instead, each feature is helpful for determining the degree to which voxel person is in a particular state. Explanations such as *large*, *medium*, or a *low* amount of each feature characterize the state descriptions. There is no crisp point where the features change between states.

The standard Mamdani type FIS is used. The membership function type for antecedent linguistic variable terms used to infer pose-based states, reported in table 6.1, is a trapezoid (μ_A^T). A trapezoid is defined with respect to the four ordered points (a, b, c, d). These values represent the trapezoid (a) left most point, (b) left central point, (c) right central point, and (d) right most point. A trapezoid is desirable because it is computationally simple, can be described compactly using two level cuts, and multiple shapes, both symmetric and non-symmetric, can be formed.

Table 6.1. Antecedent linguistic variables and terms used for the inference of pose-based states. Each variable has its input clamped to the specified domain.

	Terms		
Antecedent Linguistic Variables	*Low:* μ_A^T	*Medium:* μ_A^T	*High:* μ_A^T
Centroid - domain [0,10]	(0.0,0.0,1.0,1.2)	(0.5,1.0,2.0,2.5)	(1.8,2.0,10.0,10.0)
Eigen-based maximum height – domain [0 10]	(0.0,0.0,1.0,3.0)	(1.5,2.5,3.5,4.5)	(3.0,4.0,10.0,10.0)
Eigen-based minimum height – domain [0 10]	(0.0,0.0,0.1,0.15)	(0.1,0.15,0.25,0.3)	(0.25,0.3,10.0,10.0)
Max eigenvector and ground plane normal similarity - domain [0 1]	(0.0,0.0,0.4,0.45)	(0.35,0.4,0.6,0.65)	(0.55,0.6,1.0,1.0)

The fuzzy sets for consequent linguistic variable terms, provided in table 6.2, are a mixture of trapezoid (μ_A^T), z-shaped (μ_A^Z), and s-shaped (μ_A^S) membership functions. The μ_A^Z and μ_A^S membership functions, defined and explained in chapter 2, are specified according to two control points (a, b). The fuzzy sets in table 6.2 have membership functions whose parameters go outside the input range, specifically *very low* and *very high*. This is performed to control where the resulting centroid of each set is located. The rule set used to infer the pose-based states of voxel person are provided in Table 6.3.

Table 6.2. Terms for the consequent linguistic variables **upright**, **in-between**, **on-the-ground**, **on-the-chair**, and **lying-on-the-couch**.

Very Low: μ_A^Z	*Low:* μ_A^T	*Medium:* μ_A^T	*High:* μ_A^T	*Very High:* μ_A^S
(0.0,2.0)	(0.0,0.0,0.25,0.5)	(0.0,0.4,0.6,1.0)	(0.5,0.75,1.0,1.0)	(-1.0,1.0)

Table 6.3. Rules for recognizing the pose-based states **upright**, **in-between**, and **on-the-ground**.

Rule		Centroid	Normal Similarity	Eigen Based Height		**upright**	**in-between**	**on-the-ground**
1	If	*High*	*High*	*High*	Then	*Very High*	*Very Low*	*Very Low*
2		*Medium*	*High*	*High*		*High*	*Medium*	*Very Low*
3		*Low*	*High*	*High*		*Low*	*High*	*Low*
4		*High*	*High*	*Medium*		*Low*	*Medium*	*Low*
5		*Medium*	*High*	*Medium*		*Low*	*Very High*	*Low*
6		*Low*	*High*	*Medium*		*Low*	*Very High*	*High*
7		*Medium*	*High*	*Low*		*Very Low*	*Low*	*Medium*
8		*Low*	*High*	*Low*		*Very Low*	*Low*	*Very High*
9		*High*	*Medium*	*High*		*Very High*	*Low*	*Very Low*
10		*Medium*	*Medium*	*High*		*High*	*Medium*	*Very Low*
11		*Low*	*Medium*	*High*		*Low*	*Medium*	*Low*
12		*High*	*Medium*	*Medium*		*Low*	*High*	*Low*
13		*Medium*	*Medium*	*Medium*		*Low*	*Very High*	*Low*
14		*Low*	*Medium*	*Medium*		*Very Low*	*High*	*High*
15		*Medium*	*Medium*	*Low*		*Very Low*	*Medium*	*High*
16		*Low*	*Medium*	*Low*		*Very Low*	*Low*	*Very High*
17		*High*	*Low*	*High*		*High*	*Low*	*Very Low*
18		*Medium*	*Low*	*High*		*Medium*	*Low*	*Very Low*
19		*Low*	*Low*	*High*		*Low*	*Medium*	*Low*
20		*High*	*Low*	*Medium*		*Low*	*Medium*	*Low*
21		*Medium*	*Low*	*Medium*		*Low*	*High*	*Low*
22		*Low*	*Low*	*Medium*		*Very Low*	*Low*	*High*
23		*Medium*	*Low*	*Low*		*Very Low*	*Low*	*Medium*
24		*Low*	*Low*	*Low*		*Very Low*	*Very Low*	*Very High*

It should be noted that these rules make it possible to detect cases such as when voxel person is lying on the ground, not just lying down anywhere. If a person is lying on a couch or a bed, he or she should not have a low centroid and will not have a low height. The rule that would generally be dominant by voxel person lying on a bed or couch is rule 21. In this situation he or she would typically have a medium centroid and a medium height. In all of these mentioned situations, lying on the bed, lying on the ground, and lying on the couch, voxel person should generally have a low max eigenvector and ground plane normal similarity. This case-based approach to human activity analysis makes it relatively easy to add rules for recognizing additional states.

An example of the inferred state memberships over time is illustrated in Figure 6.1. In this situation, the camera's capture rate is 3 fps, the states are {**on-the-ground**, **in-between**, and **upright**}, and the scenario represents a person falling and not making it back up to an upright pose.

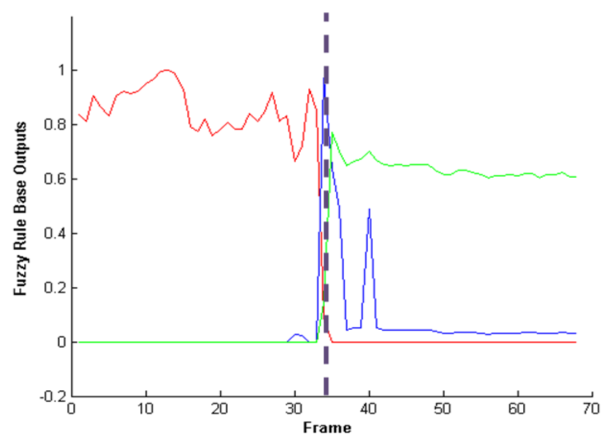


Fig. 6.1. Fuzzy inference outputs for states plotted for a voxel person fall. The x-axis is time, measured in frames, and the y-axis is the fuzzy inference outputs (activity confidence). The red state sequence is **upright**, blue is **in-between**, and green is **on-the-ground**. The frame rate is 3 fps. The purple dashed vertical line is the manually identification of the location of a fall.

Rules, based on both object interaction and human pose, for inferring **on-the-chair** and **lying-on-the-couch** are presented in table 6.4. The respective antecedent linguistic variables and terms, trapezoid membership functions, are reported in table 6.5. These states, which have nothing to do with falling at the moment, are presented in order to show the ability to extend this rule-based approach to recognize new states and ultimately their relevance to recognizing various human activity.

Table 6.4. Rules for recognizing states based on object interaction and pose.

Rule		Chair Overlap	Couch Overlap	Eigen-based Minimum	Normal Similarity		**on-the-chair**	**lying-on-the-couch**
1	IF	*High*				THEN	*Very High*	
2		*Medium*					*Medium*	
3		*Low*					*Very Low*	
4			*High*	*High*	*Low*			*Very High*
5			*Medium*	*High*	*Low*			*Medium*
6			*Low*	*High*	*Low*			*Very Low*

Table 6.5. Antecedent linguistic variables and terms used for the inference of pose and object interaction-based states. Input is clamped to the domain [0,1].

	Terms		
Linguistic Antecedent Variables	*Low:* μ_A^T	*Medium:* μ_A^T	*High:* μ_A^T
Overlap with the couch	(0.0,0.2,0.3,0.4)	(0.2,0.4,0.6,0.8)	(0.7,0.8,1.0,1.0)
Overlap with the chair	(0.0,0.2,0.3,0.4)	(0.2,0.4,0.6,0.8)	(0.7,0.8,1.0,1.0)

6.2.2 Hierarchical Framework for Linguistic Summarization – Summarization

After state is inferred at each frame, linguistic summarization is performed. Ideally, for each state and activity, different parameters would exist for terms defined over the time domain. This way, specific meaning can be assigned to "a *long* time on the ground" versus "a *long* time walking". In this dissertation, the set of proposed states share the same time term fuzzy set parameters, reported in table 6.6. This term set was designed by the nurses with fall detection in mind. The terms are specified according to seconds and are represented as trapezoidal membership functions.

Table 6.6. Terms identified by the nurses for the linguistic variable time duration. Input is clamped to the domain [0, 86400].

Brief: μ_A^T	*Short:* μ_A^T	*Moderate:* μ_A^T	*Long:* μ_A^T
(0.0,1.0,1.0,2.0)	(1.0,5.0,10.0,15.0)	(10.0,120.0,480.0,720.0)	(480.0,900.0,86400.0,86400.0)

Additionally, τ_1 and τ_2, the user defined thresholds for indeterminate maximum state identification and too low of confidence, are empirically assigned the values 0.05 and 0.5 respectively. Thus, the temporal state memberships have to only be slightly distinguishable from one another and the memberships have to be greater than the point of total uncertainty (0.5). The parameter τ_3 was experimentally determined to be $F * 2$, where F is the number of frames captured per second, $F = 3$ in this system. Thus, any summary that is less than two seconds in duration is removed. Lastly, τ_4, the maximum allowable frame gap parameter for merging consecutive summaries with the same state (or activity) label in algorithm 5.1, is selected such that the number of frames equals one second.

6.2.3 Hierarchical Framework for Linguistic Summarization - Activity

After state is inferred and summarized, activity is inferred, summarized and then subsequently used to recognized additional levels of increasing more complex and specific activity. In this dissertation, a two level activity analysis system is used to recognize falls. Thus, the system contains one level of state and one level of activity, $\{\{A_1^2, \dots, A_5^1\}, \{A_1^2, \dots, A_5^2\}\}$. The following activities are investigated here. Their respective terms are reported in table 6.7.

- **Fall** (A_1^2): This activity is generally characterized by voxel person being **on-the-ground** and a quick sudden change in acceleration before going to the ground. A fall rule base is provided by the nurses to address specific fall performances. This rule base does not account for falls in which the human is interacting with objects, such as walkers, chairs, canes, wheelchairs, etc.
- **Standing** (A_2^2): This activity is generally characterized by voxel person being in an **upright** state and not moving very far recently.
- **Walking** (A_3^2): This activity is generally characterized by voxel person having an **upright** state and moving more than some user specified distance recently.
- **Relaxing-on-the-chair** (A_4^2): This activity is generally characterized by voxel person being **on-the-chair** and having an overall low motion.
- **Lounging-on-the-couch** (A_5^2): This activity is generally characterized by voxel person **lying-on-the-couch** and having an overall low motion.

Table 6.7. Terms for the consequent linguistic variables **fall**, **standing**, **walking**, **relaxing-on-the-chair**, and **lounging-on-the-couch**.

Very Low: μ_A^Z	*Low:* μ_A^T	*Medium:* μ_A^T	*High:* μ_A^T	*Very High:* μ_A^S
(0.0,2.0)	(0.0,0.0,0.25,0.5)	(0.0,0.4,0.6,1.0)	(0.5,0.75,1.0,1.0)	(-1,1)

The terms for the linguistic variables summary activity confidence, distance traveled per second in the current summary, and number of seconds to move a fixed distance with respect to the current summary, where $\zeta = 1$ foot and $\varrho = F * 10$, up to 10 seconds, are reported in tables 6.8, 6.9, and 6.10.

Table 6.8. Linguistic variable terms for summary activity confidence ($\pi_{A^{l}_{max}, Sum'_{l,g}}$) with domain [0,1].

Low: μ_A^T	*Medium:* μ_A^T	*High:* μ_A^T
(0.0,0.0,0.2,0.4)	(0.5,0.65,0.8,0.9)	(0.6,0.8,1.0,1.0)

Table 6.9. Linguistic variable terms for distance traveled per second in the current summary ($\varphi^{\vec{c}_t}_{Sum'_{l,g}}$).

Input is clamped to the domain [0,2].

Low: μ_A^T	*Medium:* μ_A^T	*High:* μ_A^T
(0.0,0.0,0.1,0.2)	(0,0.1,0.5,0.6)	(0.4,0.5,2.0,2.0)

Table 6.10. Linguistic variable terms for number of seconds taken to move a fixed distance with respect to the current summary ($\vartheta_{1,30}^{\vec{c}^{Legs}_{t,(x,y)}}$). Input is clamped to the domain [0,10].

Low: μ_A^T	*Medium:* μ_A^T	*High:* μ_A^T
(0.0,0.0,4.0,6.0)	(4.0,6.0,7.0,9.0)	(7.0,9.0,10.0,10.0)

The parameter W, the window size for detecting a large recent change in voxel person's speed before the start of a summary, was experimentally picked to be 40, approximately 13 seconds. Elements in this window are smoothed with a mean filter of size Ψ =5. The terms for the linguistic variable recent relative difference change in voxel person's speed are reported in table 6.11.

Table 6.11. Linguistic variable terms for recent relative difference change in voxel person's speed. The minimum of the feature and the value 1 is computed. Input is clamped to the domain [0,1].

Low: μ_A^T	*Medium:* μ_A^T	*High:* μ_A^T
(0.0,0.0,0.1,0.2)	(0.0,0.1,0.2,0.3)	(0.2,0.3,1.0,1.0)

The final feature is the search backwards in time from the current time step up to a user defined maximum amount of time for oscillating behavior between the states **on-the-ground** and **in-between**, $OSC_{moderate,A_2^1,A_3^1}$. This feature assists with detecting if a resident has fallen and is trying to make it back up. The nurses *moderate* time domain term, specifically the trapezoid value d, is used to bound the search back in time. Terms, reported in table 6.12, for recent oscillating behavior between **on-the-ground** and **in-between** are *low*, *medium*, and *high*. In addition, table 6.13 is the rule set, provided by nurses, used to infer falls.

Table 6.12. Linguistic variable terms for recent oscillating behavior, $OSC_{moderate,A_2^1,A_3^1}$. Input is clamped to the domain [0,8].

Low: μ_A^T	*Medium:* μ_A^T	*High:* μ_A^T
(0.0,0.0,2.0,4.0)	(1.0,3.0,5.0,7.0)	(4.0,6.0,8.0,8.0)

Table 6.13. Rules for **fall** recognition, created in collaboration with the nursing staff.

Rule		**on-the-ground**	Time Duration	Change in Speed Before the Current Summary	Distance Traveled Per Second in the Current Summary	Oscillation Between **on-the-ground** and **in-between**		**fall**
1	If	*High*	*Moderate*	*High*			Then	*Very High*
2		*Medium*	*Moderate*	*High*	*Low*			*High*
3		*High*	*Short*	*High*				*High*
4		*Medium*	*Short*	*High*				*High*
5		*High*	*Long*					*Very High*
6		*Medium*	*Long*					*Very High*
7		*High*	*Moderate*		*Low*			*Very High*
8		*High*	*Moderate*			*High*		*Very High*
9		*High*	*Moderate*		*High*			*High*
10		*High*	*Short*			*High*		*High*
11		*Medium*	*Moderate*		*Low*			*High*
12		*Medium*	*Short*			*High*		*High*
13		*High*	*Short*			*Medium*		*Medium*
14		*Medium*	*Short*			*Medium*		*Medium*

For sake of discussion, the rules in table 6.13 are grouped into different sets based on the severity and type of fall. The rules highlighted in blue (1-6) are focused on the detection of an impact at the moment the human went to the ground or if the human is on the ground for *too long* of a time. The idea of detecting falls by observing an impact and transition from upright to on the ground is the standard approach. The rules highlighted in brown (7, 9, and 11) and orange (8, 10, 12-14) are designed with a frail elder in mind. That is, someone who is not expected to be on the ground performing activities such as stretching or exercising. If it is possible for an elder to be on the ground performing such activities, then these rules should be omitted because they can lead to false alarms. This is not perceived as a failure of the system. Instead, it is an example of the flexibility at the rule level given knowledge about a resident's physical or cognitive capability. It provides for customization of a fall detection system to a particular person. Other fall scenarios involving objects, such as a chair, walker, etc, are not included for recognition at the moment. The rules used to infer the other proposed non-fall activities are reported in tables 6.14 and 6.15.

Table 6.14. Rules used for recognizing **standing** and **walking**.

Rule		Number of Seconds Taken to Move a Fixed Distance	**upright**		**standing**	**walking**
1	If	*Low*	*High*	Then	*Very High*	
2		*Low*	*Medium*		*Medium*	
3		*Medium*	*High*		*Low*	
4		*Medium*	*Medium*		*Very Low*	
5		*Low*	*High*			*Low*
6		*Low*	*Medium*			*Low*
7		*Medium*	*High*			*High*
8		*Medium*	*Medium*			*Medium*
9		*High*	*High*			*Very High*
10		*High*	*Medium*			*High*

Table 6.15. Rules used for recognizing **relaxing-on-the-chair** and **lounging-on-the-couch**.

Rule		Number of Seconds Taken to Move a Fixed Distance	**on-the-chair**	**lying-on-the-chair**		**relaxing-on-the-chair**	**lounging-on-the-couch**
1	If	*Low*	*High*		Then	*Very High*	
2		*Low*	*Medium*			*High*	
3		*Low*		*High*			*Very High*
4		*Low*		*Medium*			*High*

6.2.4 Hierarchical Framework for Linguistic Summarization – Conflict Resolution

The proposed system infers activity at each frame. There is no reason why multiple activities at the same level cannot all be inferred with a high confidence. For example, when a person is **on-the-chair** they could also be **in-between**. For sake of reporting, it may be acceptable to report all such activities.

However, in other settings it is important to override or rank the decisions based on context or priority. For example, being **in-between**, e.g. hunched over, might be subsumed by being **on-the-chair**. The temporal linguistic summarization procedure presented in this dissertation, algorithm 5.1, assumes that only a single activity is observed with a high confidence at any one moment in time. Otherwise, there is indeterminate maximum activity membership identification and segmentation fails due to ambiguity. If necessary, this matter can be addressed in future activity segmentation research. However, in this work, one activity is always been preferred to the other. Conflict resolution rules are used to eliminate ambiguity according to priority or preference. Table 6.16 is the set of rules used in this work to decrease the confidence in the state **in-between** when the human is either **on-the-chair** or **lying-on-the-couch**.

Table 6.16. Inhibitory rules used for conflict resolution.

Rule		Chair Overlap	Couch Overlap		**in-between**
1	IF	*High*		THEN	*Very Low*
2		*Medium*			*Very Low*
3			*High*		*Very Low*
4			*Medium*		*Very Low*

6.2.5 Hierarchical Framework for Linguistic Summarization - Alert Generation

In operation, a subset of activities are monitored for adverse event detection. In this dissertation, an alert is generated if a **fall** summary is ever produced with a confidence greater than the user defined threshold amount τ_5. The value τ_5 used here is derived from the **fall** term *high*. In addition, a minimum acceptable membership degree of 0.5 for *high* is used. The location of the centroid for this resultant set is 0.78, thus $\tau_5 = 0.78$.

6.3 Hidden Markov Models

The activity set used by the HMM approach is the same as that proposed above, $\{A_1^2, \ldots, A_5^2\}$. A single model is constructed per activity. The training data is manually partitioned in order to obtain the activity sequences. The search for the best model structure, number of states and mixtures, is conducted manually in an exhaustive fashion. All features are included in each model. The features are the same set used by our proposed soft-computing approach. At each frame, all activity models are used to classify the new sequence and the most likely model is selected as the winner. Different time windows, Γ, were attempted. A value of $\Gamma = 3\mathrm{F}$, three seconds, was empirically discovered to work the best.

6.4.1 Data Sets - Introduction

All data is captured in the Computational Intelligence engineering research laboratories at the University of Missouri. No elderly fall data exists and none can be acquired due to the age of the individuals and the risk of injury. The first fall data set is captured using students as subjects. The second data set is obtained using a stunt actor coached by our nursing team. The stunt actor is trained according to the typical way, speed, and manner, that elderly activities are performed.

6.4.2 Data Sets - Student Data Set

Data set **SET1** is collected using two cameras with orthogonal placement. This set is eighteen sequences and 3,359 frames, approximately nineteen minutes of video. Figure 6.2 is a few images showing the viewing conditions.

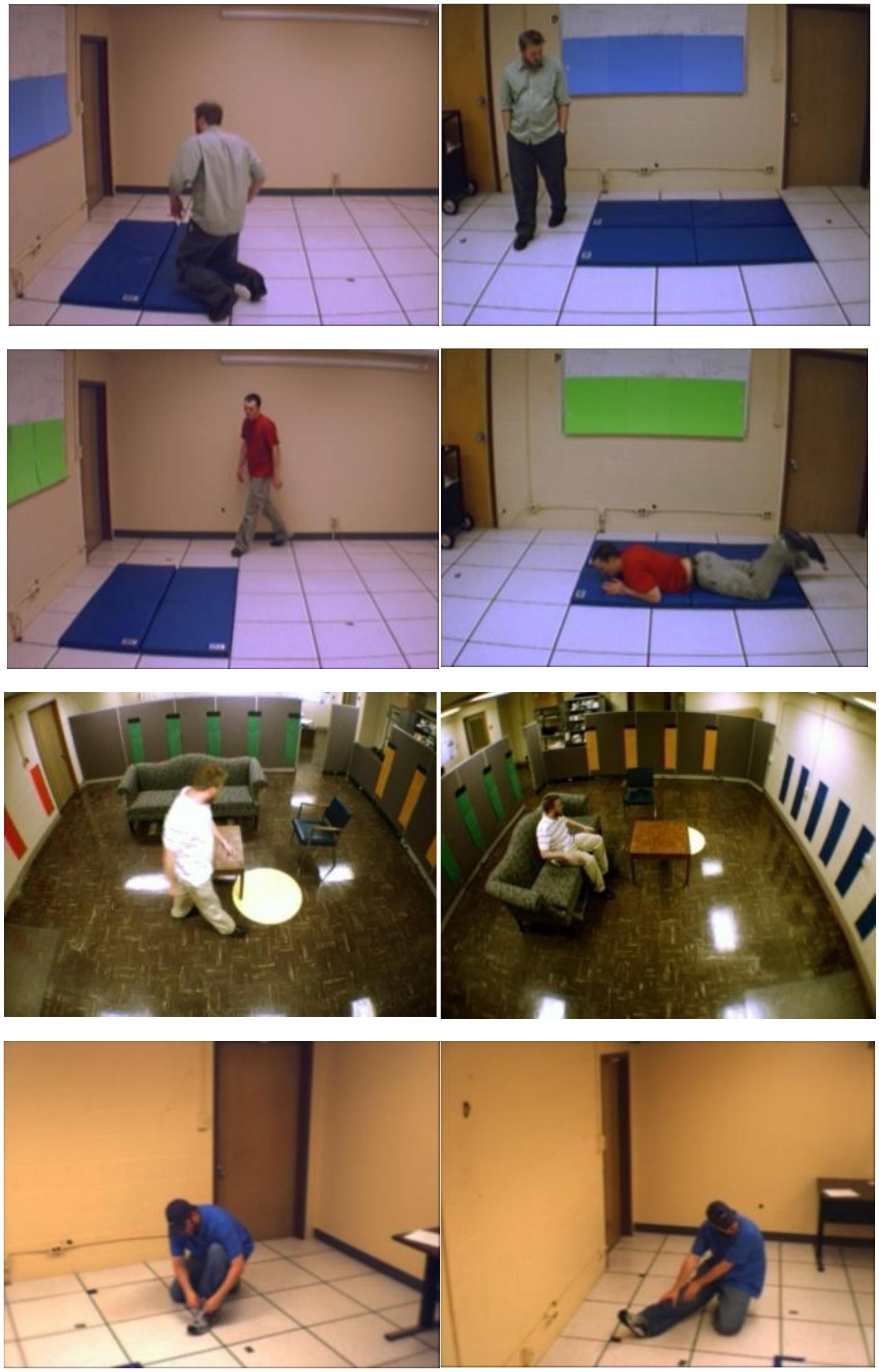
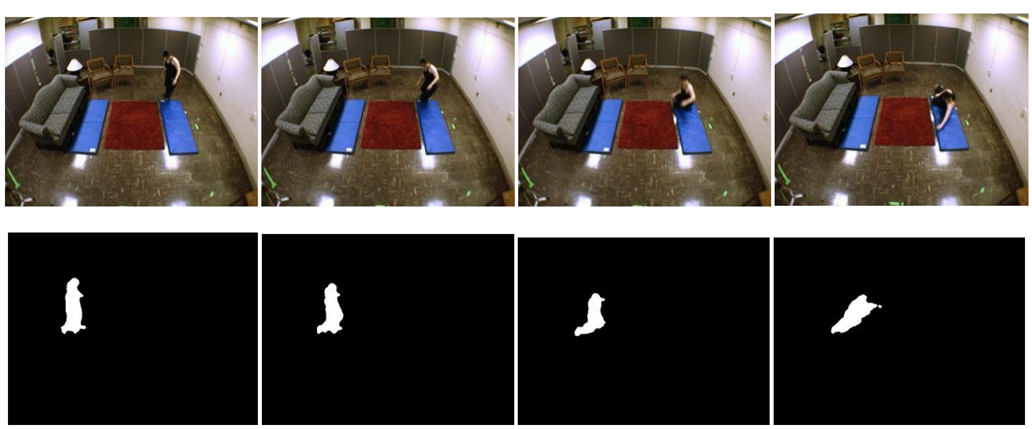


Fig. 6.2. Images from the student data set. Rows 1 and 2 are **SET1.a**, row 3 is **SET1.b** and row 4 is **SET1.c**.

Data set **SET1** is partitioned into three categories: **SET1.a**, falls only, **SET1.b**, common household activity, such as standing, walking, sitting on the couch, chairs, etc, and **SET1.c**, a longer sequence with falls along with potential false alarm activities, such as stretching, kneeling down to tie ones shoes, tripping and getting back up, etc. **SET1.a** contains sixteen sequences while **SET1.b** and **SET1.c** are one long video sequence each. A variety of falls are performed. This includes falling forward, backwards, and to each side. Scenarios include falls lasting a couple of seconds, after which the person gets back up, falls where the person stays down on the ground but attempts to get back up, and falls where a person simulates a severe injury by lying on the ground motionless.

6.4.3 Data Sets - Stunt Actor Data Set

Data set **SET2** also uses two cameras in a similar configuration to the student collection. Figure 6.3 shows the environment, camera viewing conditions, and example silhouettes and voxel person during a fall. This set is ten sequences and 20,987 frames, approximately two hours of video.



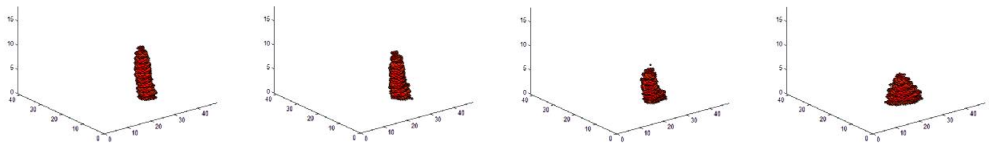


Fig. 6.3. Example raw images (row 1), silhouettes (row 2), and voxel person (row 3) for a fall from the stunt actor data set.

A fall protocol is designed by the nurses to guide the stunt actor activity performance[16]. Each activity is performed multiple times and the time spent on the ground varies. The different types of falls performed are resident looses balance (**SET2.a**), momentary loss of consciousness (**SET2.b**), and tripping and slipping falls (**SET2.c**). Potential false alarm activities are performed during the fall sequences. Examples include walking, standing, tripping and getting back up quickly, going to the ground in a safe fashion and exercising and lying down (figure 6.4). In addition, separate potential false alarm sequences, in which there are no falls, are collected (**SET2.d**).

Fig. 6.4. Example potential false alarm activities from **SET2.d**.

6.5 Evaluation Metrics

All data is hand segmented in order to acquire a ground truth for comparison against the proposed computational systems. The start and end frames for all activities are identified. This allows for both frame and interval-based statistics. The following metrics are used. Figure 6.5 shows the different ways to measure compatibility between system and human decisions.

- **Metric 1**: Each system generated state (activity) summary is associated with the human's state (activity) interval that it overlaps with the most. A classification matrix is then generated. This measure is computed for both confidence values τ_2 and τ_5.
- **Metric 2**: Matching between frame-by-frame state (activity) decisions, according to the system state (activity) with the maximum membership grade at each frame and the human ground truth frame decisions. This measure is computed for both confidence values τ_2 and τ_5.
- **Metric 3**: Number of system generated summaries per activity divided by their respective number of related frames. This value is the percentage of original decisions remaining after summarization. The lower the value the better.
- **Metric 4**: This metric is the most complex. Intuitively, this metric indicates if the system can produce a reduced acceptable number of summaries. It is an analysis of what happens when the criteria for comparing activity is varied at the summary level. First, all system summaries are associated with the human designated interval which they overlap the most. For each of the following, the two consequent domain confidences, τ_2 and τ_5, are considered. This metric determines success if the system identified at least one and up to a user defined maximum number of correct summaries. Additionally, the sum of the lengths of the correctly classified summaries must account for a user specified percentage of the humans labelled length. Both of these conditions are varied, specifically $\{1,2,5,10\}$ for the number of intervals and $\{0.1,0.25,0.5,0.75,0.9\}$ for overlap percentage.

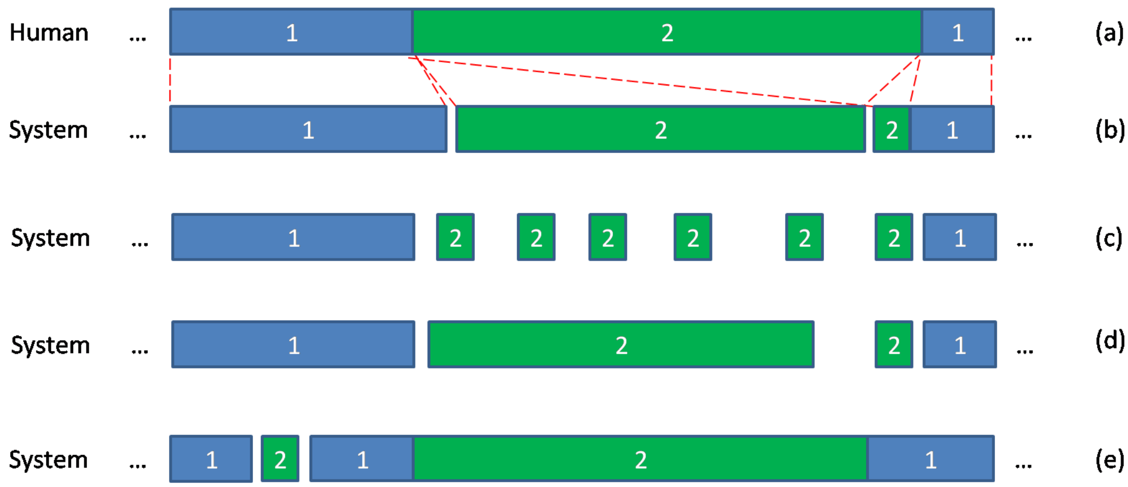


Fig. 6.5. Approach taken to measuring similarity between human ground truth intervals and system summaries. The x-axis is time. The summaries of interest are shown in green. Intervals in (a) are ground truth and (b) shows each system summary mapped to the human interval it overlaps with the most. In (c), all relevant summaries are contained in the human's interval, label 2, but an undesirable number are present. In (d), all summaries are correctly contained in the human's interval, there are not *very many* of them, and together they account for a *good* percentage of the human's total interval. In (e), the first green interval is a false alarm and the second green interval is a correct match. Cases (b), (d) and (e) are positive Metric 4 scenarios.

6.6 Ground Truth

There are 24,346 frames in {**SET1.a, SET1.b**, **SET1.c**, **SET2.a**, **SET2.b**, **SET2.c**, **SET2.d**}. This is approximately 8,115 seconds or 136 minutes of video data. In addition, 12 states and activities are tracked. Due to the large number of frames and activities, the frame-by-frame human ground truth is provided by myself. Future work includes using a larger group of individuals to hand segment the ground truth. Discrepencies are expected to exist in such work and future work will need to also include experiments about the ability of a group of humans to provide similar labels and methods to compare variation between the system and such variation in a groupd of labelers. In the ground truth, the start and end frames for all activities being tracked are identified. Tables 6.17 and 6.18 are the number of human state and activity labels for all data sets.

Table 6.17. Number of state labels for all data sets.

upright	**in-between**	**on-the-ground**	**lying-on-the-couch**	**on-the-chair**	unknown
90	102	41	2	7	7

Table 6.18. Number of actiivty labels for all data sets.

fall	**standing**	**Walking**	**lounging-on-the-couch**	**relaxing-on-the-chair**	unknown
31	16	113	2	8	81

For state, **upright** and **in-between** occur the most often, 90 and 102 respetively. The majority of **upright** intervals contain known activities, **standing** or **walking**. The vast majority of **in-between** intervals are either brief transition periods, such as moving between **upright** and **on-the-ground**, or during false alarm activities. The state **on-the-ground** had fewer samples, 41. This is mainly because each **on-the-ground** event lasts a fair amount of time in order to simulate a fall. The states **lying-on-the-couch** and **on-the-chair** occurr very infrequently, two and seven times respetively. They are included to demonstrate the general extendability of the system based on location and/or object information instead of just pose. These states are also included in order to provide a few additional activities to compare the fall recognition system against. Lastly, only seven intervals of type unknown state is identified. Most occurred during state transition. The majority of human labeled unknown state intervals are ignored by this system because they are too short, $|Sum_g| < \tau_3$, in time duration. Only 307 frames make up unknown state, approximately 1 minute.

For activity, many **walking** events are observed, 113. This is due to the obvious reason that it is the primary mechanism by which humans move through a scene and transition between different activities. Sixteen **standing** events, 31 **fall** events, 2 **lounging-on-the-couch** events, and 8 **relaxing-on-the-chair** events are observed. The more important quantity is the number of unknown activity, 81, which is 5,135 frames, over seventeen minutes.

It is asserted here that the primary culprit behind observing such few state intervals of type unknown, relative to the number observed for activity, is the definition of the pose-based states **upright**, **in-between** and **on-the-ground**. An attempt is made in this work to essentially partition human pose according to the height of an individual. The result is few occurrences of unknown state. The reason is that if the human is **upright**, there should be no confidence in a fall. Falls are primarily based on charesterics of **on-the-ground** intervals. The state **in-between** was proposed to help detect some instances where someone might be attempting to get back up after a fall and they have yet to make it succsfully back up to an **upright** state. The proposed set of activities are more disjoint in definition and it is believed that more unknowns are observed as a result of this.

6.7 Summarization of Unknown Activity

Membership time series segmentation algorithm 5.1 results in frames not assigned a known activity label. Discluding the merging steps in algorithm 5.1, frames not assigned a known activity label are due to the maximum membership at time t not having a large enough membership value or the maximum membership value is not clearly distinguishable from, i.e. greater than, the other membership values. Let $I_U = \{i_1, \dots, i_{N''}\}$ be the set of frame indicies with no known activity label, where $i_j < i_{j+1}$.

The set I_U is partitioned into intervals, i.e. summaries of unknown activity, according to $(i_j + 1) \neq i_{j+1}$.

This procedure is required to compare human ground truth and system unknown activity.

Results and Discussion

7.1 Introduction

The activity analysis systems put forth in this dissertation are compared to ground truth. The main goal is to understand how well each system measures up to the gold standard in this domain, humans. A secondary goal is to observe the similarities and differences in the two automated systems. No single metric definitively illustrates all advantages or deficiencies. It is the combined performance exhibited by the systems for all metrics that is ultimately important. An example is performance at both the frame-by-frame and summary levels. According to the metrics developed, perfect summary matching can be reached without requiring perfect frame-by-frame matching. Human and system summaries might overlap for the most part, meaning the exact frame-by-frame alignment might be off by one, two or more frames. It is important to analyze the system and its tendencies under various alignment conditions. Individual key points are discussed as encountered per section and metric and the global picture is assembled in this chapter's conclusion section.

As discussed in chapter 5, HMM-based activity is inferred at each frame using a fixed size sliding time window. At each frame, the fixed size sliding time window is the observation sequence input to the HMMs. The Viterbi algorithm [78] is a procedure designed to compute the most likely state sequence for a single model and single observation sequence. Thus, the Viterbi algorithm is valid for a specific activity during a single fixed size sliding time window. In addition, all HMMs are not forced to include the same states. Even more importantly, state as perceived by a human, i.e. upright, does not have to be represented using a single HMM state or a single probability density function. Overall, it is unclear how to exactly compute the frame-by-frame state for the proposed fixed size sliding time window approach using HMMs and a procedure such as the Viterbi. As a result of this, state is explicitly analyzed in this chapter only for the linguistic summarization system, not the HMM system. However, activity is compared against the human ground truth for both systems. There is no difficulty in directly comparing activity at the frame-by-frame level, an output of both systems, however summarization algorithm 5.1 is required in addition to frame-by-frame HMM activity recognition to generate activity summaries for comparison.

7.2 State

The purpose of this section is to analyze the similarity between inferred state and the human ground truth. Figure 7.1 is a plot of inferred state for a stunt actor fall sequence, **SET2.a**. An example summarization interval is frames ten to approximately 1300 for the **on-the-ground** state, shown in green. In this interval, the human fell.

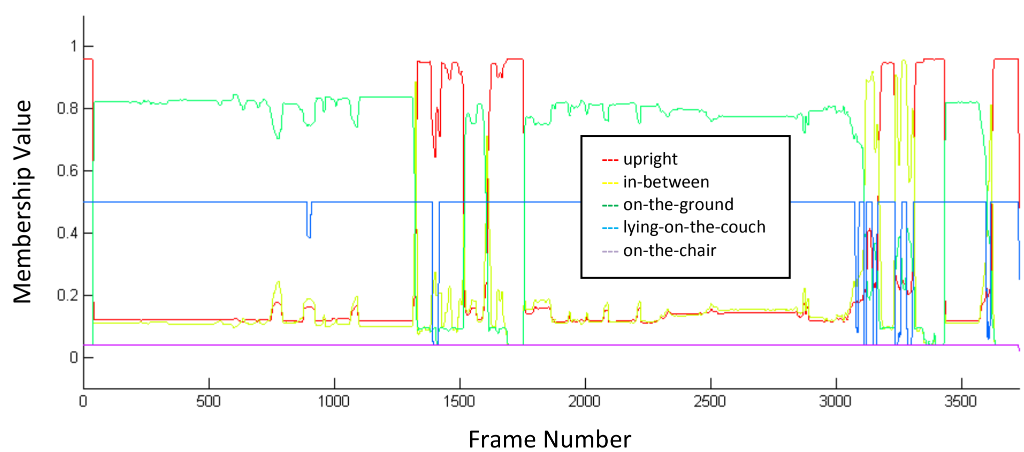


Fig. 7.1. Plot of inferred state for the stunt actor fall data set **SET2.a**. The state **upright** is red, **in-between** is yellow, **on-the-ground** is green, **lying-on-the-couch** is blue, and **on-the-chair** is purple.

Table 7.1 is metric 1 for state. That is, each system generated summary is associated with the human activity interval that it overlaps with the most. The system does not always produce the same number of summaries as the human. Table 7.1 is not a confusion matrix. In addition, for different confidence thresholds, i.e. τ_2 and τ_5, the summary merging steps in the membership time series segmentation algorithm 5.1 will result in different numbers of total reported summaries.

Table 7.1. Comparison between state summaries for the linguistic summarization system, for summary confidences τ_2 and τ_5, and the human ground truth for all data sets.

		Human Ground Truth					
		upright (90)	**in-between** (102)	**on-the-ground** (41)	**lying-on-the-couch** (2)	**on-the-chair** (7)	unknown (7)
Linguistic Summarization System Each cell contains $\pi_{A_i^l, Sum'_{l,g}} > \tau_2$ $\pi_{A_i^l, Sum'_{l,g}} > \tau_5$	**upright**	103 101	0 0	0 0	0 0	0 0	0 0
	in-between	1 0	74 8	1 0	0 0	0 0	0 0
	on-the-ground	1 0	17 0	44 27	0 0	0 0	0 0
	lying-on-the-couch	0 0	0 0	0 0	2 2	0 0	0 0
	on-the-chair	1 1	0 0	0 0	0 0	7 7	0 0
	unknown	1 3	22 49	2 13	0 0	0 0	1 1

Table 7.1 is the number of times that the linguistic summarization system and the human agree and disagree about state summarization. The number of human identified intervals per state is reported for each column next to the state label in table 7.1. Each classification matrix cell has two criteria for minimum acceptable summary confidence, $\pi_{A_i^l, Sum'_{l,g}} > (\tau_2 = 0.5)$ and $\pi_{A_i^l, Sum'_{l,g}} > (\tau_5 = 0.78)$. The matrix diagonal, correct classification, is highlighted in blue. Incorrect decisions in which the system

called an event a known state while the human called the event unknown, last column, is highlighted in brown. Incorrect decisions in which the system said unknown but the human called it a known state, last row, is highlighted pink. Cells that are not shaded are miss-classified summaries of system known state.

Overall, metric 1 shows that the states **upright**, **lying-on-the-couch**, and **on-the-chair** perform the best. Table 7.1 also shows that **in-between** has the most mistakes, followed by **on-the-ground**. No human labeled unknown state is classified as a system known state. However, some human labeled known state, **upright**, **in-between**, and **on-the-ground,** is occasionally incorrectly called unknown by the system. Table 7.2, the false alarm data set **SET2.d**, reveals the source of many of these problems.

Table 7.2. Comparison between state summaries for the linguistic summarization system, for summary confidences τ_2 and τ_5, and the human ground truth for the false alarm data set **SET2.d**.

		Ground Truth					
		upright (62)	**in-between** (54)	**on-the-ground** (26)	**lying-on-the-couch** (1)	**on-the-chair** (7)	unknown (3)
Linguistic Summarization System Each cell contains $\pi_{A_i^l, Sum_{l,g}'} > \tau_2$ $\pi_{A_i^l, Sum_{l,g}'} > \tau_5$	**upright**	62 60	0 0	0 0	0 0	0 0	0 0
	in-between	1 0	53 3	0 0	0 0	0 0	0 0
	on-the-ground	1 0	16 0	9 2	0 0	0 0	0 0
	lying-on-the-couch	0 0	0 0	0 0	1 1	0 0	0 0
	on-the-chair	1 1	0 0	0 0	0 0	6 6	0 0
	unknown	1 3	20 38	1 6	0 0	0 0	0 0

In Table 7.2, three human labeled unknown intervals are not recognized by the system. This means the system never generated a summary of label unknown that overlapped with the human's labeled unknown interval the most. The human labeled unknown state intervals were very brief in time. Additionally, the pose-based states **upright**, **in-between**, and **on-the-ground** effectively partition the objects height domain, thus little unknown state is observed. Activity is a different story. A total of 81 relatively long in time unknown intervals are recorded.

Table 7.2 shows that a significant majority of state recognition error is due to activity unknown to the system. This means the system definitions are good for the task at hand, fall recognition, but the terms are not as compatible with a human's broader understanding of them for other non-fall activities. Thus, the states **in-between** and **on-the-ground** are not as similar to a human's understanding when an activity such as exercising is considered. However, the activity recognition rates, i.e. fall recognition, are shown later to still do very well in light of this wider reaching concept definition compatibility matter. In particular, the state **in-between** is rather ill-defined. While **in-between** has helped with the recognition of falls, the primary goal of this work, there appears to be significant disagreement in the broad regard outside of fall detection between the human and system for this state. Tables 7.1 and 7.2 show that this phenenomon is also present for **on-the-ground**, but to a lesser degree.

Table 7.3, metric 2, is the similarity between frame-by-frame state decisions inferred by the linguisic summarization system and the human ground truth. Again, two system decision criteria are considered, $\mu_{t,max}^1 > \tau_2$ and $\mu_{t,max}^1 > \tau_5$, where $\mu_{t,max}^1 = maximum_k\left(\mu_{t,k}^1\right)$. The "raw" classification matrix is $M(i,j)$, for $i,j \in \{1,\dots,6\}$. For example, $M(2,2)$ is the number of times the system called a frame **in-between** and the human called the frame **in-between**. Reporting raw matching scores is hard to understand, especially when thousands of frames are involved. It helps to have a normalization number. Value is added here by analyzing $M(i,j)$ in the context of the number of system decisions, specifically $M(i,j)/\sum_{j=1}^{6} M(i,j)$. That is, when the system says activity i, how often is it correct?

Table 7.3. Frame-by-frame state decision comparison between the linguistic summarization system and the human ground truth for all data sets and $M(i,j)/\sum_{j=1}^{6} M(i,j)$.

		Ground Truth					
		upright	**in-between**	**on-the-ground**	**lying-on-the-couch**	**on-the-chair**	unknown
Linguistic Summarization System Each cell contains $\mu_{t,max}^{1} > \tau_2$ $\mu_{t,max}^{1} > \tau_5$	**Upright**	0.97 0.98	0.03 0.02	0.00 0.00	0.00 0.00	0.00 0.00	0.00 0.00
	in-between	0.01 0.00	0.98 0.99	0.01 0.00	0.00 0.00	0.00 0.00	0.00 0.00
	on-the-ground	0.01 0.00	0.12 0.00	0.86 1.00	0.00 0.00	0.00 0.00	0.01 0.00
	lying-on-the-couch	0.00 0.00	0.01 0.00	0.00 0.00	0.93 0.97	0.00 0.00	0.06 0.00
	on-the-chair	0.07 0.07	0.00 0.00	0.00 0.00	0.00 0.00	0.93 0.93	0.00 0.00
	Unknown	0.06 0.18	0.64 0.53	0.05 0.26	0.00 0.00	0.00 0.00	0.24 0.03

Due to the large number of metrics and subsequent tables presented in this chapter, it was determined that no value was added by the inclusion of $M(i,j)/\sum_{i=1}^{6} M(i,j)$. That matrix only reinforces the earlier claims put forth in response to tables 7.1 and 7.2. No new conclusions were reached. Table 7.3 shows that the system is very accurate when it makes a decision, except for the class unknown.

Table 7.4, metric 3, is the amount of information reduction according to the number of summaries produced by the system per state divided by their respective total number of origional frames per state. A value of one is no information reduction, while a value of 0.01 is 1% of the origional number of decisions remaining. Therefore, lower numbers are better. Table 7.4 shows that summarization results in drastic information reduction.

Table 7.4. Percentage of frame-by-frame decisions remaining after linguistic summarization of state for all data sets. The lower the value the better the system performance.

	upright	**in-between**	**on-the-ground**	**lying-on-the-couch**	**on-the-chair**	unknown
$\pi_{A_i^l,Sum'_{l,g}} > \tau_2$	0.010	0.018	0.007	0.010	0.008	0.081
$\pi_{A_i^l,Sum'_{l,g}} > \tau_5$	0.010	0.002	0.003	0.010	0.009	0.208
human	0.009	0.025	0.005	0.010	0.007	0.023

Now consider metric 4. This metric indicates if the system is capable of producing a reduced acceptable number of correctly time aligned summaries. The criteria for maximum number of allowable correct system generated summaries and the required amount of human ground truth interval overlap are varied. Table 7.5 is the matrix cell format for tables 7.6 and 7.7.

Table 7.5. Maxtirx cell format for tables 7.6 and 7.7. The left column is for $\pi_{A_i^l,Sum'_{l,g}} > \tau_2$ and the right column is for $\pi_{A_i^l,Sum'_{l,g}} > \tau_5$. While tables 7.6 and 7.7 include different combinations of activities, depending on what is present in the data set, the order and color of entries does not change.

upright for $\pi_{A_i^l,Sum'_{l,g}} > \tau_2$	**upright** for $\pi_{A_i^l,Sum'_{l,g}} > \tau_5$
in-between for $\pi_{A_i^l,Sum'_{l,g}} > \tau_2$	**in-between** for $\pi_{A_i^l,Sum'_{l,g}} > \tau_5$
on-the-ground for $\pi_{A_i^l,Sum'_{l,g}} > \tau_2$	**on-the-ground** for $\pi_{A_i^l,Sum'_{l,g}} > \tau_5$
lying-on-the-couch for $\pi_{A_i^l,Sum'_{l,g}} > \tau_2$	**lying-on-the-couch** for $\pi_{A_i^l,Sum'_{l,g}} > \tau_5$
on-the-chair for $\pi_{A_i^l,Sum'_{l,g}} > \tau_2$	**on-the-chair** for $\pi_{A_i^l,Sum'_{l,g}} > \tau_5$
unknown for $\pi_{A_i^l,Sum'_{l,g}} > \tau_2$	unknown for $\pi_{A_i^l,Sum'_{l,g}} > \tau_5$

Table 7.6. Variation of metric 4 matching success criteria for the linguistic summarization system and the human ground truth for all data sets.

		Required Overlap (Percentage) with the Human Interval									
		0.1		0.25		0.5		0.75		0.9	
Maximum Number of Correct System Generated Summaries	1	0.91	0.89	0.91	0.89	0.91	0.89	0.88	0.86	0.86	0.83
		0.54	0.07	0.5	0.06	0.44	0.06	0.34	0.02	0.17	0
		0.66	0.32	0.66	0.32	0.66	0.32	0.63	0.29	0.56	0.27
		1	1	1	1	1	1	1	1	1	1
		1	1	1	1	1	1	1	1	1	1
		0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0	0
	2	0.94	0.92	0.94	0.92	0.94	0.92	0.91	0.89	0.88	0.86
		0.59	0.07	0.54	0.06	0.47	0.06	0.35	0.02	0.17	0
		0.68	0.32	0.68	0.32	0.68	0.32	0.66	0.29	0.59	0.27
		1	1	1	1	1	1	1	1	1	1
		1	1	1	1	1	1	1	1	1	1
		0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0	0
	5	0.94	0.92	0.94	0.92	0.94	0.92	0.91	0.89	0.88	0.86
		0.61	0.07	0.57	0.06	0.49	0.06	0.35	0.02	0.17	0
		0.68	0.32	0.68	0.32	0.68	0.32	0.66	0.29	0.59	0.27
		1	1	1	1	1	1	1	1	1	1
		1	1	1	1	1	1	1	1	1	1
		0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0	0
	10	0.94	0.92	0.94	0.92	0.94	0.92	0.91	0.89	0.88	0.86
		0.61	0.07	0.57	0.06	0.49	0.06	0.35	0.02	0.17	0
		0.68	0.32	0.68	0.32	0.68	0.32	0.66	0.29	0.59	0.27
		1	1	1	1	1	1	1	1	1	1
		1	1	1	1	1	1	1	1	1	1
		0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0	0

Table 7.6 shows that for a weak criteria, such as up to ten allowable number of correct system generated summaries and 10% required overlap with the human interval, the system performs well for the states **upright**, **lying-on-the-couch**, and **on-the-chair**. Cells highlighted in red are considered to be realistic criteria for evaluating overall system success. The cell highlighted in yellow is one-to-one coorespondence and near perfect overlap. What table 7.6 shows is that **in-between** and **on-the-ground** do not perform well overall. While their respective definitions need improving to better match that of a human, later activity analysis results show that they do still lead to good fall detection results. Table 7.7, metric 4 for {**SET2.a**, **SET2.b**, **SET2.c}**, shows that **in-between** still performs poorly overall while **on-the-**

ground improves for the stunt actor fall data set. This reinforces the claim that the significant reason for performance degregation resides in activity unknown to the system.

Table 7.7. Variation of the metric 4 matching success criteria for the linguistic summarization system and the human ground truth for the stunt actor fall data set, {**SET2.a**, **SET2.b**, **SET2.c**}. Only **upright**, **in-between**, and **on-the-ground** are shown because the other states are not performed for these data sets.

		Required Overlap (Percentage) with the Human Interval									
		0.1		0.25		0.5		0.75		0.9	
Maximum Number of Correct System Generated Summaries	1	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94
		0.69	0.25	0.63	0.25	0.38	0.25	0.13	0.06	0	0
		0.85	0.69	0.85	0.69	0.85	0.69	0.77	0.62	0.77	0.62
	2	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94
		0.69	0.25	0.63	0.25	0.38	0.25	0.13	0.06	0	0
		0.93	0.69	0.93	0.69	0.93	0.69	0.85	0.62	0.85	0.62
	5	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94
		0.69	0.25	0.63	0.25	0.38	0.5	0.13	0.03	0	0
		0.93	0.69	0.93	0.69	0.93	0.69	0.85	0.62	0.85	0.62
	10	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94
		0.69	0.25	0.63	0.25	0.8	0.25	0.13	0.06	0	0
		0.93	0.9	0.93	0.9	0.93	0.69	0.85	0.62	0.85	0.62

In summary, significant information reduction for state is observed using the membership time series segmenation algorithm. Additionally, **upright**, **on-the-chair**, and **lying-on-the-couch** perform well in general, while **in-between** and **on-the-ground** are less compatible with a human's broad understanding of these states, specifically for activity that is unknown to the system. Few unknown states are observed as a result of the partitioing of the height domain due to **in-between**, **on-the-ground**, and **upright**. It is hard to tell how well the system performs in recognizing unknown state. The following results for activity shed more light on this matter. While state has led to good fall (activity) recognition results, the **on-the-ground** definition needs to revisited in order to scale better to other non-fall related activity. Lastly, **in-between** has been of use with respect to fall detection, but the state may need to be removed and replaced with a set of more specific and less ill-defined states in the future.

7.3 Activity

The purpose of this section is to analyze the similarity between inferred activity and the human ground truth. In addition, the linguistic summarization system is compared to the HMM-based activity analysis system. Figure 7.2 is a plot of inferred state and corresponding inferred activity for the stunt actor fall sequence **SET2.a**. An example summarization interval is frames ten to approximately 1300. During this interval, the state is **on-the-ground** and the activity is **fall**.

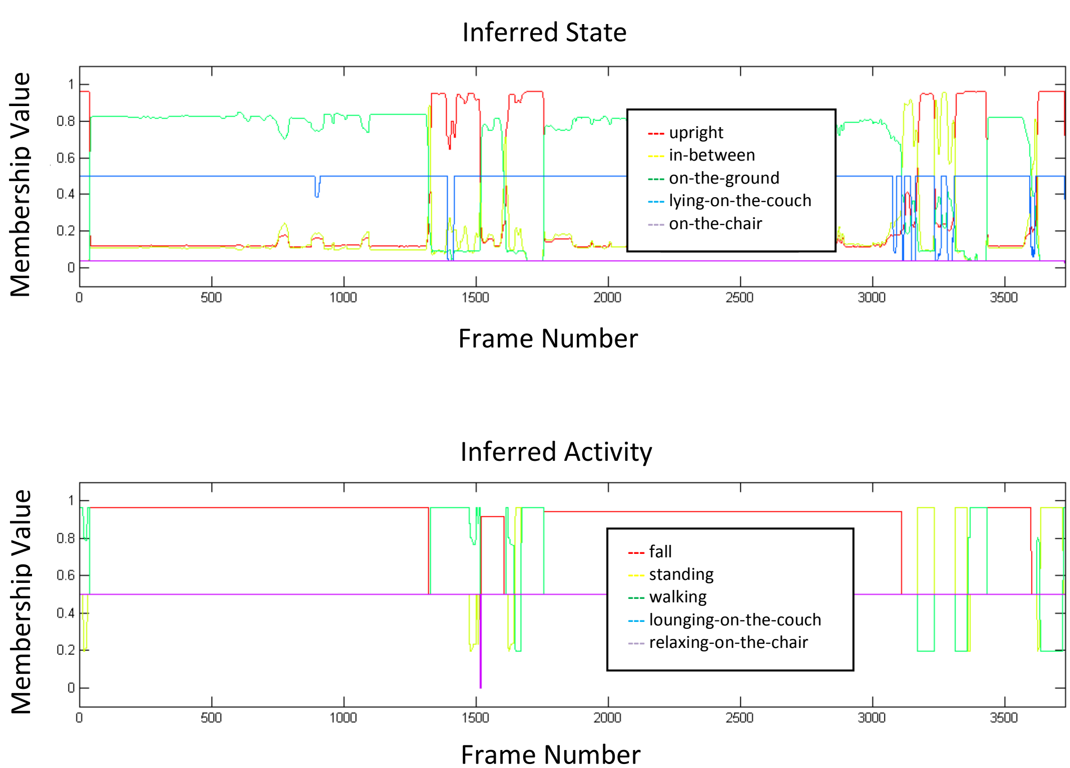


Fig. 7.2. Plot of inferred state and inferred activity for the stunt actor fall data set **SET2.a**. In the top image, the state **upright** is red, **in-between** is yellow, **on-the-ground** is green, **lying-on-the-couch** is blue, and **on-the-chair** is purple. In the bottom image, the activity **fall** is red, **standing** is yellow, **walking** is green, **lounging-on-the-couch** is blue, and **relaxing-on-the-chair** is purple.

Table 7.8 is metric 1 for activity. That is, each system generated activity summary is associated with the human activity interval that it overlaps with the most. Again, metric 1 is not a confusion matrix.

Table 7.8. Comparison between activity summaries for the linguistic summarization system, for summary confidences τ_2 and τ_5, the HMM system, and the human ground truth for all data sets.

Linguistic Summarization and HMM Systems

Each cell contains

$\pi_{A_i^l, Sum'_{l,g}} > \tau_2$
$\pi_{A_i^l, Sum'_{l,g}} > \tau_5$
HMM

Linguistic Summarization and HMM Systems	Ground Truth: **fall** (31)	**standing** (16)	**walking** (113)	**lounging-on-the-couch** (2)	**relaxing-on-the-chair** (8)	unknown (81)
fall	33 33 29	0 0 0	0 0 10	0 0 1	0 0 0	4 1 54
standing	0 0 1	15 15 15	9 6 70	0 0 0	0 0 0	0 0 25
walking	0 0 1	5 5 5	105 102 145	0 0 2	0 0 1	4 3 23
lounging-on-the-couch	0 0 0	0 0 0	0 0 0	2 2 2	0 0 0	1 1 1
relaxing-on-the-chair	0 0 0	0 0 0	0 0 1	0 0 0	7 7 7	0 0 0
unknown	4 4 NA	0 0 NA	2 5 NA	1 1 NA	8 8 NA	70 68 NA

Table 7.8 shows that the proposed linguistic summarization system does exceptionally well for nearly every activity except for a few **standing** false alarms. Of particular interest is the number of unknowns, 81. Unknown activity is 5,135 frames, over 17 minutes. Overall, the activities identified for this work do not suffer from the phenomenon of partitioning a significant portion of the description space, such as the partitioning of the height domain for the states **upright**, **in-between**, and **on-the-ground**. The system also does exceptionally well in classifying unknown activity, while the HMM is not applicable (NA).

The activities **standing** and **walking** are the most difficult to distinguish between. It was incorrectly assumed here that these two activities would be trivial to infer at the desired definition resolution, versus walking with a limp or walking briskly, and for an elderly population. The approach taken to recognize these activities is that when someone is walking they are **upright** and they move a *sufficient* distance over some *acceptable* time period. Otherwise, if they are standing, then they are assumed to be **upright** and have not traveled such a distance *recently*. It was thought that this would be a satisfactory general definition for these activities and that there is natural fuzziness in their definition. Results show that these activity definitions really only work in the most extreme cases. The real problem resides in the large fuzzy area in between these activities. Part of the problem is that these two activities exhibit the definition partitioning phenomenon. Another problem is believed to reside in the fact that the human is able to observe the entire activity sequence, has access to a larger set of rich activity definitions, and he/she uses this to determine the activity and its boundaries. It is asserted that in this fuzziest of regions, some different and more intelligent method is used by the human to reach a crisp decision. For example, fuzziness occurs when a person is walking and pauses. Should the pause be classified as **standing** or was it too brief in time and should just be classified as **walking**? In the future, voxel person features, such as the footfall features extracted by Stone, Anderson, et al [16], will be included and rule-definitions based on human gait will be explored.

On a final metric 1 note, the HMM-based system recognizes the majority of activity but has a significant number of false alarms. Bottom line, it cannot reject unknown activity. In the case of unknown activity, the system has to classify the activity into one of the existing models. In addition, the lower performance for **standing** is also observed in this approach.

Table 7.9, metric 2, is the similarity between frame-by-frame activity definitions inferred by the linguistic summarization and HMM systems in relation to the human ground truth. As in table 7.3, a system decision normalization of $M(i,j)/\sum_{j=1}^{6} M(i,j)$ is used.

Table 7.9. Frame-by-frame activity decision comparison between the linguistic summarization system, HMM-based system, and the human ground truth for all data sets and $M(i,j)/\sum_{j=1}^{6} M(i,j)$.

Linguistic Summarization and HMM Systems		Ground Truth: fall	standing	walking	lounging-on-the-couch	relaxing-on-the-chair	unknown
	fall	0.95	0.00	0.00	0.00	0.00	0.05
		0.98	0.00	0.00	0.00	0.00	0.02
		0.64	0.00	0.05	0.00	0.00	0.31
	standing	0.00	0.64	0.28	0.00	0.00	0.08
		0.00	0.66	0.25	0.00	0.00	0.09
		0.00	0.28	0.46	0.00	0.00	0.26
	walking	0.00	0.03	0.93	0.00	0.00	0.03
		0.00	0.02	0.94	0.00	0.00	0.03
		0.00	0.01	0.89	0.00	0.00	0.09
	lounging-on-the-couch	0.00	0.00	0.00	0.80	0.00	0.20
		0.00	0.00	0.00	0.81	0.00	0.19
		0.00	0.00	0.00	0.91	0.00	0.02
		0.00	0.00	0.03	0.58	0.00	0.39
	relaxing-on-the-chair	0.00	0.00	0.01	0.00	0.99	0.00
		0.00	0.00	0.01	0.00	0.99	0.00
		0.00	0.00	0.20	0.00	0.79	0.01
	unknown	0.03	0.00	0.02	0.00	0.05	0.89
		0.02	0.02	0.08	0.00	0.05	0.82
		NA	NA	NA	NA	NA	NA

Each cell contains

$\pi_{A_i^l Sum_{ig}^l} > \tau_2$
$\pi_{A_i^l Sum_{ig}^l} > \tau_5$
HMM

Metric 2 truly illustrates the superiority of the linguistic summarization system to the HMM-based approach. In particular, notice the excellent recognition rates for all activity outside of **standing**. The number of false alarms resulting from the HMMs is large, 31% for **fall**, 26% for **standing**, and 39% for **lying-on-the-couch**. Additionally, when mistakes are made by the linguistic summarization system for **standing**, the error is mainly due to confusion with **walking**, not unknown. However, the HMM system has significant error in both distinguishing between **walking** and unknown. On a final note, 89% of unknown activity can be correctly recognized using the linguistic summarization system. In summary, approximately 127 out of 136 minutes of activity is correctly classified. Figure 7.3 shows a significant category of frame-by-frame activity error encountered.

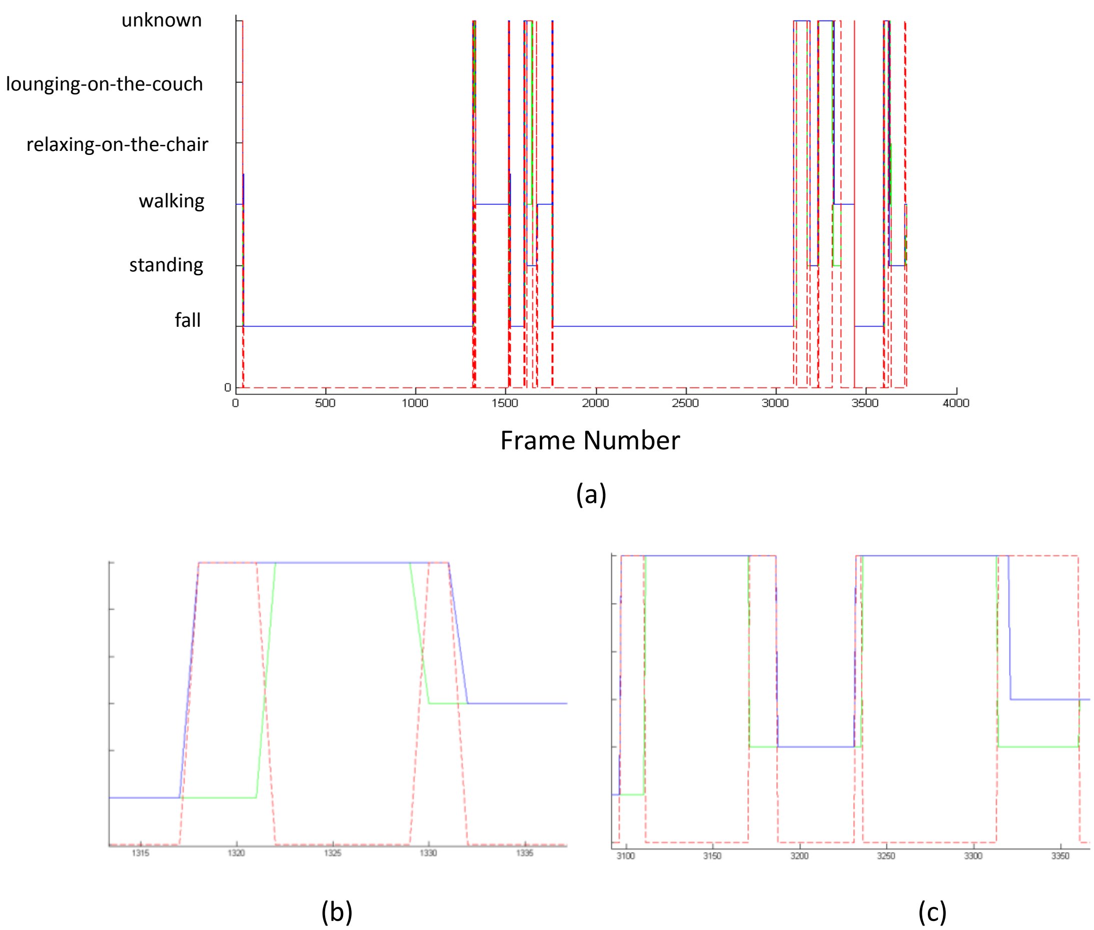


Fig. 7.3. Illustration of a significant type of frame-by-frame activity error encountered. At each frame, in (a), (b), and (c), the activity with the maximum membership grade is selected. Based on this index, a plot is created for the linguistic summarization system (green) versus the human's labels (blue). Moments of disagreement, which typically occur at summary endpoints, are indicated by red dashed lines. Images (b) and (c) are smaller time intervals from (a). They help simplify the understanding of this type of error.

Figure 7.3 shows that a good percentage of encountered frame-by-frame error is due to summary endpoint disagreement between the system and human. To no surprise, this behaviour is also exhibited by the HMM system. This is not shocking because this is a significant grey or fuzzy area of the work, transition between activities.

Table 7.10, metric 3, is the amount of information reduction according to the number of summaries produced per activity divided by the respective total number of origional frames per activity. Again, a value of one is bad while a lower value, e.g. 0.01 (1%), is good. Table 7.10 shows that summarization once again results in drastic information reduction.

Table 7.10. Percentage of frame-by-frame decisions remaining after linguistic summarization fo activity for all data sets. The lower the value the better.

	fall	**standing**	**walking**	**lounging-on-the-couch**	**relaxing-on-the-chair**	unknown
$\pi_{A_i^l, Sum'_{l,g}} > \tau_2$	0.005	0.040	0.023	0.001	0.035	0.025
$\pi_{A_i^l, Sum'_{l,g}} > \tau_5$	0.004	0.035	0.022	0.001	0.035	0.026
HMMs	0.012	0.123	0.020	0.008	0.013	NA
Human	0.004	0.024	0.023	0.000	0.031	0.025

Next, metric 4 indicates if the system is capable of producing a reduced acceptable number of correctly time aligned summaries. Again, the criteria for maximum number of allowable correct system generated summaries and the required amount of human ground truth interval overlap are varied. Table 7.11 is the matrix cell format for tables 7.12 and 7.13.

Table 7.11. Maxtirx cell format for tables 7.12 and 7.13.

fall for $\pi_{A_i^l, Sum_{i,g}^{\prime}} > \tau$	**fall** for HMM
standing for $\pi_{A_i^l, Sum_{i,g}^{\prime}} > \tau$	**standing** for HMM
walking for $\pi_{A_i^l, Sum_{i,g}^{\prime}} > \tau$	**walking** for HMM
lounging-on-the-couch for $\pi_{A_i^l, Sum_{i,g}^{\prime}} > \tau$	**lounging-on-the-couch** for HMM
resting-on-the-chair for $\pi_{A_i^l, Sum_{i,g}^{\prime}} > \tau$	**resting-on-the-chair** for HMM
unknown for $\pi_{A_i^l, Sum_{i,g}^{\prime}} > \tau$	unknown for HMM

Table 7.12. Variation of metric 4 matching success criteria for the linguistic summarization system, for $\pi_{A_i^l,Sum'_{l,g}} > \tau_2$, and the human ground truth for all data sets. This metric should be analyzed in conjunction with 7.8 and 7.9, which highlight false alarms.

Maximum Number of Correct System Generated Summaries	Required Overlap (Percentage) with the Human Interval: 0.1		0.25		0.5		0.75		0.9	
1	0.97	0.94	0.97	0.94	0.97	0.94	0.97	0.94	0.55	0.94
	0.94	0.88	0.94	0.88	0.5	0.81	0.31	0.75	0.13	0.50
	0.82	0.60	0.82	0.59	0.82	0.53	0.77	0.41	0.69	0.27
	1	1	1	1	1	1	1	1	0.5	0.5
	0.63	0.88	0.63	0.88	0.5	0.88	0.25	0.88	0	0.88
	0.80	NA	0.79	NA	0.75	NA	0.62	NA	0.57	NA
2	0.97	0.94	0.97	0.94	0.98	0.94	0.97	0.94	0.55	0.94
	0.94	0.88	0.94	0.88	0.5	0.81	0.31	0.75	0.13	0.50
	0.88	0.79	0.88	0.79	0.88	0.71	0.82	0.51	0.73	0.32
	1	1	1	1	1	1	1	1	0.5	0.5
	0.75	0.88	0.75	0.88	0.63	0.88	0.38	0.88	0	0.88
	0.83	NA	0.81	NA	0.77	NA	0.62	NA	0.57	NA
5	1	0.94	0.97	0.94	0.97	0.94	0.97	0.94	0.55	0.94
	0.94	0.88	0.94	0.88	0.5	0.81	0.31	0.75	0.13	0.50
	0.88	0.88	0.88	0.88	0.88	0.78	0.82	0.56	0.73	0.34
	1	1	1	1	1	1	1	1	0.5	0.5
	0.75	0.88	0.75	0.88	0.63	0.88	0.38	0.88	0	0.88
	0.83	NA	0.81	NA	0.77	NA	0.62	NA	0.57	NA
10	1	0.94	0.97	0.94	0.97	0.94	0.97	0.94	0.55	0.94
	0.94	0.88	0.94	0.88	0.5	0.81	0.31	0.75	0.13	0.50
	0.88	0.88	0.88	0.88	0.88	0.78	0.82	0.56	0.73	0.34
	1	1	1	1	1	1	1	1	0.5	0.5
	0.75	0.88	0.75	0.88	0.63	0.88	0.38	0.88	0	0.88
	0.83	NA	0.81	NA	0.77	NA	0.62	NA	0.57	NA

Table 7.13. Variation of the metric 4 matching success criteria for the linguistic summarization system, for $\pi_{A_i^l,Sum_{l,g}'} > \tau_5$, and the human ground truth for all data sets. This metric should be analyzed in conjunction with 7.8 and 7.9, which highlight false alarms.

		Required Overlap (Percentage) with the Human Interval									
		0.1		0.25		0.5		0.75		0.9	
Maximum Number of Correct System Generated Summaries	1	0.97	0.94	0.97	0.94	0.97	0.94	0.97	0.94	0.55	0.94
		0.75	0.88	0.75	0.88	0.38	0.81	0.25	0.75	0.13	0.50
		0.74	0.60	0.74	0.59	0.74	0.53	0.69	0.41	0.61	0.27
		1	1	1	1	1	1	1	1	0.5	0.5
		0.63	0.88	0.63	0.88	0.5	0.88	0.25	0.88	0	0.88
		0.77	NA	0.75	NA	0.72	NA	0.59	NA	0.56	NA
	2	0.97	0.94	0.97	0.94	0.98	0.94	0.97	0.94	0.55	0.94
		0.75	0.88	0.75	0.88	0.38	0.81	0.25	0.75	0.13	0.50
		0.79	0.79	0.79	0.79	0.79	0.71	0.73	0.51	0.65	0.32
		1	1	1	1	1	1	1	1	0.5	0.5
		0.75	0.88	0.75	0.88	0.63	0.88	0.38	0.88	0	0.88
		0.77	NA	0.75	NA	0.72	NA	0.59	NA	0.56	NA
	5	1	0.94	0.97	0.94	0.97	0.94	0.97	0.94	0.55	0.94
		0.75	0.88	0.75	0.88	0.38	0.81	0.25	0.75	0.13	0.50
		0.79	0.88	0.79	0.88	0.79	0.78	0.73	0.56	0.65	0.34
		1	1	1	1	1	1	1	1	0.5	0.5
		0.75	0.88	0.75	0.88	0.63	0.88	0.38	0.88	0	0.88
		0.77	NA	0.75	NA	0.72	NA	0.59	NA	0.56	NA
	10	1	0.94	0.97	0.94	0.97	0.94	0.97	0.94	0.55	0.94
		0.75	0.88	0.75	0.88	0.38	0.81	0.25	0.75	0.13	0.50
		0.79	0.88	0.79	0.88	0.79	0.78	0.73	0.56	0.65	0.34
		1	1	1	1	1	1	1	1	0.5	0.5
		0.75	0.88	0.75	0.88	0.63	0.88	0.38	0.88	0	0.88
		0.77	NA	0.75	NA	0.72	NA	0.59	NA	0.56	NA

Tables 7.12 and 7.13 tell the following story. For a weak criteria, such as up to ten allowable number of correct system generated summaries and 10% required overlap with the human interval, both systems perform exceptionally well. Again, cells highlighted in red are considered to be realistic criteria for evaluating overall system success. This criteria indicates that **fall** is recognized at an acceptionally high rate, while the majority of other activities do not hold up as well. To save space and eliminate redundancy, these metrics are not shown for only the stunt actor false alarm data set, **SET2.d**. These tables reinforce what is already known. Activities, except for **standing**, have an exceptionally higher

performance outside of **SET2.d**. Thus, the system does significantlly better when activity defnitions are applied in the context they are designed for. On a final note, this metric makes the HMMs appear to perform exceptionally well. However, this metric only asks the question if a limited number of correct summaries that make up at least some percentage of the human ground truth intervals can be found. What this metric does not highlight is false alarms. There is little point in achieving a high metric 4 score if the approach has a significant number of false alarms, as shown by tables 7.8 and 7.9 for the HMM.

The following is computed only for the goal of **fall** recognition. This table is quite possibly the most important result in the context of this dissertation.

Table 7.14. Confusion matrix for the activity **fall** according to the linguistic summarization system and the HMM-based system with respect to the human ground truth for all data sets.

		Ground Truth	
		fall	non-**fall**
Linguistic Summarization and HMM Systems Each cell contains $\pi_{A_i^l, Sum'_{l,g}} > \tau_2$ $\pi_{A_i^l, Sum'_{l,g}} > \tau_5$ HMM	**Fall**	$\frac{31}{31} = 100\%$ $\frac{31}{31} = 100\%$ $\frac{29}{31} = 94\%$	$\frac{4}{220} = 2\%$ $\frac{1}{220} = 0\%$ $\frac{51}{220} = 23\%$
	non-**fall**	$\frac{0}{31} = 0\%$ $\frac{0}{31} = 0\%$ $\frac{2}{31} = 6\%$	98% 100% 77%

Table 7.14 shows that the linguistic summarization system achieves excellent true-positive score, while the HMM-based approach does not detect two falls. This type of error is the worst scenario. In this situation, an elder falls and does not receive assitance. In addition, four human labeled non-**fall** intervals are incorrectly called a **fall** by the linguistic summarization system with respect to τ_2 (one mistake for τ_5). This type of error is not as severe as missing a **fall**. A caregiver is dispatched but the elder is okay. All four mistakes come from the stunt actor false alarm data set, **SET2.d**. The false alarms are due to activity unknown to the system while the human is on the ground, e.g. exercising at a very slow pace. On a final note, an exceptionally high number, 51, of human labeled non-**fall** intervals are inccorectly called a **fall** by the HMM approach. This further stresses the defeciency associated with HMMs for activity analysis. When similar unknown activity is performed, it has to be labeled something. The linguistic summarization system addresses this by explicitly addressing generation meaningful confidencs and detecting unknown activity not compaiable with definitions of tracked activity.

7.4 Summary

The following has been demonstrated in the results chapter. Linguistic summarization is successful in that it leads to significant information reduction for both state and activity. The rule-definitions for state perform exceptionally well in the context of fall recognition. However, state unknown to the system results in lower observed performance in **on-the-ground** and poor performance for the ill-defined state **in-between**. Additionally, a reason for such little observed unknown state is due to the partitioning of the humans pose in the height domain for **on-the-ground**, **in-between**, and **upright**.

Examination into activity reveals the following. First and foremost, the proposed linguistic summarization system outperforms the HMM-based system for **fall** recognition. The explicit addressing of inference and segmentation is key to overcoming the mentioned HMM most likely model pitfalls. The majority of activities are inferred with high accuracy. The primary problem encountered is the simplicity and definition space partitioning behaviour of **standing** and **walking**. Also, a larger number of unknown activities are observed compared to unknown state. The significant amount of unknown activity is correctly classified, thus demonstrating the power of the proposed linguistic summarization system.

Hence, the results show significant progress is made in addressing two critical activity analysis topics. The first topic is the ability of a system to explicitly summarize a time series and yield linguistic summarizations that humans can understand. The second advance is in the design of a system that produces meaningful confidences that assist in the rejection of a larger range of unknown activity.

Conclusions

8.1 Summary

The work presented in this dissertation comprises several contributions to computer vision, namely human activity analysis. The most novel contributions are activity inference and temporal linguistic summarization from voxel person. Secondary contributions include human silhouette extraction and creation of voxel person. The proposed soft-computing framework results in significant information reduction in a linguistic format that humans can understand. In addition, meaningful activity confidence values are inferred. The resulting confidences are shown to lead to impressive rejection of unknown activity. The proposed linguistic summarization system is compared to the current state of the art, HMMs, and a human generated ground truth. Results show that the linguistic summarization approach achieves outstanding activity recognition rates and it is superior to an HMM-based approach. Specifically, this is with respect to the main activity sought after in this work, fall detection, which has been validated on a stunt actor trained to fall like the elderly.

8.2 Future Work

While the overall work is a great success, future works will look to enhance the following areas. As already discussed in great length, human silhouette segmentation in image space is an extremely difficult problem that is not yet solved. The stereo vision and voxel space-based approach of Luke, Anderson, et al. [14] is asserted to be a big step forward in the robust segmentation of humans in dynamic and complex indoor environments. The integration of this stereo vision-based system into the linguistic summarization work presented in this dissertation will be explored.

Another important area of future work is moving beyond using domain experts to assist in the design and specification of many system parameters. This approach is useful for a specific domain, eldercare, and is valuable in that it was demonstrated that such a framework is capable of achieving very good activity recognition. Future work will need to address the learning of fuzzy sets and rules.

Currently, terms like voxel person's *height* are not personalized to each individual. While the proposed terms can be defined for each resident, scalability of this general approach has not been explored. This is especially true if one desires to extend the work to the tracking of multiple people. Even if tracking and correspondence works properly, there is still the possibility that unknown individuals enter the monitored space. A better approach is to start with a general definition of these terms and refine them based on observations.

On a final note, numeric features are currently extracted from linguistic summarizations. This information is used to infer higher levels of more complex activity using fuzzy logic. Instead of extracting numeric features from summaries, future work will include exploring procedures for directly computing with summarizations. This work is motivated by the computing with word (CWW) research of Zadeh [123][124][3] and Mendel [125][126].

Bibliography

[1] D. T. Anderson, R. H. Luke, J. M. Keller, M. Skubic, M. Rantz, M. Aud, "Modeling Human Activity from Voxel Person Using Fuzzy Logic," in *IEEE Transactions on Fuzzy Systems*, vol 17, issue 1, pp. 39-49, 2009.

[2] D. T. Anderson, R. H. Luke, J. M. Keller, M. Skubic, M. Rantz, M. Aud, "Linguistic Summarization of Video for Fall Detection Using Voxel Person and Fuzzy Logic," in *Computer Vision and Image Understanding*, vol 113, issue 1, pp. 80-89, 2009.

[3] L. Zadeh, "Generalized Theory of Uncertainty (GTU) - Principal Concepts and Ideas," in *Computational Statistics & Data Analysis*, vol 51, issue 1, pp. 15-46, 2006.

[4] D. Gibbins, G. N. Newsam, M.J. Brooks, "Detecting Suspicious Background Changes in Video Surveillance of Busy Scenes," in *3rd IEEE Workshop on Applications of Computer Vision*, pp. 22-26, 1996.

[5] D. Thirde, M. Borg, J. Ferryman, F. Fusier, V. Valentin, F. Bremond, M. Thonnat, "A Real-Time Scene Understanding System for Airport Apron Monitoring," in *Proceedings of the Fourth IEEE International Conference on Computer Vision Systems*, p. 26, 2006.

[6] G.L. Foresti, L. Marcenaro, C.S. Regazzoni, "Automatic Detection and Indexing of Video-Event Shots for Surveillance Applications," in *IEEE Transactions on Multimedia*, pp. 459-471, 2002.

[7] R.T. Collins, A.J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, L. Wixson, "A System for Video Surveillance and Monitoring," in *American Nuclear Society 8th Internal Topical Meeting on Robotics and Remote Systems*, pp. 734–741, 1999.

[8] B. Yao, H. Ai, S. Lao, "Person-Specific Face Recognition in Unconstrained Environments: a Combination of Offline and Online Learning," in *8th IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 1-8, 2008.

[9] Y.M. Mustafah, A.W. Azman, A. Bigdeli, B.C. Lovell, "An Automated Face Recognition System for Intelligence Surveillance: Smart Camera Recognizing Faces in the Crowd," in *First ACM/IEEE International Conference on Distributed Smart Cameras*, pp. 147-152, 2007.

[10] B. Banks, G. Jackson, J. Helly, D. Chin, T.J. Smith, A. Schmidt, P. Brewer, R. Medd, D. Masters, A. Burger, W.K. Krebs, "Using behavior analysis algorithms to anticipate security threats before they impact mission critical operations," in *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, pp. 307-312, 2007.

[11] D. Duque, H. Santos, P. Cortez, "Prediction of Abnormal Behaviors for Intelligent Video Surveillance Systems," in *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining*, pp. 362-367, 2007.

[12] N. Kiryati, T. Riklin-Raviv, Y. Ivanchenko, S. Rochel, "Real-time Abnormal Motion Detection in Surveillance Video," in *19th International Conference on Pattern Recognition*, pp. 1-4, 2008.

[13] T. Martin, B. Majeed, L. Beum-Seuk, N. Clarke, "Fuzzy Ambient Intelligence for Next Generation Telecare," in *IEEE International Conference on Fuzzy Systems*, pp. 894 – 901, 2006.

[14] R. H. Luke, D. T. Anderson, J. M. Keller, "Human Change Detection in Voxel Space using Stereo Vision," submitted to *Computer Vision and Image Understanding*, 2010.

[15] D. T. Anderson, R. H. Luke, J. M. Keller, "Linguistic Summarization of Scenes in a Stereo-Vision Acquired Voxel Space," accepted for publication in *IEEE International Conference on Fuzzy Systems at the World Congress on Computational Intelligence*, 2010.

[16] E. Stone, D. T. Anderson, M. Skubic, J. M. Keller, "Footfall extraction and visualization from voxel data," accepted for publication in *Gerontechnology* , 2010.

[17] D. T. Anderson, R. H. Luke, M. Skubic, J. M. Keller, M. Rantz, "Evaluation of a Video Based Fall Recognition System for Elders Using Voxel Space," in *6th Conference of the International Society for Gerontechnology*, pp. 77-82, 2008.

[18] D. T. Anderson, R. H. Luke, J. M. Keller, M. Skubic, "Extension of a Soft-Computing Framework for Activity Analysis from Linguistic Summarizations of Video," in *IEEE International Conference on Fuzzy Systems at the World Congress on Computational Intelligence,* pp. 1404-1410, 2008.

[19] D. T. Anderson, J. Keller, M. Skubic, X. Chen, Z. He, "Recognizing Falls from Silhouettes," in *IEEE 2006 International Conference of the Engineering in Medicine and Biology Society*, pp. 6388-6391, 2006.

[20] N. Thome and S. Miguet, "A HHMM-based approach for robust fall detection," in *9th International Conference on Control, Automation, Robotics and Vision*, pp. 1-8, 2006.

[21] Sebastian Lühr, Hung Bui, Svetha Venkatesh, Geoff West, "Recognition of Human Activity through Hierarchical Stochastic Learning," in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, pp. 416, 2003.

[22] S. Fine, Y. Singer, N. Tishby, "The hierarchical hidden markov model: Analysis and applications," in *Machine Learning*, pp. 41-62, 1998.

[23] N.M. Oliver, B. Rosario, A.P. Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions," in *IEEE Transactions on Pattern Recognition*, vol 22, issue 8, pp. 831-843, 2000.

[24] C. Stauffer and W. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," in *Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 246-252, 1999.

[25] W.L. Buntine, "Operations for Learning with Graphical Models," in *Journal of Artificial Intelligence Research*, vol 2, issue 1, pp. 159-225, 1994.

[26] M. Brand and V. Kettnaker, "Discovery and Segmentation of Activities in Video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 844-851, 2000.

[27] R. H. Luke, D. T. Anderson, J. Keller, M. Skubic, "Human segmentation from video in indoor environments using fused color and texture features," Technical Report – University of Missouri, http://cirl.missouri.edu/vision/papers/TextureFusionHumanSegmentation.pdf, 2008.

[28] D. T. Anderson, R. H. Luke, E. Stone, J. M. Keller, "Fuzzy Voxel Object," in *International Fuzzy Systems Association*, pp. 282-287, 2009.

[29] W Heisenberg, "Ueber die Grundprincipien der "Quantenmechanik" ," *Forschungen und Fortschritte* , 1927.

[30] D. Marr, *Vision*. W. H. Freeman and Company, 1982.

[31] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum, NY, 1981.

[32] M. Brand, N. Oliver, A. Pentland, "Coupled Hidden Markov Models for Complex Action Recognition," in *Proceedings IEEE Computer Vision and Pattern Recognition*, pp. 994-999, 1997.

[33] L. Zadeh, "Fuzzy sets," in *Information Control*, pp. 338-353, 1965.

[34] L. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," in *IEEE Transactions on System, Man, and Cybernetics*, pp. 28-44, 1973.

[35] L. Zadeh, "The Concept of a Linguistic Variable and its Applications to Approximate Reasoning I," in *Information Sciences*, pp. 199-249, 1975.

[36] L. Zadeh, "The Concept of a Linguistic Variable and its Applications to Approximate Reasoning II," in *Information Sciences*, pp. 301-357 , 1975.

[37] L. Zadeh, "The Concept of a Linguistic Variable and its Applications to Approximate Reasoning III," in *Information Sciences*, pp. 43-80, 1975.

[38] B. Yuan and G. Klir, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, 1995.

[39] L. Zadeh, "Possibility Theory and Soft Data Analysis," in *Mathematical Frontiers of Social and Policy Sciences*, pp. 69-129, 1981.

[40] J. Bezdek, "Editorial: Fuzzy Models – What are they, and why?," *IEEE Transactions on Fuzzy Systems*, pp. 1-6, 1993.

[41] J. Bezdek, "Editorial – Fuzziness vs. Probability – Again (! ?)," *IEEE Transactions on Fuzzy Systems*, pp. 1-3, 1991.

[42] L. Zadeh, "Fuzzy sets as a basis for a theory of possibility," in *Fuzzy Sets and Systems*, pp. 3-28, 1978.

[43] L. Zadeh, "A fuzzy-set-theoretic interpretation of linguistic hedges," in *Cybernetics*, pp. 4–34, 1972.

[44] E. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," in *International Journal of Man-Machine Studies*, vol 7, issue 1, pp. 1-13, 1975.

[45] T. Martin, B. Majeed, L. Beum-Seuk, N. Clarke, "Fuzzy Ambient Intelligence for Next Generation Telecare," in *IEEE International Conference on Fuzzy Systems*, pp. 894 – 901, 2006.

[46] J. Xiong, B. Seet, J. Symonds, "Human Activity Inference for Ubiquitous RFID-Based Applications," in *Sixth International Conference on Ubiquitous Intelligence and Computing*, pp.304-309, 2009.

[47] S. Morris and J. Paradiso, "A Compact Wearable Sensor Package for Clinical Gait," in *Offspring* , vol 1, issue 1, pp. 7-15, 2003.

[48] R. Nevatia and T. Zhao, "Tracking Multiple Humans in Complex Situations," in *IEEE Transactions on Pattern Recognition and Machine Intelligence* , vol 26, issue 9, pp. 1208-1221, 2004.

[49] T. Zhao, R. Nevatia, B. Wu, "Segmentation and Tracking of Multiple Humans in Crowded Environments," in *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol 30, issue 7, pp. 1198-1211, 2008.

[50] X. Wang, T. X. Han, S. Yan, "An HOG-LBP Human Detector with Partial Occlusion Handling," in *IEEE International Conference on Computer Vision*, 2009.

[51] K. Toyama, J. Krumm, B. Brumitt, B. Meyers, "Wallflower: Principles and Practice of Background Maintenance," in *International Conference on Computer Vision*, pp. 255-261, 1999.

[52] H. Kawanaka, H. Fujiyoshi, Y. Iwahori, "Human Head Tracking in Three Dimensional Voxel Space," in *Proceedings of the 18th International Conference on Pattern Recognition* , pp. 826 – 829, 2006.

[53] R. Muñoz-Salinas, E. Aguirre, M. García-Silvente, "People detection and tracking using stereo vision and color," in *Image and Vision Computing*, vol 25, issue 6, pp. 995-1007, 2007.

[54] P Power and J Schoonees, "Understanding Background Mixture Models for Foreground Segmentation," in *Proceedings Image and Vision Computing New Zealand*, pp. 267-271, 2002.

[55] M Turk and A Pentland, "Eigenfaces for Recognition," in *Journal of Cognitive Neuroscience*, vol. 3, issue 1, pp. 77-86, 1991.

[56] T. Han, "Computing Principal Components with the Eigen-decomposition of a Low Dimenisonal Matrix," Technical Report, University of Missouri, 2010.

[57] L Wang, M. Wen, Q. Zhuo, W. Wang, "Background Subtraction using Incremental Subspace Learning," in *International Conference on Image Processing*, pp. 45-48, 2007.

[58] J. Sun, W. Zhang, X. Tang, H. Shum, "Background Cut," in *European Conference on Computer Vision*, pp. 628-641, 2006.

[59] Y. Boykov and V. Kolmogorov, "An Experimental Comparsion of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 26, issue 9, pp. 1124-1137, 2004.

[60] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: theory andits application to image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, issue 11, pp. 1101-1113, 1993.

[61] S. Roy and I. Cox, "A maximum-flow formulation of the n-camera stereo correspondence problem," in *Proceedings of the International Conference on Computer Vision*, pp. 492-299, 1998.

[62] J. Shi and J. Malik, "Normalized cuts and image segmentation," in *IEEE Transactions on Pattern*

Analysis and Machine Intelligence, vol 22, issue 8, pp. 888-905, 2000.

[63] C. Huang and C. Cheng, "Color images segmentation using scale space filter and Markov random field," *Pattern Recognition* , vol 25, issue 10, pp. 1217-1229, 1992.

[64] D. Greig, B. Porteous, A. Seheult, "Exact maximum a posteriori estimation for binary images," in *Journal of the Royal Statistics Society*, vol 51, issue 1, pp. 271-279, 1989.

[65] J. Sun, W. Zhang, X. Tang, H. Shum, "Background Cut," in *European Conference on Computer Vision* , pp. 628-641, 2006.

[66] H. Kim, D. Min, S. Choi, K. Sohn, "Real-Time Disparity Estimation using Foreground Segmentation for Stereo Sequences," in *Optical Engineering*, vol 45, issue 3, pp. 037402-1 - 037402-10, 2006.

[67] R. Muñoz-Salinas, E. Aguirre, M. García-Silvente, "People detection and tracking using stereo vision and color," *Image and Vision Computing*, vol 25, issue 6, pp. 995-1007, 2007.

[68] M. Brown, D. Burschka, G. Hager, "Advances in Computational Stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 25, issue 8, pp. 993-1008, 2003.

[69] R. H. Luke, "A System for Change Detection and Human Recognition in Voxel Space Using Stereo Vision," PhD Thesis, University of Missouri, 2010.

[70] D. T. Anderson, R. H. Luke, J. Keller, "Segmentation and Linguistic Summarization of Voxel Environments using Stereo Vision and Genetic Algorithms," in *IEEE International Conference on Fuzzy Systems at the World Congress on Computational Intelligence*, 2010.

[71] Point Grey. www.ptgrey.com/, 2010

[72] H. Kim, D. Min, S. Choi, K. Sohn, "Real-Time Disparity Estimation using Foreground Segmentation for Stereo Sequences," in *Optical Engineering*, vol 45, issue 3, pp. 037402-1 – 037402-10, 2006.

[73] R. Muñoz-Salinas, E. Aguirre, M. García-Silvente, "People detection and tracking using stereo vision and color," *Image and Vision Computing*, vol 25, issue 6, pp. 995-1007, 2007.

[74] K. Murphy, "Dynamic Bayesian Networks: Representation, Inference and Learning," PhD Thesis, UC Berkeley, 2002.

[75] George Casella, *Statistical Inference*, Second Edition, Duxbury Press, 2001.

[76] R Diestel, *Graph Theory*, Third Edition, Springer-Verlag, 2005.

[77] W. Zajdel, A. Taylancemgil, B. Krose, "Dynamic bayesian networks for visual surveillance with distributed cameras," in *European Conference on Smart Sensing & Context*, 2006.

[78] L. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," in *Proceedings of the IEEE*, vol 77, issue 2, pp. 257-286, 1989.

[79] W. Freeman, E. Pasztor, O. Carmichael, "Learning Low-Level Vision," in *International Journal of Computer Vision*, vol 40, issue 1, pp. 25-47, 2000.

[80] L. Wilcox and M. Bush, "Training and Serach Algorithms for an Interactive Wordspotting System," in *International Conference on Acoustics, Speech, and Signal Processing*, pp. 12-49, 1991.

[81] D. T. Anderson, C. Bailey, M. Skubic, "Hidden Markov Model Symbol Recognition for Sketch Based Interfaces," in *AAAI Fall Workshop on Making Pen-Based Interaction Intelligent and Natural*, pp. 15-21, 2004.

[82] M Mohamed and P Gader, "Generalized hidden Markov models. I. Theoretical frameworks," in *IEEE Transactions on Fuzzy Systems*, vol 8, issue 1, pp. 67-81, 2000.

[83] M Mohamed and P Gader, "Generalized hidden Markov models. II. Application to handwritten word recognition," in *IEEE Transactions on Fuzzy Systems*, vol 8, issue 1, pp. 82-94, 2000.

[84] S. Miaou, P. Sung, C. Huang, "A customized human fall detection system using omni-camera images and personal information," in *1st Distributed Diagnosis and Home Healthcare*, pp. 39-42, 2006.

[85] M. Ryoo and J. Aggarwarl, "Recognition of Composite Human Activities through Context-Free Grammar Based Representation," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1709-1718, 2006.

[86] S. Park and K. Aggarwarl, "Simultaneous tracking of multiple body parts of interacting persons," in *Computer Vision and Image Understanding*, vol 102, issue 1, pp. 1-21, 2006.

[87] D. Moore and I. Essa, "Recognizing Multitasked Activities using Stochastic Context-Free Grammars," in *Proceedings of National Conference on Artificial Intelligence*, pp. 770-776, 2002.

[88] K. Lari and S. Young, "The Estimation of Stochastic Context-Free Grammars Uisng the Inside-Outside Algorithm," in *Computer, Speech, and Language*, vol. 24, issue 2, pp. 35-56, 1990.

[89] A. Stolcke, "Bayesian Learning of Probabilistic Langauge models," PhD Thesis, UC at Berkeley, 1994.

[90] Y. Ivanov and A. Bobick, "Recognition of visual activities and interactions by sotchastic parsing," in *IEEE Pattern Analysis and Machine Learning*, vol 22, issue 8, pp. 852-872, 2000.

[91] B. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," in *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 3, no. 1, p. 3, 2007.

[92] A. Hanjalic and H. Zhang, "An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis," in *IEEE Transactionson Circuts Systems Video Technology*, vol. 9, no. 8, pp. 1280-1289, 1999.

[93] J. Wu, M. Kankanhalli, J. Lim, D. Hong, *Perspective on Content-Based Multimedia Systems*. Kluwer Academic, 2000.

[94] S. Lee and M. Haye, "An application for interactive video abstraction," in *Proceedings of the ICASSP Conference*, pp. 905-908, 2004.

[95] D. Gibson and N. Thomas, "Visual abstraction of wildlife footage using Gaussian mixture models," in *Proceedings of the 15th International Conference on Vision Interface*, pp. 814-817, 2002.

[96] C. Ngo, "Video Summarization and Scene Detection by Graph Modeling," in *IEEE Transactions on Circuts and Systems for Video Technology*, vol 15, issue 2, pp. 104, 2005.

[97] E. Kang, S. Kim, J. Choi, "Video retrieval based on scene change detection in compressed domain," in *IEEE Transactions on Consumer Electronics*, vol 45, issue 3, pp. 932-936, 1999.

[98] M. Amadasun and R. King, "Textural features corresponding to textural properties," in *IEEE Transactions on Systems, Man and Cybernetics*, vol 19, issue 5, pp. 1264-1274, 1989.

[99] M. Flicker, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, D. Yanker, "Query by image and video content: the qbic system," in *IEEE Computer* , vol 28, issue 9, pp. 23-32, 1995.

[100] R. Haralick, "Statistical and Structural Approaches to Texture," *IEEE Proceedings* , vol 67, no. 5, pp. 786-804, 1979.

[101] Q. Iqbal and J. Aggarwal, "Cires: A system for content-based retrieval in digital image libraries," in *Proceedings of International Conference on Control, Automation, Robotics and Vision*, pp. 205-210, 2002.

[102] H. Tamura, S. Mori, T. Yamawaki, "Textural features corresponding to visual perception," in *IEEE Transactions on Systems, Man and Cybernetics*, vol 8, issue 6, pp. 460-473, 1978.

[103] S. Arivazhagan, L. Ganesan, V. Angayarkanni, "Color Texture Classification using Wavelet Transform," in *Sixth International Conference on Computational Intelligence and Multimedia Applications*, pp. 315-320, 2005.

[104] S. Belongie, C. Carson, H. Greenspan, J. Malik, "Color-and texture-based image segmentation using EM and its application to content-based image retrieval," in *Sixth International Conference on Comptuer Vision*, pp. 675-682, 1998.

[105] M. Haindl, J. Grim, P. Somol, P. Pudil, M. Kudo, "A Gaussian Mixture-Based Colour Texture Model," in *International Conference on Patter Recognition*, pp. 177-180, 2004.

[106] G. Paschos, "Perceptually uniform color spaces for color texture analysis: An experimental evaluation," in *IEEE Trnsactions on Image Processing*, vol. 10, issue 6, pp. 932-937, 2001.

[107] R. Gonzalez and R. Woods, *Digital Image Processing*, Third Edition, Prentice Hall, 2007.

[108] S. Edelman, N. Intrator, T. Poggio, "Complex Cells and Object Recognition," Unpublished Manuscript: http://kybele.psych.cornell.edu/~edelman/archive.html, Cornell, 1997.

[109] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," in *International Journal of Computer Vision*, vol 60, issue 2, pp. 91-110, 2004.

[110] R. Yager, "On a General class of fuzzy connectives," in *Fuzzy Sets and Systems*, vol. 4, issue 3, pp. 235-242, 1980.

[111] P. Blauensteiner, H. Wildenauer, A. Hanbury, M. Kampel, "Motion and Shadow Detection with an Improved Colour Model," in *International Conference on Signal and Image Processing*, pp. 627-632, 2006.

[112] B. Baumgart, "Geometric Modeling for Computer Vision," Technical Report AIM-249, Stanford, 1974.

[113] B. Vemri and J. Aggarwal, "3-D model construction from multiple views using range and intensity data," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 435-437, 1986.

[114] A. Laurentini, "The visual hull concept for silhouette-based image understanding," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 150-162, 1994.

[115] G. Dudek and D. Daum, "On 3-D surface reconstruction using shape from shadows," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 461-468, 1998.

[116] M. Pardas and J. Landabaso, "Foreground regions extraction and characterization towards real-time object tracking," in *Proceedings of Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, pp. 37-42, 2005.

[117] J Bouguet. (2010, May) Camera Calibration Toolbox for Matlab, http://www.vision.caltech.edu/bouguetj/calib_doc/, 2010.

[118] Z. Zhang, "Flexible Camera Calibration By Viewing a Plane From Unknown Orientations," in *International Conference on Computer Vision*, pp. 666-673, 1999.

[119] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, pp. 1106-1112, 1997.

[120] L. Zadeh, "Fuzzy Sets and Information Granularity," in *Advances in Fuzzy Set Theory and Applications*. North-Hollard, pp. 3-18, 1979.

[121] L. Zadeh, "Possibility theory and soft data analysis," in *Mathematical frontiers of the social and policy sciences*, eds. L. Cobb and R. M. Thrall, pp. 69-129, 1981.

[122] A. Blimes, "Gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," Technical Report, TR-97-021, 1998.

[123] L. Zadeh, "Fuzzy Logic = Computing with Words," in *IEEE Transactions on Fuzzy Systems*, vol 4, issue 2, pp. 103-111, 1996.

[124] L. Zadeh, *Outline of Computational Theory of Perceptions Based on Computing with Words*. Academic Press, 2000.

[125] J. Mendel and D. Wu, "Perceptual Reasoning for Perceptual Computing," in *IEEE Transactions on Fuzzy Systems*, vol 16, issue 6, pp. 1550-1564, 2008.

[126] J. Mendel, "An Architecture for Making Judgments Using Computing With Words," in *International Journal Applied Math Computer Science*, vol 12, issue 3, pp. 325-335, 2002.

VITA

Derek Anderson's expertise resides in the fields of pattern recognition, computational intelligence, computer vision, clustering, and information fusion. His primary research focus is linguistic summarization of human activity, with applications in eldercare. His research is internationally recognized by multiple journals, such as the IEEE Transactions on Fuzzy Systems, Elsevier's Computer Vision and Image Understanding, as well as many international conferences. Derek was awarded the best student paper at the IEEE international conference on Fuzzy Systems (FUZZ-IEEE) at the 2008 World Congress on Computational Intelligence (WCCI). He currently has eight journal articles in print, one under review, and nineteen conference publications in print. He is also a journal reviewer for the Transactions on Fuzzy Systems and a co-chair for special session 'Computational Intelligence for Activity Recognition from Sensed Data' at FUZZ-IEEE in the 2010 WCCI.

Derek has two degrees in Computer Science, an anticipated Ph.D. in Electrical and Computer Engineering, and seven years of teaching and curriculum development experience in Computer Science and Information Technology. He has worked for and alongside individuals in Electrical and Computer Engineering, Computer Science, Information Technology, Anthropology, Health Informatics, Social Work, Physical Therapy, and Nursing. Derek's background is diverse and shows that he is interested in and has worked on many interdisciplinary collaborative research projects. He is currently a National Library of Medicine Predoctoral Fellow and has held nationally funded Research Assistantship positions from the Naval Research Laboratory and the National Science Foundation.

During his time at the University of Missouri, Derek was awarded the 2004 Computer Science superior graduate achievement award, an IEEE 2008 student travel grant to attend the WCCI conference held in Hong Kong China, he was awarded the 2009 outstanding graduate student award in Electrical and Computer Engineering, he is a recipient of a University of Missouri John D. Bies International Travel Scholarship to attend the 2009 International Fuzzy Systems Association (IFSA) conference held in Lisbon Portugal, a recipient of a 2009 University of Missouri graduate school student and faculty spotlight, and is featured in a University of Missouri's graduate research matters publication titled "Interdisciplinary Approach Hopes to Improve Eldercare with Technology Advancements".