A Person-Search System for an Assistive Robot

by

Pavel Bovbel

A thesis submitted in conformity with the requirements
for the degree of Masters of Applied Science

Mechanical and Industrial Engineering

University of Toronto

A Person-Search System for an Assistive Robot

Pavel Bovbel

Master of Applied Science

Department of Mechanical and Industrial Engineering

University of Toronto

2015

Abstract

This thesis presents the design of an assistive robotic system, developed to aid a growing elderly population in maintaining quality of life and age in place independently. The robot is designed to operate in a home environment and aid a person with the performance of independent activities of daily living, primarily focusing on guided meal preparation. This work begins with the development of a novel person-search approach using a time-discretized Hidden Markov Model (HMM). Subsequently the prototyping and development of the robotic system is discussed, followed by a benchmarking of the person-search system and an information gathering experiment evaluating the robot's interactive capabilities. Experiments show that the HMM-based person-search system is effective at locating a target person considerably faster than an uninformed baseline approach.

Acknowledgements

I would like to thank my supervisor, Professor Goldie Nejat, for her continued support, input, and guidance throughout my research. I would also like to thank my M.A.Sc. thesis committee for their time and feedback. Thank you to all the people involved with creating Casper: Crosswing Inc. for their generous support in material and guidance; Chetna Pannu for designing and developing the tablet interface; Leo Zeng and Raphael Silva for their aid in creating and controlling Casper's arms; Elma Li for her contributions in designing Casper's head and hands; and Markus Schwenk for developing the simulation environment. I would like to thank my wife Rebecca Bovbel for her everlasting patience and support throughout my Master's program, my friends and family for never forgetting to ask when I will be done, and finally my lab-mates for their advice and comradeship.

Table of Contents

List of Tables

List of Figures

List of Appendices

Chapter 1
Introduction

1.1. Motivation

A rapidly aging population creates significant issues in maintaining the health and wellbeing of the elderly demographic. By 2051, 25% of the Canadian population will be 65, creating an unprecedented load on healthcare infrastructure [1]. Advancements in home care and assisted living have the potential to maintain quality of life for elderly patients in spite of long-term health issues [2]. Accordingly, members of the elderly demographic overwhelmingly prefer to stay in their homes and age-in-place as independently as possible [3]. A major exclusionary criterion for aging-in-place is cognitive impairment, which affects approximately 25% of elderly individuals [4]. Cognitive impairments, can progressively diminish a person's memory, orientation, verbal skills, visuospatial ability, abstract reasoning and attentional skills [5], increasing the need for assistance with everyday activities.

A key metric in evaluating the functional status of a person is the ability to perform Basic and Instrumental Activities of Daily Living (BADLs and IADLs respectively) [6]. While BADLs cover fundamental self-care tasks such as eating and mobility, IADLs include more complex tasks such as housework and meal preparation which are necessary for independent living. A significant effect of cognitive impairment is the reduction of the executive function, which leads to impairment in maintaining IADLs [7]. This decline can in turn accelerate the decline of cognitive ability, and accordingly reduce quality of life [8].

While acute health risks can be managed through improvements in primary healthcare, new tools for home care and assisted living are necessary to promote preventative care, avoid clinical deterioration requiring care escalation, and alleviate the strain on secondary healthcare providers such as personal support workers [9]. This relief can be achieved by considering devices that can help patients maintain their ability to perform IADLs.

Malnutrition, as a condition of nutritional deficit and a consequence of deficiency in dietary intake, is highly prevalent amongst the elderly population, with incidence rates rising 29%-61%

above that of the general population [10]. The condition is also highly associated with cognitive impairment, and both serve as predictors of mortality [11]. Due to the prevalence of malnutrition in elderly individuals, particularly those with cognitive impairments, promoting healthy meal preparation and eating habits is pertinent to maintaining higher quality of life. To date, even though a number of smart home technologies have been developed to assist the elderly with IADLs, few technologies have been targeted for the kitchen environment.

1.2. Problem Definition and Thesis Objective

The objective of this thesis is to design an assistive robotic system for assisting elderly individuals with performing IADLs. Meal preparation is selected as the target activity due to the benefits of reducing malnutrition as outlined in Section 1.1. The overall assistive scenario is broken down into two primary segments: (i) locating the target individual in the home environment, and (ii) guiding the individual through a meal preparation task in the kitchen area.

1.2.1. Person Search

Prior to commencing the meal preparation activity, the robot has to locate the target person in the environment. To approach the problem intelligently, the system must plan the person search using historical information on person behaviour and location, and new data collected during the search process.

1.2.2. Assistive Activity

During the guided meal preparation activity, the robotic system must be able to intuitively guide a person through the activity, encouraging the person to complete the activity, and to continue using the system in the future.

1.3. Proposed Approach

To satisfy the goals outlined in Section 1.2, a prototype assistive robotic system has been designed and prototyped, capable of locating a person, and executing a guided meal preparation activity. The primary contribution of this thesis is the development of the person search approach, and the subsequent evaluation of the approach using the robot. The overall design and testing process is comprised of the following components, corresponding to appropriate thesis chapters:

1.3.1. Literature Review

Chapter 2 provides a literature review of existing approaches in person search. The first section provides an overview of high-level approaches to modelling person location, and discussion of applicability to the proposed global person search planner. The second section of the literature review outlines investigated approaches to robot exploration planning in the context of the required local person search planner.

1.3.2. Person Search Approach

Chapter 3 describes the design of the system's person search module, which is separated into the global and local search planners. The global search planner is responsible for prioritizing regions in the environment for search, using a novel approach developed to query a time-discretized Hidden Markov Model (HMM) which encodes a user's behaviour and location based on time and evidence of past user activity. The local search planner then controls the robot's search of each region, using a Frontier-based exploration approach to plan a search path.

1.3.3. Robotic System Design

Chapter 4 discusses the design and implementation of the robotic system, including hardware, software, and simulation system design. The hardware design encompasses sensor and actuator configuration for mobility and social interaction, as well as design of the system's user interface. The software design relates to an overall modular system architecture, as well as implementation of the person search, navigation, perception, and interaction components. The simulation system overview describes the steps taken to emulate the robot's low-level systems to allow extensive verification of the search planner

1.3.4. Person Search Experiments

Chapter 5 follows the evaluation of the person search system, including model training and validation, simulated trials, and robot trials. The HMM model is initially validated using the leave-one-out cross validation approach, and further verified using time-compressed simulation. Robot trials are then performed at the Toronto Rehabilitation Institute's HomeLab.

1.3.5. Information Gathering Experiment

Chapter 6 presents the configuration and the results of an information gathering experiment, evaluating the interactive capability of the robotic system as a meal preparation guide amongst a group of participants drawn from our research group. A modified Technology Acceptance Model (TAM) is used to evaluate the robotic system, which incorporates the participant's enjoyment of the system, together with the traditional metrics evaluating perceived ease of use, perceived usefulness, and behavioural intention to use.

1.3.6. Conclusion

Chapter 7 provides concluding remarks on the system development, including highlights of contributions, and a discussion of future work.

Chapter 2
Literature Review

This section provides an overview of work related to the person search component of the robotic system. The two investigated components are (i) the person location prediction model which is used to prioritize the order in which regions in the environment are searched, and (ii) the exploration algorithm used to search for the person in a region of interest.

2.1. Global Person Location Estimation

In the scope of modeling person location based on sequences of location data, a variety of statistical approaches have been explored in prior work; the most common are a variety of nonparametric estimation approaches. In [12], Mozer et al. utilized motion sensors to record real user movements, and implemented an artificial neural network trained with backpropagation to anticipate the next room entered by a person inside of a smart home. This implementation was successful in estimating a person's next, immediate location, and this information was used to control the home's environmental systems (e.g. illumination, temperature). Similarly, Akoush et al. predicted a person's next location using a neural network trained with backpropagation, however in the context of movement between cellphone towers [13]. The authors found that the model generalized well, but only when separate models were trained for groups of users with similar activity patterns. In [14], Oh et al. evaluated both neural networks and decision trees for predicting person locations based on past history and found that both approaches generalized poorly and required extensive per-person tuning. Gambs et al. used higher-order discrete-time Markov Chains to extrapolate person location from a sequence of past locations [15], in [16] the authors note that the approach delivers diminishing returns in terms of generalization for models with order of 3 or higher.

A limitation of the above approaches is the time-agnostic nature of the models used, where time is a dimension that can be extracted from schedule data and is informative for location prediction. The techniques that follow can be classified as time-aware. Luber et al. developed an alternative approach for place-dependent people tracking, by creating a Spatial Affordance Map [17]. This data structure associates space, time and event probability. For each space-time grid, the authors estimate a gamma distribution of an event occurring - in this case the probability of a person being located in a particular area. The concept of the spatial affordance map was extended for human-robot interaction scenarios in a simulated environment by Tipaldi et al [18]. Here, a Markov Decision Process (MDP) was used to iteratively find an optimal path where the probability of a robot encountering a person is maximized. The benchmarking experiments used simulated agents that follow pre-set paths through the environment, based on user data collected through surveys. The authors found that the MDP planner, backed by a Spatial Affordance Map model outperformed uninformed and informed heuristic-based planners. Ihler et al. implemented a Nonparametric Bayesian approach with Poisson processes to cluster profiles from user location, and separated the profiles into distinct patterns for weekend and weekend signatures [19]. The authors' system exhibited a capability to classify the profile of an unlabeled sequence of locations against the labeled training profiles, with the goal of detecting abnormal outlier datasets.

To improve the fidelity of a person location model, it is also possible to reason about the activity or behavioural context of a given location. The subsequent techniques can be classified as behaviour-aware. One approach to applying behavioural context is the use of belief networks to model person location data. Asahara et al. implemented a Mixed Markov Model (MMM) approach to predict pedestrian activity, with the pedestrian's personality serving as a fixed, unobservable, parameter [20]. The authors found that the MMM approach outperforms traditional Markov and HMM approaches for their application by a large margin (74% to 45% and 2% accuracy, respectively). Mathew et al. applied HMMs to location prediction by processing a high volume of GPS data (178 participants, 3 years) [21]. The authors discretized the collected location data based on similarity characteristics, and then trained individual HMM models for each location. Notably, unlabeled data was used during the training process, which necessitates manual tuning to select the number of hidden states in the model. The authors found a low predictive accuracy when using many individual models, and the optimal configuration used one global HMM; this was attributed to large distance scales in the dataset and high variability amongst sampled participants.

While using layered belief networks allows models to infer behavioural context information from location data, another effective approach is to label context directly during data collection. Lee et al. collected data in the form of person location and action chains [22]. This data was then used to train a variety of Bayesian models, namely Naïve Bayesian Networks, Tree Augmented Bayesian Networks, and Dynamic Bayesian Networks (DBNs). The best generalizing approach implemented a DBN model, with a structure that considered not only sequences of locations, but also sequences of actions. Due to the acyclic nature of DBNs, the authors modeled the locations as influencing the actions, however the reverse approach would also be valid depending on the data collected. Similarly, Singla et al. collected fully labeled location and activity data for two residents in a smart home environment [23]. This data was then used to train individual HMMs for each resident, to effectively recognize activities based on movement patterns.

2.2. Local Region Search

A variety of techniques exist for robotic exploration and searching, here the application of these methods to re-exploring a known environment is considered. A classic robot exploration approach was developed in [24], Frontier-Based Exploration (FBE). In FBE, a 2D occupancy grid of the environment is searched for ‘frontiers’ – boundaries between explored and unexplored areas unobstructed by obstacles. The nearest frontier is then provided as a movement goal for the robot. Once the robot reaches the goal, the process is repeated, until all accessible areas of the environment are explored. The frontier-based method is naïve in terms of utility maximization, as selecting the nearest frontier is not necessarily time-optimal exploring the largest amount of unknown area [25]. Stachniss et al. optimized the exploration path to maximize an Information Gain (IG) criterion, where IG is a function of pose and map uncertainty [26]. In several works [27],[28], the IG approach was found to be improve the effectiveness of robotic exploration performed in conjunction with mapping, using a Simultaneous Localization and Mapping approach that provides the requisite information on robot pose and feature uncertainty.

For person-search in a known environment, the FBE method is more directly applicable. Holz et al. evaluate the efficiency of various improvements to the baseline FBE approach, and find two key improvements [29]. First, the authors found that continuously scanning the environment and processing new frontiers, without waiting for the system to reach an initially selected goals, reduces the length of the robot's path during exploration by 10%. A further 10% benefit is achieved by segmenting the environment into regions, and only moving into a neighbouring region after the current region is fully explored.

2.3. Chapter Summary

The first section of the literature review provides an overview of statistical models and approaches used for person location estimation, in the context of the proposed global person search planner. The investigated approaches to modeling location can be classified as generally time-aware, behaviour-aware, or neither. The goal of this research is to incorporate a person location model within the person search approach that is both time- and behaviour-aware. The second section of the literature review outlines methods of robotic exploration applicable to the local person search planner.

Chapter 3
Person Search Approach

To allow for a variety of activity types and schedules, it is important for the robotic system to be able to quickly locate a person in their home environment. Unlike systems with sensing capability distributed through the environment such as remote cameras and motion sensors [12], [30], the robot's sensors are restricted to be self-contained, with the goal of reducing the intrusiveness of the robotic system.

The overall person search approach must enable the robotic system to quickly locate a target person in the environment before the scheduled activity begins. It may utilize information from the robot's perception system, and the approach has access to a historical sample of user activity data. To avoid biasing the results, the approach cannot not rely on notifying the target person of the robot's presence by calling out before the target person is located. Finally, the approach may assume that the robot always starts the search from a fixed location.

The proposed approach is split into two components: the global person search planner guides the overall search process by prioritizing discrete regions (i.e. rooms); and the local person search planner plans the robot's motion through each region during the search operation (Figure 1).

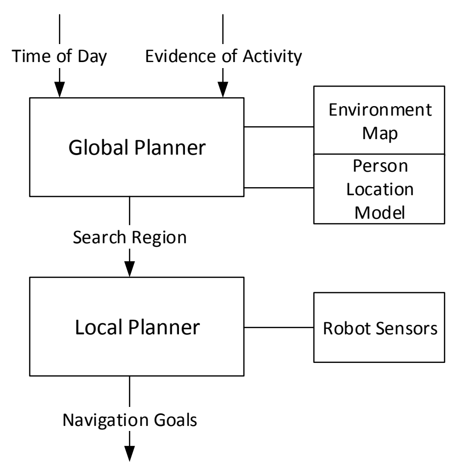

Figure 1: Person Search Planner Structure

3.1. Global Search Planner

The global search planner is prioritizes the overall search process by estimating the person's current location in the environment from a statistical model. The planner applies a novel approach to person location prediction, implementing a Hidden Markov Model-based statistical model to predict current person location. The model is trained using collected user schedules labeled with past location and behaviour, and the model structure incorporates both time- and behaviour-aware components outlined in Section 2.1. The advantage of the implemented approach is that both temporal and behavioural context is incorporated directly in the model, with the goal of improving training efficacy and generalization ability. Additionally, the model can flexibly generate a location estimate with varying amounts of information, specifically in two modes: (i) a naïve inference method where only the time of day is considered, and (ii) an informed inference method where evidence of past user activity is incorporated into the estimation.

3.1.1. Location Model Selection

The selection of the inference model is driven largely by the nature of the data available to the system. In one approach, the system could be trained by having a robot passively observe and track a person during the day-to-day operation of the system, recording data on person location and activity. This approach is undesirable due to the intermittent and time-consuming nature of passive data collection, in addition to the sparse and unlabeled resultant data. Instead, labeled schedules are collected from a questionnaire on daily living patterns (Appendix A).

With labeled schedule data available, the inputs for inference were considered. Given the assumption that the person follows a regular schedule, the most direct input to predict the location of a person is the time of day. For example, the spatial affordance map approach [17] assigns each location a separate learned probability distribution based on the time of day, creating a three dimensional (2D-space, 1D-time) representation. However, this model treats each map segment as conditionally independent, and a model based on location frequency alone may perform poorly due to poor generalization across participants who have high schedule variability [21].

To mitigate intra- and inter-person schedule variations, sequences of person locations were considered to form the basis of the person location model. While the time that a person visits the kitchen may vary on a day-to-day basis, it is reasonable that after visiting the kitchen, the person will often go to the dining room. A location-sequence based structure would enable the model to better generalize across a population who do not demonstrate chronologically similar schedules, but do have similar patterns of movement. A discrete-time Markov chain [31] is well suited to model a sequence of states, in Figure 2 the time period $1{:}\,t$ is represented as having a sequence of location states $x_{1:t}$:

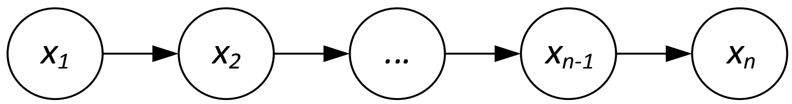

Figure 2: Markov Chain Diagram

A Markov chain represents a memory-less sequence of states. According to the Markov property, the belief at each state x_t is conditionally dependent only on state x_{t-1}, calculated via the transition probabilities in matrix T [31]:

$$P(x_t = j | x_{t-1} = i) = T_{ij} \tag{1}$$

If the state probability distribution x_t is represented as a probability vector $x = \{\mathbb{R}_0, \mathbb{R}_1, \ldots \mathbb{R}_k\}$ for all potential states *k*, each successive state can be calculated as [31]:

$$x_t = T x_{t-1} \tag{2}$$

Location sequences alone fail to take advantage of the multiple forms of context available from user schedules. To incorporate behavioural context into the model, a multilayered Markov model is used instead. Unlike Markov chains, Hidden Markov Models (Figure 3) separate the sequence into two layers: hidden states $x_{1:t}$ and observable symbols $o_{1:t}$ [31] .

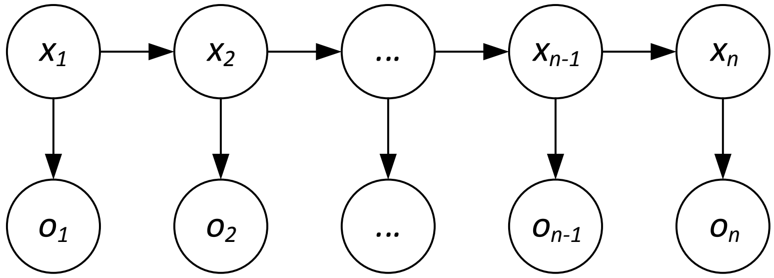

Figure 3: HMM Model Diagram

Instead of learning sequences in location, sequences in behaviour states $x_{1:t}$ are learned instead, and mapped to location symbols $o_{1:t}$. Hidden states x_t model person behaviour, such as 'eating' and 'meal preparation', which are impractical for the system to observe directly. Observable symbols o_t represent the associated locations such as 'kitchen' and 'dining room', person location being a property a robot can perceive directly. The probability of location symbol o_t being observed during behaviour state x_t is governed by the emission probabilities in matrix E [31], where:

$$P(o_t = j | x_t = i) = E_{ij} \tag{3}$$

Each state x_t is conditionally dependent only on the prior state x_{t-1} and the observation o_t, and observations are conditionally dependent only on their associated hidden state. The following equation applies for the probability vector form of $o = \{\mathbb{R}_0, \mathbb{R}_1, \ldots \mathbb{R}_l\}$ for all possible locations l [31]:

$$o_t = Ex_t \tag{4}$$

The HMM approach is a reduced version of the DBN structure used by Lee et al. [22], where neighbouring location sequences were also conditionally dependant. This simplification allows the use of efficient training and inference methods as described in the following section.

Finally, to add temporal context and allow the HMM model to predict person location at different times of day, the time quantization technique previously used in the spatial affordance maps [17] is applied. The day is discretized into a quantity of time-blocks, and separate HMM sub-models are trained for each time-block, using the associated subset of training data to calculate parameter matrices T and E. The exact number of sub-models must be determined empirically from the schedule data, to strike a balance between model accuracy and the coverage of collected training data. Too few sub-models would generalize poorly across differing schedules, while too many would require splitting the schedules such that there is insufficient training data in each sub-model to learn behaviour and location sequences.

3.1.2. Location Model Training

The goal of training the HMM is to select parameters in the transition matrix T and emission matrix E. In the unlabeled case, the training data is comprised of a sequence of observable symbols $\varphi_{1:t}$, while in the labeled case the data also includes a sequence of hidden states $\pi_{1:t}$. If only the symbols φ are available, the Baum Welch [31] algorithm can be used to iteratively update T and E until they converge using an expectation maximization approach. This technique may be used for training the model in a live environment, while collecting data about a person's location activity, and inferring the hidden state while gradually building the model over time. Operating a robot in this manner would be a very lengthy and intrusive process, so the model is trained with collected participant-reported schedules of behaviour and location (π and φ respectively).

Each participant submits a typical schedule of his/her activities, providing information regarding daily behaviour and location (e.g. 9 AM – prepare breakfast in kitchen, 9:30 AM – eat meal in dining room; full schedule sample provided in Appendix A). In order to add temporal context to the model, separate HMM sub-models are trained based on a specific time-interval of all user schedules. Each behaviour is classified as one of set X, and each location as one of set O. Since the collected schedules are labeled, maximum likelihood estimation is used to directly evaluate all the elements of matrices T and E, where the count function $\delta(\pi_n, X_i) = 1$ when the n^{th} state of sequence $\pi_{1:t}$ is of class X_i, and $\delta(\varphi_n, O_i) = 1$ when the n^{th} observation of sequence $\varphi_{1:t}$ is of class O_i [32]:

$$T_{ij} = \frac{\sum_{n=1}^{t-1} \delta(\pi_n, X_i)\delta\left(\pi_{n+1}, X_j\right)}{\sum_{n=1}^{t-1} \delta(\pi_n, X_i)} \tag{5}$$

$$E_{ij} = \frac{\sum_{n=1}^{t} \delta(\pi_n, X_i)\delta(\varphi_n, O_j)}{\sum_{n=1}^{t} \delta(\pi_n, X_i)} o_t = Ex_t \tag{6}$$

For smaller datasets, it is possible that certain states or observations classes will go unused, causing the equations above to be undefined due to zero-counts in the numerator and denominator. Due to the time quantization of the model and gaps in the training data, the situation may arise where certain locations or behaviours are not selected by participants during a specific time period. This effect is prevented by using additive smoothing [33] to augment each count in the equations above, where $|X|$ and $|O|$ represent the size of the sets of hidden states and observations respectively, and matrices T and E represent the smoothed estimation for the parameters of the model:

$$T_{ij} = \frac{\sum_{n=1}^{t-1} \left(\delta(\pi_n, X_i)\delta(\pi_{n+1}, X_j)\right) + 1}{\sum_{n=1}^{t-1} \delta(\pi_n, X_i) + |X|} \tag{7}$$

$$E_{ij} = \frac{\sum_{n=1}^{t} \left(\delta(\pi_n, X_i)\delta(\varphi_n, O_j)\right) + 1}{\sum_{n=1}^{t} \delta(\pi_n, X_i) + |O|} \tag{8}$$

These parameters can then be used for the two inference modes outlined below.

3.1.3. Naive Inference

At the start of the person search process, the primary information available to query the model is the current time of day. A naive estimate for a person's behaviour at a particular time of day, given no other information, can be derived from the stationary distribution of the backing Markov chain. The stationary distribution is defined as the probability distribution α where [31]:

$$\alpha T = \alpha \tag{9}$$

This can be obtained iteratively by multiplying a uniform probability distribution U by the transition matrix T until it converges [31]:

$$\lim_{n\to\infty} UT^n = \alpha \tag{10}$$

In effect, the stationary distribution is the estimated behaviour state of the person that the model settles into, when no information about the person's current or past location is available. The time of day is used only to select the sub-model with parameter matrices T and E. The unbiased location estimate can then be extracted from the stationary [31]:

$$o^* = E\alpha \tag{11}$$

The probability vector o^* has components associated with each possible person location, and the person-search priorities can be obtained by ranking the components of o^*.

3.1.4. Informed Inference

Throughout the search process, a robot is capable of collecting information that would change the way the overall person search should be prioritized. This information can be captured in the form of 'evidence' of past activity – the canonical example involves the robot finding dirty dishes in the kitchen sink. This evidence translates directly into HMM symbols, or observations of past location. An observation of dirty dishes, indicating the person was recently in the kitchen, informs the past symbol distribution, and biases the past state likelihood distribution towards behaviours such as meal preparation and eating.

Since the symbol distribution is non-uniform, the stationary distribution shortcut used for the naïve case can no longer be taken. Instead, the forward-backward algorithm [32] is used to calculate the probability vectors of the underlying behaviour sequence $x_{1:t}$, which is then forward-simulated to estimate o^*.

The forward-backward algorithm is a dynamic programming method to efficiently calculate the marginal probabilities of the hidden states associated with a sequence of observations, given model parameters T and E and a prior hidden state distribution. Multiple pieces of evidence provide a sequence of symbols $o_{1:t-1}$, and allow calculating the corresponding sequence of past behaviour states $x_{1:t-1}$. The prior distribution x_0 can be drawn from the previously calculated stationary distribution, as this is the best estimate of the state prior to any observations. To obtain the marginal probability vector of state x_k given the sequence $o_{1:t-1}$, the chain rule is used to split the proportional joint probability [32]:

$$P(x_k|o_{1:t-1}) = \frac{P(x_k, o_{1:t-1})}{P(\, o_{1:t-1})} \tag{12}$$

$$= \frac{P(x_k, o_{1:k})P(o_{k+1:t-1}|x_k, o_{1:k})}{P(\, o_{1:t-1})} \tag{13}$$

Conveniently, due to the structure of the HMM, $o_{1:k}$ and $o_{k+1:t-1}$ are conditionally independent given x_k (Figure 4):

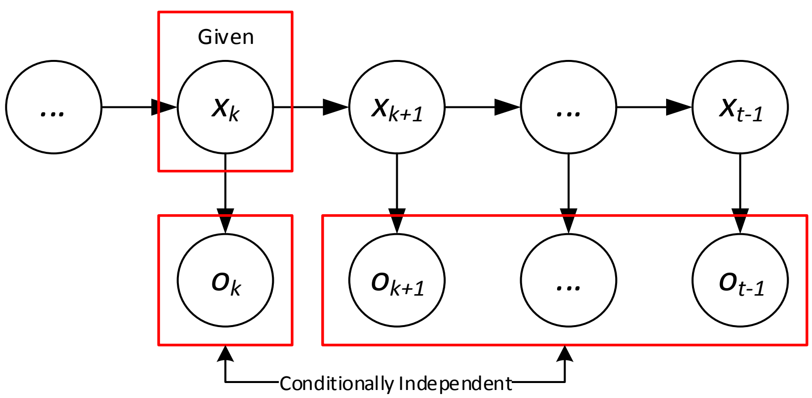

Figure 4: HMM Conditional Independence

This property, known as d-separation [32], reduces equation (13) to:

$$P(x_k|o_{1:t-1}) = \frac{P(x_k, o_{1:k})P(o_{k+1:t-1}|x_k)}{P(\, o_{1:t-1})} \tag{14}$$

Due to exponential complexity, this distribution is impractical to calculate directly for any individual state x_k, but can be determined across the whole model in a two-pass approach using the forward-backward algorithm [32].

3.1.4.1. Forward Probability

For the first pass, the forward probability vector α_k is defined as the joint probability of emitting the first k observations ($o_{1:k}$) and transitioning to state x_k [32]:

$$\alpha_k = P(x_k, o_{1:k}) \tag{15}$$

The forward probability is calculated recursively and normalized, with the prior being equivalent to the stationary distribution x_0, Where $(x)^D$ is the diagonalization of vector x, and c^{-1} is a normalization constant across the vector components [32]:

$$\alpha_0 = x_0 \tag{16}$$

$$\alpha_k = c^{-1}\alpha_{k-1}T(Eo_k)^D \tag{17}$$

3.1.4.2. Backward Probability

For the second pass, the backward probability vector β_k defines the joint probability of emitting the remaining observations $o_{k+1:t-1}$ after state x_k [32]:

$$\beta_k = P(o_{k+1:t-1}|x_k) \tag{18}$$

The backward probability is calculated recursively, but in the opposite direction for forward probability, with the posterior β_t being given analytically evaluated as a uniform probability vector [32]:

$$\beta_t = c^{-1}[1 \quad \dots \quad 1] \tag{19}$$

$$\beta_{k-1} = c^{-1}T(Eo_k)^D\beta_k \tag{20}$$

3.1.4.3. Location Estimation

After calculating the forward and backward probabilities recursively for each state x_k, the marginal probability distribution can be efficiently calculated given sequence $o_{1:k}$ by combining the components α_k and β_k via Equation (14) [32]:

$$P(x_k|o_{1:t-1}) = \frac{P(x_k, o_{1:k})P(o_{k+1:t-1}|x_k)}{P(\ o_{1:t-1})} \tag{21}$$

$$= c^{-1}\alpha_k\beta_k \tag{22}$$

The distribution of interest is for state x_{t-1}; due to the nature of the HMM, the last state is the only one that directly affects the estimation of states $x_{t:\infty}$. To that end, the transition matrix T is applied to obtain the 'present' state x_t from the marginal distribution of x_{t-1}. However, since the exact time elapsed between the present time and when the symbols observed by the robot is unknown, the estimation is smoothed by forward-simulating the Markov behaviour chain using the transition matrix T and obtaining a set of some n future states $\{x_t \dots x_{t+n}\}$.

To extract the predicted future location from the simulated state sequence, the emission matrix E is applied to each future state x_{t+i}. To prevent the future states from converging to the stationary distribution as in the naïve case, a discount factor γ is also applied to each state x_{t+i}, causing each successive state's contribution to the estimate to decrease with time distance from the 'present' state x_t.

The location estimate of probability vector o^*, aggregated over the sequence future states, can be formulated as:

$$o^* = \lim_{n\to\infty}\sum_{i=0}^{n} E(1-\gamma)^i x_{t+i} \tag{23}$$

$$= \lim_{n\to\infty}\sum_{i=0}^{n} E(1-\gamma)^i T^i x_t \tag{24}$$

A larger discount factor causes the iterative estimation process to converge faster and use less forward-simulated states in the process. Similar to the naïve inference case in Section 3.1.3 the magnitude of each of the components of o^* is ranked to obtain priorities for person-search locations.

3.2. Local Search Planner

The local search planning process involves using a map of the environment to select navigation goals to locate the target person. Since the map is already known, this can be abstracted as a re-exploration process, with the goal of exploring a bounded region while continuously scanning for the person. To that end, the Frontier-Based Exploration (FBE) strategy [24] is used for the local person search planning.

3.2.1. Exploration Approach

Prior to starting exploration, the region map, obtained from the global planner, is discretized into a grid, with navigable areas marked as 'free cells', and obstructed areas marked as 'occupied cells' (Figure 9a). Since a previously explored region is being re-explored, we label all free cells outside the robot's field of view as 'unknown cells', and overlay the region boundary as additional occupied cells to prevent the exploration process from leaving the target area. The exploration goal is then determined with the following procedure:

1. Determine set of unknown cells U_R reachable from the current robot position via breadth-first-search on a 4-connected neighbourhood (Figure 6a) on all free cells (Figure 5).

function breadth-first-search(position $\boldsymbol{P}$, grid $\boldsymbol{G}$, free $\boldsymbol{F}$, unknown $\boldsymbol{U}$)
$\quad Let\ Q\ = \{P\}$
$\quad Let\ U\ = \{\}$
$\quad while\ (Q \neq \{\})$
$\quad\quad Let\ C = Q_1$
$\quad\quad Q = Q - \{C\}$
$\quad\quad foreach\ X \in (C_{n4} \cap \mathrm{G})$
$\quad\quad\quad if\ X \in F \rightarrow Q = Q + \{C\}$
$\quad\quad\quad if\ X \in U \rightarrow U_R = U_R + \{C\}$
$\quad return\ U_R$

Figure 5: Breadth-first-search approach

Figure 6: a) 4-connected neighbourhood b) 8-connected neighbourhood

2. Cluster all unknown cells U in 8-connected neighbourhoods (Figure 6b) into frontiers F, ignoring any frontiers that only contain one unknown cell (Figure 7). The resultant frontier clusters are shown in Figure 9b.

function cluster-frontiers(cells $\boldsymbol{U_R}$, grid $\boldsymbol{G}$)

$\quad Let\ F\ = \{\}$
$\quad while(U\ \neq\ \{\})$
$\quad\quad Let\ X\ = \{\}$
$\quad\quad Let\ Q\ = U_1$
$\quad\quad while\ (Q\ \neq\ \{\})$
$\quad\quad\quad Let\ C = Q_1$
$\quad\quad\quad Q = Q - \{C\}$
$\quad\quad\quad X = X + C_{n8} \cap \mathrm{U}$
$\quad\quad\quad U = U - C_{n8} \cap \mathrm{U}$
$\quad\quad if\ |X| >\ 1 \rightarrow F = F + \{X\}$
$\quad return\ F$

Figure 7: Frontier clustering approach

3. Determine the search goal G as the center of the frontier closest to the current robot position (Figure 8).

function get-goal(position $\boldsymbol{P}$, frontiers $\boldsymbol{F}$)

$\quad Let\ D_{min}\ =\ \infty$
$\quad foreach\ X \in \mathrm{F}$
$\quad\quad Let\ D = distance(X_{mid}, P)$
$\quad\quad if\ D\ <\ D_{min} \rightarrow$
$\quad\quad\quad min\ =\ D$
$\quad\quad\quad Let\ G\ =\ X_{mid}$
$\quad return\ G$

Figure 8: Closest frontier selection approach

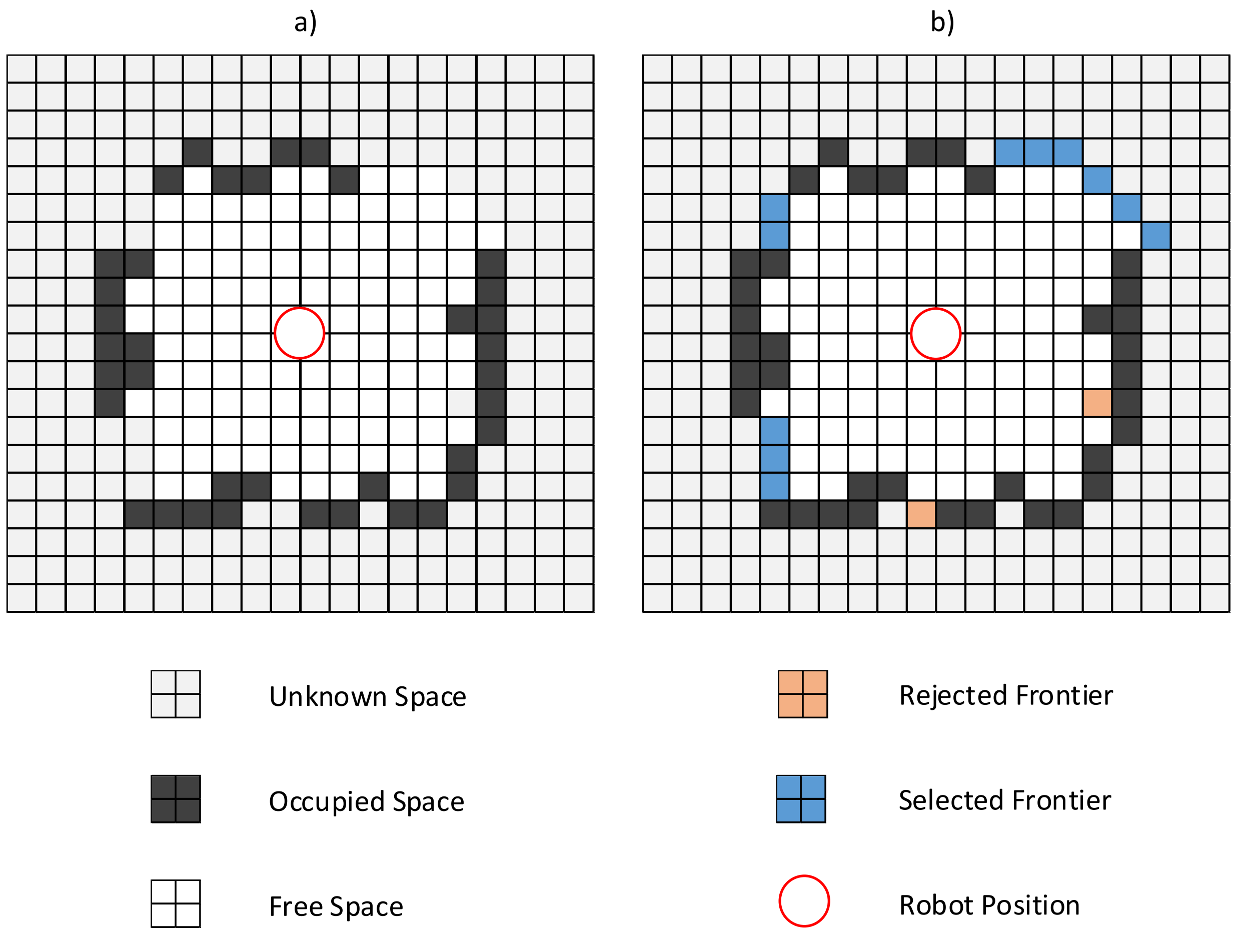

Figure 9: a) Initial gridmap, b) Clustered and filtered frontiers

Once the target frontier is selected, the robot is provided with the spatial coordinates of the frontier's centre as a navigation goal. As the robot moves through the environment, cells within the robot sensor's field of view (FOV) are explored and marked clear on the gridmap. If the target person is located during exploration, the process is halted and the next interaction phase can begin; otherwise the robot moves on to the next exploration region, as dictated by the global search planner.

To improve Frontier-based exploration performance, the frontiers are continuously recalculated to prevent the robot from planning longer paths than necessary [29]. Additionally, the global/local search planner architecture intrinsically applies the room-segmentation optimization from [29], as each bounded region provided to the local search planner already represents an individual room. The limitation of this local search approach is the reliance on locating the person within the robot sensor's limited FOV; this can be mitigated by selecting a person detection approach with a consistently high detection rate, and masking the exploration sensor's field of view with the effective field of view of the person detection system.

3.3. Chapter Summary

This chapter discusses the person search approach used in the development of the assistive robotic system, separated into the global and local search planning components. The goal of the person search planner is to intelligently locate a target person in a home environment, prior to the start of a scheduled activity. The global search planner implements a time-based Hidden Markov Model approach, where multiple HMM sub-models are responsible for discrete blocks of time, and each sub-model encodes association in behaviour and location extracted from labeled schedule data. The global search planner ranks regions in the environment in terms of likelihood that the person will be found there. The local search planner then controls the robot's search of each region, using a Frontier-based exploration approach to plan a search path.

Chapter 4
Robotic System Design

The Casper socially-assistive robotic system was designed to execute the person search approach and an assistive meal preparation interaction. Casper is comprised of a humanoid torso, with attached arms and head, mounted on an omni-directional mobile base. Software for the robot is designed using the Robot Operating System (ROS) [34] framework, which allows for many independent software components (nodes) to run on distributed hardware components. This chapter details design specifics and considerations for the robot's hardware and software components. The robot is shown in Figure 10.

Figure 10: Casper Robot

4.1. Hardware and Low-level Software Design

The Casper robot is comprised of off-the-shelf electronics components integrated into a custom wheeled-humanoid robot frame. The frame consists of 3D printed ABS plastic components, with two steel stiffener rods running vertically through the frame. These stiffeners help reduce the oscillations present in the chassis due to the robot's relatively tall height and small footprint. These oscillations are additionally mitigated by placing the heaviest robot components (i.e. battery, drive motors, and computer) in the base of the frame.

4.1.1. Mobile Base

The Casper robot is built on top of the Nav2 omni-directional mobile base, manufactured by Crosswing Inc. [35]. The base is holonomic due to the triangular configuration of its three omnidirectional wheels, shown in Figure 10, allowing the base to move freely in 3 degrees of freedom (DOF).

Figure 11: Nav2 Mobile Base with initial sensor integration

The holonomic omni-wheel drive arrangement provides Casper with great flexibility in confined indoor environments, and exit unexpected dead-ends created by dynamic obstacles. Additionally, the drive system reduces the constraints on trajectory sampling during path planning and obstacle avoidance, allowing the robot to follow complex curved paths, making the robot appear more graceful to an observer. Velocity control and dead-reckoning is performed by an independent embedded Linux microcontroller, which receives velocity commands and provides a net odometry estimate to the robot's central computer using an Ethernet connection.

4.1.2. Non-contact Arms

Casper requires actuated arms primarily to perform prompting and social gestures during activity interaction. While contact-capable arms typically require 6+ DOF designs, a non-contact design that implements a subset of gestures can be accomplished with significantly less complexity and cost. Prior to formulating the design, gestures required for the target interactions were isolated as: waving, pointing, and raised 'hand-talking' during conversation. In [36], Y. Zeng determined that Casper could formulate these gestures with as little as 3 DOF – shoulder pitch, shoulder roll, and elbow pitch, shown in Figure 12. Specifically, this configuration allows Casper to wave with either arm, point to any landmark in front of the torso, and appear to naturally move its arms during conversation.

For design and control simplicity, the end 'effector' of the arm was designed by Y. Li in [37] as static, but multi-purposed as an open palm and a pointer. This allowed performing the pointing and waving gestures effectively during interactions, shown in Figure 13. The non-contact design constraint also reduces the load requirement on the arm actuators, which are required to support only the arm's weight. The arms were fabricated by Crosswing Inc. [35] from lightweight ABS plastic, allowing the selection of low-power servomotors by Y. Zeng in [36] to be used at each linkage. The servomotors are sufficiently limited in power such that they cannot generate an arm momentum significant enough to harm a person, and can be easily back driven. This allowance also permits the design to forgo the use of compliant actuators. In [36], Y. Zeng created the initial arm design, which was improved and prototyped in the scope of this work. Specific refinements included a compact form factor with reduced gaps in the arms' plastic shell, and integration with the electrical actuators and other components.

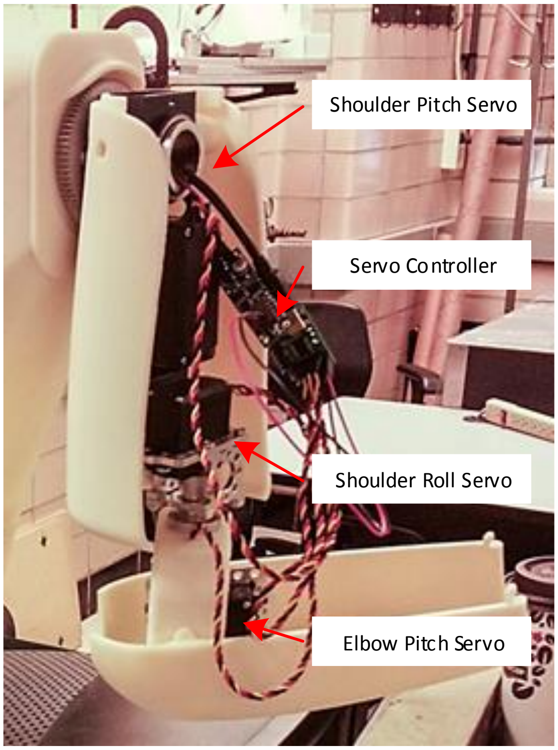

Figure 12: Casper Arm Internals

Figure 13: Casper Waving Gesture

Each arm carries an independent servo controller that performs low level control and trajectory interpolation, receiving high-level angle commands from the robot's central computer. During robot operation, software components actively monitor the arm workspace, preventing arm operation when a person or obstacle is inside the workspace and could trigger a collision. The 3-DOF arm configuration allows for simple closed-form implementations of static (e.g. greeting, waving) and dynamic (pointing to objects) gestures.

4.1.3. Emotive Head

Casper's head was designed to fulfil basic social functions during interactions, mimicking human facial emotion and supplementing arm gestures during interaction. The initial physical concept was created in the scope of this thesis, and Y. Li performed the final design and prototyping in [37]. The head consists of LED matrices representing the robot's mouth and eyebrows, animatronic eyes, and a 2-DOF (pitch and yaw) servo configuration for neck actuation, shown in

Figure 14. The head components are mounted inside a lightweight ABS plastic shell (manufactured by Crosswing Inc. [35]), and are controlled using an Arduino Mega microcontroller. The micro-controller receives a target head orientation and emotion from the central PC, and then executes the commands using all applicable components.

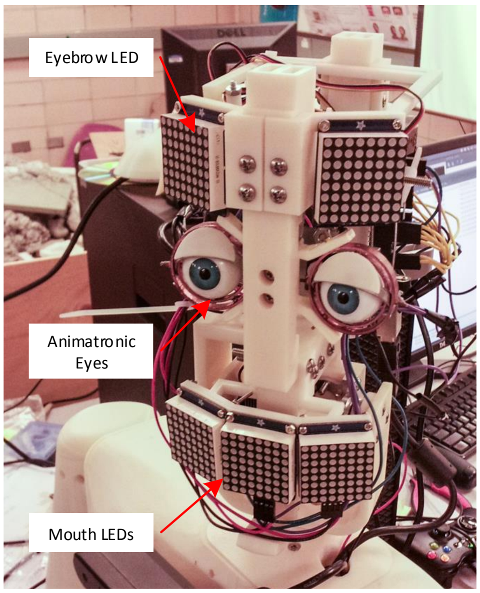

Figure 14: Casper Head Internals

Figure 15: Casper 'Speaking' Animation

The head is capable of displaying five generic human-life facial expressions (happy, sad, angry, surprised, and neutral, shown in Appendix C) to accompany the robot's social interactions, as well as a 'speaking' animation, shown in Figure 15. The neck actuation is used for basic communicative gesturing, as well as accompanying the arms during 'pointing' gestures to indicate the location of interest. To make head orientation changes appear more lifelike, Casper's eyes lead the head rotation before settling to the center position.

4.1.4. Sensing and Perception

To perform the necessary autonomous navigation and interaction requirements, Casper is equipped with several exteroceptive sensors, connected to the central computer module via USB. The primary navigational sensor is a Hokuyo URG-04LX-UG01 2D scanning rangefinder [38], which has an update rate of 10 Hz, wide FOV of 240°, and high resolution (0.36° per scan). This allows the rangefinder to be used effectively for 2D mapping, localization, and global path planning within the target environment when mounted at the robot's midsection, shown in Figure 16.

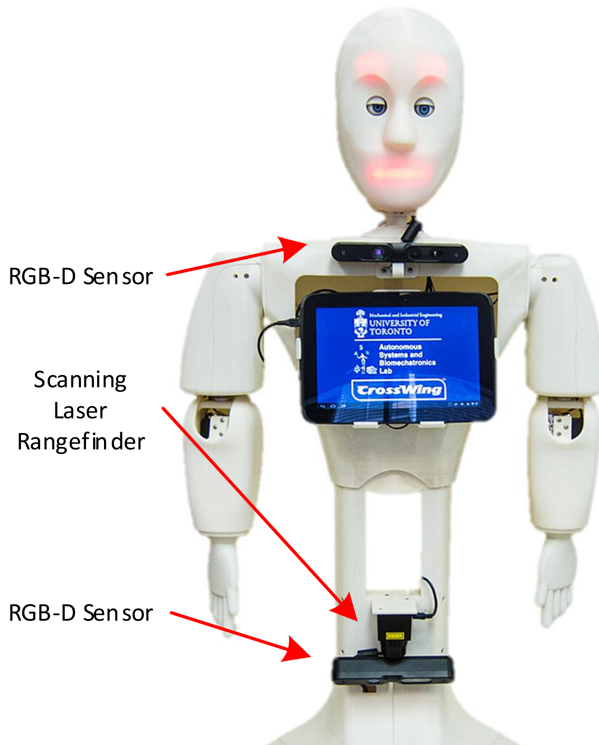

Figure 16: Sensor Diagram

Due to the presence of variable-height obstacles within the home environment (e.g. table surfaces), a 2D planar sensor is insufficient for robust obstacle avoidance. Two ASUS Xtion RGB-D [39] cameras are mounted on the robot, which operate at 60 Hz in QVGA resolution. To capture information along the full height of the robot, one camera is placed on the mobile base for detecting low-lying obstacles, and the other is located on the upper torso for detecting tall obstacles. The resultant point clouds are processed using the PCL statistical outlier removal (SOR) and voxel filters [40], to remove noise and reduce cloud density respectively. The reduced point clouds are transformed into the rangefinder's coordinate frame, and projected onto the floor plane into an emulated laserscan for use by the navigation module described in Section 4.2.1. The overall point cloud processing pipeline is shown in Figure 17.

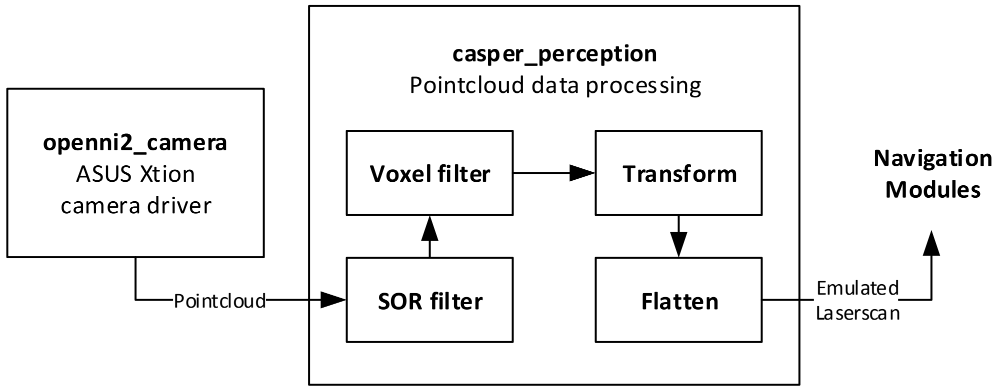

Figure 17: Point Cloud Pipeline Diagram

4.1.5. User Interface

Casper's chest-mounted Android tablet serves two main purposes: to guide the person through the meal preparation activities, and to implement the text-to-speech service used for the robot's spoken interactions. The text-to-speech service was created in the context of this thesis, and the tablet user interface (UI) was designed by C. Pannu [41] to target a user population of elderly and cognitively impaired individuals. Due to potential limitations in cognitive capacity of the user, and the structured nature of the meal preparation activity, a simple sequential interaction flow was implemented. To accommodate issues with vision and fine motor control, individual UI elements have control surfaces marked with large, clear text labels and associated graphic symbols. Screen captures of the designed interface, along with images of the proposed user interaction flow can be found in Appendix B.

The finite state machine for interaction during meal preparation is outlined in Figure 18. At each screen, the user is presented with four actions: return to the initial recipe selection screen (i.e. the 'Home' screen), proceed to the next step in preparation, return to the previous step, or request additional help from the robot.

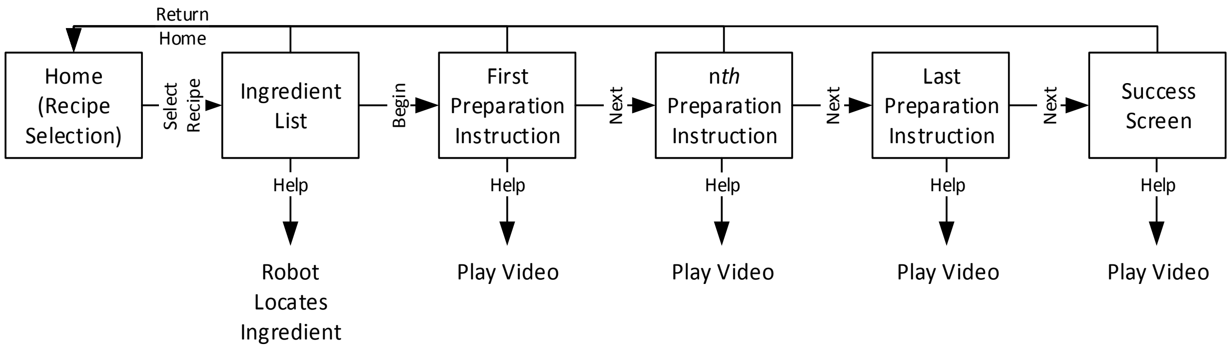

Figure 18: Interaction Finite State Machine Diagram

The interaction begins with the user selecting a meal recipe from the ‘Home’ screen. The UI then enters the ‘Ingredient List’ screen, where the user has the option of requesting the robot’s help in locating an ingredient within the environment. Once the user has located the ingredients, the UI moves into a series of ‘Preparation Instruction’ screens. Here, requesting help causes the UI to play a pre-recorded video of the respective preparation step. Once all preparation steps have been completed, the UI enters the ‘Success’ screen to notify the user that the interaction is complete.

4.2. High-level Software Systems

Casper's software modules are responsible for processing functions involved in planning, navigation and interaction, as well as the control of physical subsystems and sensors outlined in Section 4.1. All software modules are implemented using the ROS API for communication, scheduling, and coordinate transformation. Individual modules run primarily on Casper's central computer, which serves as the main processing hub for all sensing and control operation, exposing the requisite control interfaces to external controllers (i.e. high level planners) through a wireless (Wi-Fi) connection. Peripheral components connect to the central computer using USB and Ethernet; these peripherals include a chest-mounted Android tablet PC for user interaction, as well as independent low-level controllers for the robot's arms, head, and mobile base.

4.2.1. Navigation System

To achieve the mobility required for the target application, the Casper robot implements a multi-tiered navigation system, processing data from a combination of 2D and 3D sensors via global and local planners (Figure 9); sensor selection considerations are discussed in section 4.1.4. Initially, a map of the environment is created by exploring the environment using the Frontier – based exploration method, processing 2D scanning rangefinder and odometry data using a grid-based Simultaneous Localization and Mapping (SLAM) approach [42], implemented in [43]. Since the environment is predominantly static, this map is used during subsequent operation for localization via the Adaptive Monte Carlo localization approach [44], implemented in [45].

The navigation system receives movement requests from the higher level path planner in the form of a target 2D pose in the environment (x, y, and yaw angle). The system then passes the request to the global planner node, which uses the static grid map, 2D rangefinder data, and the localization estimate to create a path plan using the A* search algorithm [46]. Once an approximate path through the environment is generated, the waypoints are passed to the local planner node. The local planner uses the 2D scanning laser rangefinder data, along with supplementary obstacle data from the 3D sensors, to generate velocity commands using the Dynamic Window Approach algorithm [47], implemented in [46]. The mobile base driver node then travels from point to point by executing these velocities. Incorporating the 2D and 3D data alongside the static map allows the planners to avoid collisions with static obstacles that weren't captured by the 2D rangefinder during mapping (e.g. relocated chair, closed door, table surface). If a global plan is found to be infeasible by the local planner, a re-plan at the global level can be triggered with the updated environment data, and a new path plan created.

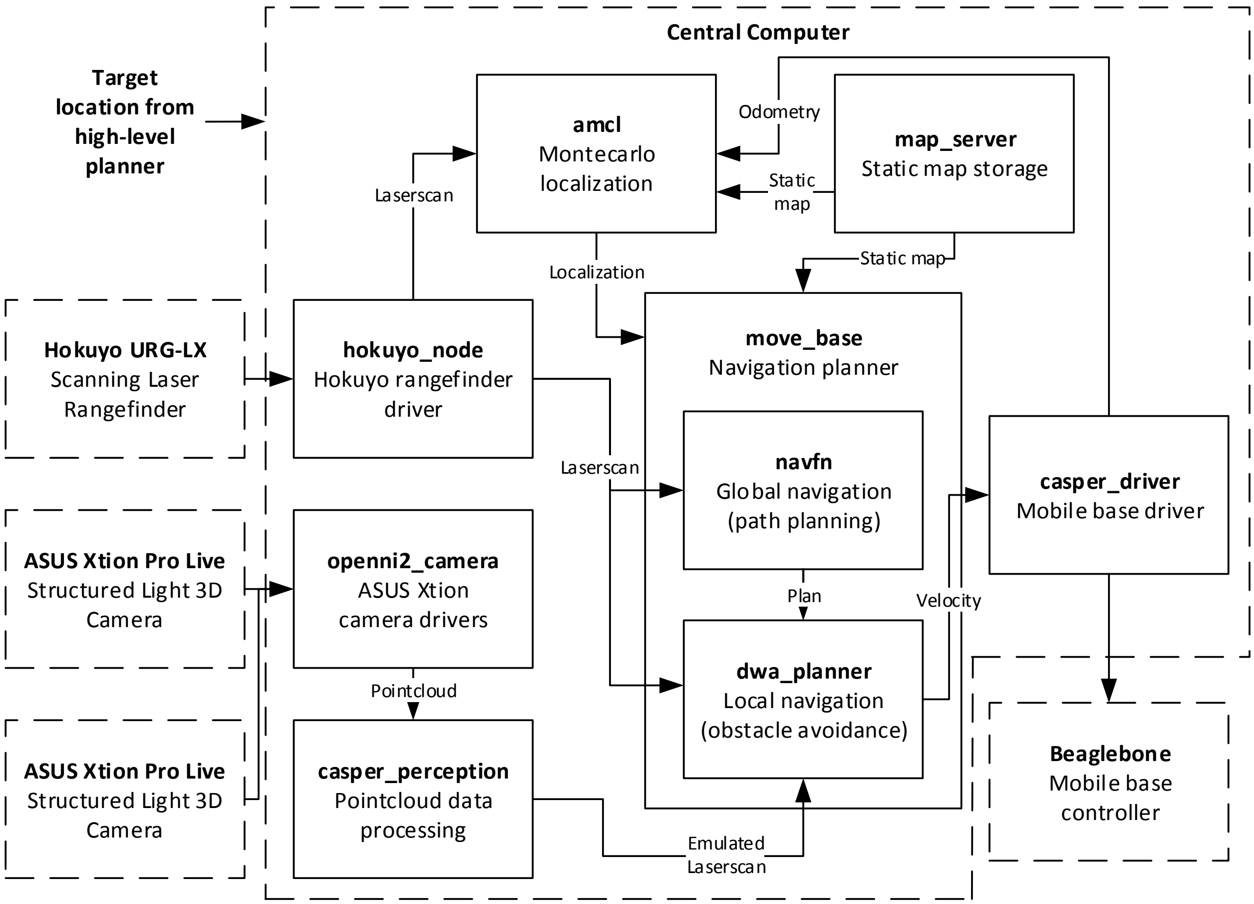

Figure 19: Navigation System Diagram

4.2.2. Person Search Planner

The person-search planner described in Chapter 3 is implemented in a robot-agnostic fashion using ROS, ensuring that the planner can function on any robot or simulator that follows ROS conventions and API. The planner's model is a time discretized HMM that can be used to predict a target person's probable location in the environment. The planner is provided a pre-generated environment map, with boundaries defined for each sub-region (e.g. kitchen, dining room). Once a person-search is initiated, the model is queried to obtain a room-by-room search prioritization. The person-search planner then begins searching the environment, exploring each individual room using the modified Frontier-based search method described in Section 0. Movement commands are passed to the navigation planner outlined in Section 4.2.1. The robot continues searching rooms in sequence until either the target person is found, or detected evidence of past presence in the environment triggers an update in the room priority rankings.

Initially, Casper was designed to detect the target person and evidence markers using HOG feature detection [48] algorithm, using 3D point cloud data from the RGB-D cameras. In practice, it was found that the central computer's processor (Intel 3rd Gen Core i7-3517UE [49]) was insufficiently powerful to perform this function simultaneously with the other software components. Performing this analysis on a remote computer was investigated as an alternative, but it was found that streaming the necessary 3D point cloud information over a wireless link was impractical due to bandwidth and latency limitations. A QR code recognition module was used instead [50], since the requisite 2D camera data could easily be compressed and streamed to a remote computer. This allowed marking the target person and evidence of interest with a unique QR identification code, which is then processed by the search planner on detection by the robot.

4.3. Simulated Systems

In order to evaluate the high-level person search planner, the relevant hardware and low-level software modules were implemented as simulation modules for a simulator developed in collaboration with M. Schwenk et al. [51]. The simulator models the target environment (TRI HomeLab, Section 5.1) via a 2D map segmented into rooms. The mobility system of the robot is emulated with an idealized 2D motion model, allowing the system to deterministically move through the simulation environment. The person- and evidence- detection components of the person-search planner were emulated with a detection scheme that recognized objects of interest within the robot's FOV at a fixed success rate. The idealized simulation components allowed for testing of the high-level systems at highly compressed rate, running at 20:1 real-time ratio.

4.4. Interaction Library

Casper's user interaction modules are designed to fulfil the functions required by the target application, including performing arm gestures, displaying facial emotions, communicating via speech, and guiding participants through meal preparation. All interaction functionality is exposed through an intermediary library API, which then sends the appropriate command to the appropriate hardware module (Figure 20).

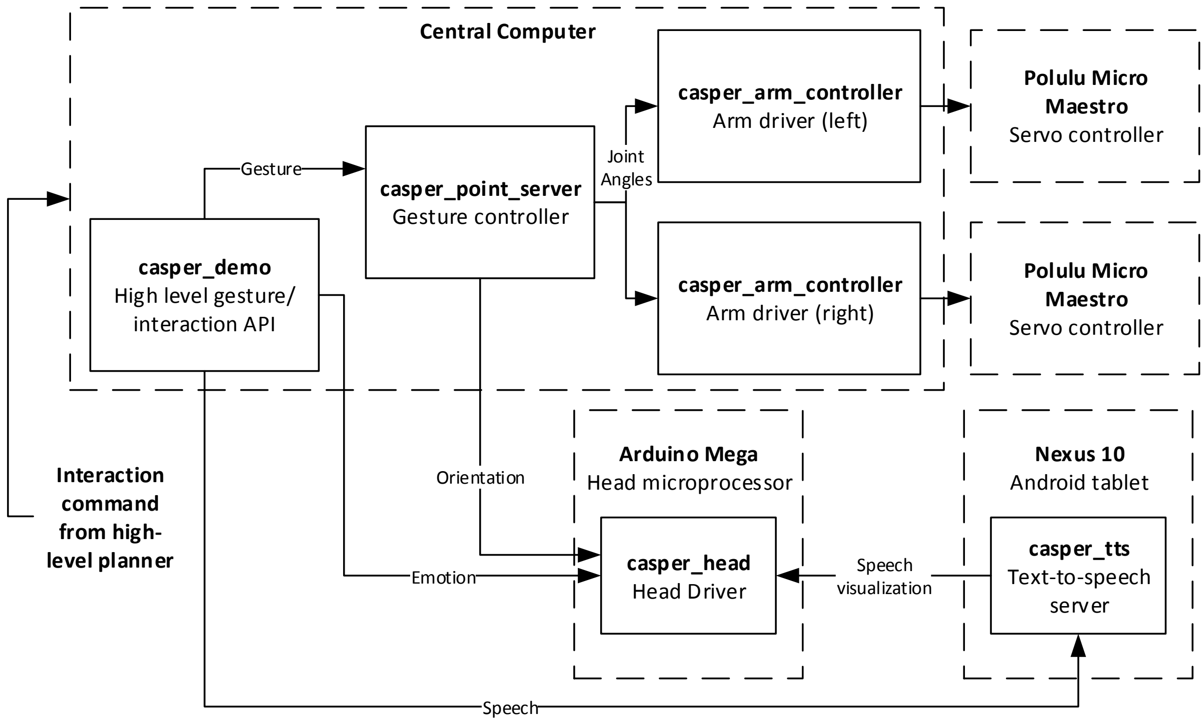

Figure 20: Interaction System Diagram

Arm gestures include static gestures (e.g. waving), as well as dynamic gestures (e.g. point at object). The gesture controller node calculates the angle trajectories necessary for the requested interaction, and streams the appropriate angles to the arm driver nodes. Some gestures also require an associated head movement, which is similarly sent to the head driver module by the gesture controller. The head driver module is also responsible for processing facial emotion requests, which control the robot's eyebrows and mouth. The facial emotions are primarily static (e.g.

happy smiling face, Figure 38), however a moving-mouth animation is implemented for when the robot is speaking. Spoken interactions are processed by the text-to-speech module. The robot also interacts with participants through the aforementioned tablet interface, which allows the person to select from a list of available recipes for meal preparation.

4.5. Chapter Summary

In this chapter, the overall robotic system design is presented, including components of the hardware, software, and simulation system design. The hardware design discusses sensor and actuator configuration for mobility and social interaction, as well as design of the system's user interface. The software design comprises of a modular system architecture, as well as implementation and integration of the person search, navigation, perception, and interaction components. The simulation system overview describes the steps taken to emulate the robot's low-level systems to allow verification of the person search approach.

Chapter 5
Person Search Experiments

The person search system proposed in Chapter 3 is evaluated in three stages. First, the requisite schedule data is collected from users drawn from our research group. Subsequently, the backing HMM inference model is trained and tuned using performance benchmarks. Finally, the person-search system is evaluated on simulated and real systems.

5.1. Target Environment

The person search system is generally applicable to any home environment, and for system testing the person search approach was evaluated at the Toronto Rehabilitation Institute's HomeLab [52] – a staged single-story dwelling used to "test new products to help older people and those with disabilities stay at home longer and more safely". An idealized floorplan of the environment is shown in Figure 21.

Figure 21: HomeLab Idealized Floor Plan

For the purposes of our experiments, the mapping and exploration approach outlined in Section 4.2.1 is used to generate a navigable 2D map of the environment, shown in Figure 22. The navigable area of each room is limited in certain areas, most noticeable in the 'Bedroom' area, due to the large bed obstructing the majority of the room.

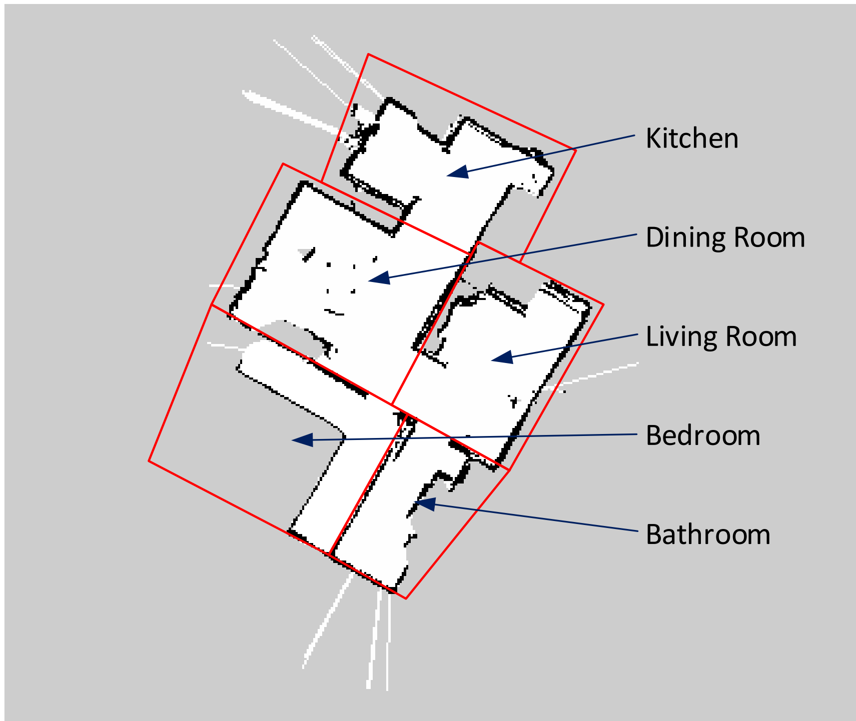

Figure 22: HomeLab Navigable Map (Partitioned)

Figure 22 also shows the partitioning scheme used to separate the environment into regions for the global and local search planners. An automated partitioning scheme was considered using a previously developed Voronoi [29] partitioning method, however it was found that the algorithm had difficulty segregating the 'Kitchen', 'Dining Room' and 'Living Room' areas, due to the lack of doorways that would typically create local minima. Instead, a manually partitioned map is used for the simulated and live segments of the system evaluation (Section 5.4).

5.2. Data Collection and Pre-processing

To collect schedule data for model training, fifteen participants from our research group were requested to complete a schedule template as shown in Appendix A. Participants were asked to log location and behaviour data throughout a typical day at home in 15 minute intervals, as well as their typical wake and sleep times.

The data was then normalized based on each person's typical wake and sleep times. Across the collected data, average wake time was 7:48 AM, sleep time was 11:17 PM, and average total time awake was 15.49 hours. Data was normalized to be portable across the group regardless of sleep patterns, as shown in Figure 23:

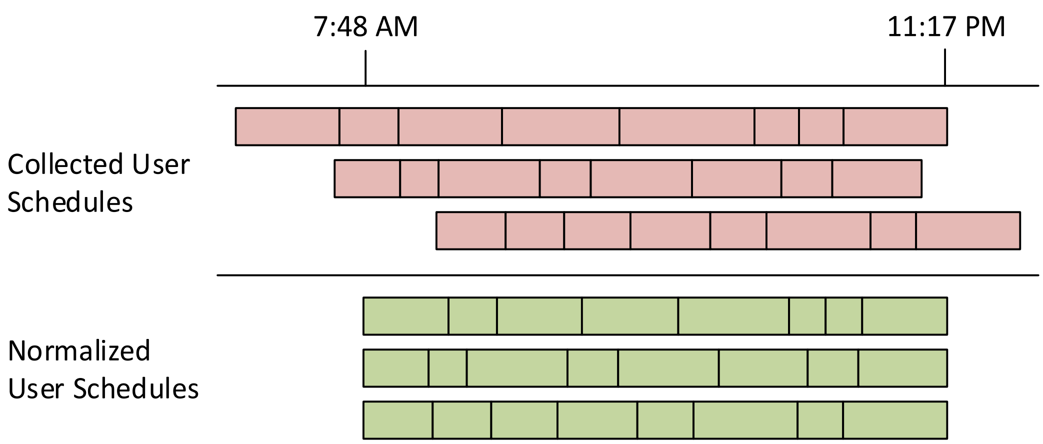

Figure 23: Data Normalization Diagram

Participants were requested to categorize their location into one of the five available rooms at the sample environment: 'Bedroom', 'Kitchen', 'Living Room', 'Dining Room', and 'Bathroom'. The reported behaviours tended to vary between each reported schedule, so common behaviours were categorized into six discrete states: 'Sleeping', 'Meal Preparation', 'Eating', 'Personal Hygiene', 'Passive Leisure' (Reading, Writing, Relaxing), and 'Active Leisure' (Exercise, Cleaning). This categorization was selected due to the general behaviour of interest being meal-related. An alternative activity selection may favour a categorization scheme with more diversity.

5.3. Model Training

The overall inference model is divided into sub-models, each associated with a particular time of day. Ideally such time-periods could be labeled as *Morning*, *Afternoon*, and *Evening*, however in practice a higher level of discretization is needed to achieve a higher predictive accuracy. The number of sub-models is a key tuning parameter of the model and is discussed in more detail below. Each sub-model has an intrinsic start and end time, and is trained on the accompanying interval of schedule data, generating distinct parameters (i.e. emission and transition matrices). Since fully-labeled behaviour and location data is available, the maximum-likelihood approached outlined in Section 3.1.2 was used for the training procedure.

5.3.1. Location Estimation Process

To estimate person location from the model, at least two pieces of information are required: (i) the time of day and (ii) the person's profile parameters, namely typical wake and sleep time. Using the information in the participant's profile, the time of day is normalized by applying the reverse transformation with respect to the data normalization process in Figure 23.

Subsequently, two approaches exist to estimate the user's location. At the start of the person search process, no evidence of past activity is available, and the naïve inference approach is taken to obtain an unbiased location estimate and initialize the global search priority. When evidence of past activity is discovered by the robot during the ongoing search process, a re-plan is triggered with the informed inference approach to obtain a location estimate and update the search priorities.

5.3.2. Model Parameter Tuning

The transition time step of the HMM model ($t_1 - t_0$) is fixed at 15 minutes, to mirror the resolution of the sample schedule data collected. For the location estimation discount factor used in Equation (23), 0.5 is applied. This results in state distributions further than 4 time steps (i.e. 1 hour) away having little (< 5%) effect on the overall location estimate.

The number of sub-models is the major tuning parameter for the overall inference model, and is determined using a leave-one-out cross-validation approach (LOOCV) [32] - the overall model is trained on all user schedules, less one holdout schedule. The model's ability to generalize the holdout schedule is measured by testing the model's ability to predict each time-location pair recorded in the holdout schedule. The model is successively allowed access to information regarding to none, one, and two past person locations prior to making an estimation. The HMM model was repeatedly trained with a varying sub-model count, and an optimal sub-model quantity of 9 was determined to maximize overall model accuracy.

5.3.3. Model Training Results

The finalized parameters were used to re-train the inference model with all user schedule data included; this model was subsequently used in simulated and robot trials. The net predictive accuracy of this model after LOOCV tuning is shown in Table 1.

Table 1: Training Benchmarks

		Search success with *x* rooms explored				
Search Prioritization		1	2	3	4	5
Random		20.0%	40.0%	60.0%	80.0%	100%
HMM Based	0 Prior Locations	53.0%	80.1%	91.9%	98.5%	100%
	1 Prior Location	61.9%	90.1%	96.6%	99.5%	100%
	2 Prior Locations	69.4%	90.9%	96.8%	99.7%	100%

The HMM-based inference method predicts user location significantly better than a random baseline approach - even with no evidence, the HMM prioritization method correctly estimates the user's true location on the first try 53% of the time. Providing the user's prior location as evidence to the planner allows the use of the informed inference approach, which results in a significant improvement in predictive accuracy. With one and two prior locations given, the HMM model determines the user's true location correctly 61% and 69% of the time, respectively. It is important to note that these results are overly optimistic, as in practice the planner only obtains prior location information after the robotic system has already explored a portion of the environment.

5.3.4. Model Introspection

The final overall model contains nine sub-models, each responsible for a discrete segment of time. Since the average awake time found in the training data was 15.49 hours, each model spans 1.72 hours, or 103.3 minutes. The transition matrix T describes the likelihood of a person transitioning behaviours every 15 minutes, while the emission matrix E describes the likelihood of a person being located in one of the discrete locations given a specific behaviour. Each sub-model also has an associated stationary distribution, which is used to establish a search priority for the naïve inference method.

Examining the HMM sub-model parameters provides introspection into the model's generalization of person behaviour and location. If the 4th sub-model, which is in effect between 12.56 PM and 2:41 PM, is selected, the following parameter matrices are found:

Table 2: 4th Sub-model Transition Matrix

	Sleep	**Mealprep**	**Eating**	**Hygiene**	**Passive**	**Active**
Sleep	0.81	0.00	0.00	0.09	0.09	0.00
Mealprep	0.00	0.53	0.47	0.00	0.00	0.00
Eating	0.00	0.00	0.62	0.09	0.28	0.00
Hygiene	0.25	0.13	0.00	0.37	0.25	0.00
Passive	0.02	0.09	0.00	0.05	0.84	0.00
Active	0.17	0.17	0.17	0.17	0.17	0.17
Stationary Distribution	0.15	0.12	0.15	0.08	0.49	0.00

Table 3: 4th Sub-model Emission Matrix

	Bedroom	**Kitchen**	**Dining R**	**Living R**	**Bathroom**
Sleep	1.00	0.00	0.00	0.00	0.00
Mealprep	0.00	1.00	0.00	0.00	0.00
Eating	0.00	0.00	0.25	0.75	0.00
Hygiene	0.00	0.10	0.00	0.00	0.90
Passive	0.49	0.00	0.10	0.41	0.00
Active	0.20	0.20	0.20	0.20	0.20
Naïve Prioritization	0.40	0.13	0.09	0.32	0.07

An important observation can be made about the *Active Leisure* behaviour. Due to the smoothing factor applied in Equation (7), a uniform transition distribution is calculated for the state, due to no person recording the behaviour for the sub-model training set. This is an artifact of the training approach - since no other behaviour state transitions into *Active Leisure* for this sub-model, the net effect is that the stationary distribution weight for this behaviour is 0. It is preferable to mark this behaviour as unused during any informed inference estimates, as the uniform emission distribution for *Active Leisure* will disproportionately bias the result of the forward-backward estimation. In fact, the *Active Leisure* state comes close to the weight of the *Eating* state when evidence is found in the *Dining Room* location.

Interestingly, the diagonals of the transition matrix represent the tendency of a person to remain in a specific state between transitions. A person will tend to stay in the *Sleep* state and the *Passive Leisure* statue for extended periods of time (> 15 minutes), but tends to finish *Personal Hygiene* and *Meal Preparation* quickly. Patterns in sequences of behaviour can then be recognized in the transition matrix based on high weights in off-diagonal elements. Some common patterns of behaviour are observed in the transition matrix (e.g. *Meal Preparation* to *Eating*), however the highest stationary ranking goes to the *Passive Leisure* behaviour. This is due to other behaviours commonly transitioning into it, and the tendency of a person to stay in the *Passive Leisure* behaviour for extended periods of time. Since the most commonly emitted locations from the *Passive Leisure* behaviour are *Bedroom* and *Living Room*, those locations accordingly receive the highest prioritization in naïve search.

Table 3 and Table 4 show parameters extracted from the sub-model responsible for the 11:14 AM to 12.56 PM time period:

Table 4: 3rd Sub-model Transition Matrix

	Sleep	**Mealprep**	**Eating**	**Hygiene**	**Passive**	**Active**
Sleep	0.01	0.01	0.01	0.01	0.01	0.95
Mealprep	0.00	0.90	0.10	0.00	0.00	0.00
Eating	0.00	0.00	0.98	0.00	0.00	0.00
Hygiene	0.00	0.05	0.00	0.86	0.10	0.00
Passive	0.00	0.04	0.00	0.02	0.92	0.01
Active	0.00	0.13	0.00	0.00	0.13	0.75
Stationary Distribution	0.00	0.15	0.62	0.04	0.15	0.03

Table 5: 3rd Sub-model Emission Matrix

	Bedroom	Kitchen	Dining R	Living R	Bathroom
Sleep	0.96	0.01	0.01	0.01	0.01
Mealprep	0.00	0.86	0.13	0.00	0.00
Eating	0.00	0.00	0.99	0.00	0.00
Hygiene	0.00	0.00	0.00	0.00	1.00
Passive	0.06	0.00	0.00	0.94	0.00
Active	0.00	0.00	0.00	1.00	0.00
Naïve Priority	0.01	0.13	0.64	0.17	0.05

Of interest for this sub-model is the heavy stationary weight on the *Eating* behaviour and the associated *Dining Room* location, due to the traditional 'lunch time' being captured in this time interval. Another artifact of the HMM training approach is evident in the *Sleep* behaviour for this time period. No other behaviour transitions into *Sleep* due to one person reporting sleeping during this time interval; this person's behaviour immediately transitioned into *Active Leisure*. Overall, the relatively small quantity of training data is reflected in the occurrence of smoothing artifacts. Increasing the size of the training dataset several orders of magnitude would be helpful in improving the consistency of the model.

5.4. System Evaluation

To evaluate the trained model in a non-deterministic setting, and also evaluate the robotic system, the person search approach was tested in both simulated and real environments. In both environments, the evaluation is broken up into individual trials, with no experiment state preserved between subsequent trials.

5.4.1. Experiment Set-up

The initial conditions of each trial are configured by picking a random person identity and time of day. The person identity provides a corresponding schedule, wake time, and sleep time. In continuing with the LOOCV approach, a model trained on schedule data excluding that person's information is used when estimating a person's location in each trial. This is because the trials themselves are configured based on the person's schedule data, so not applying LOOCV would result in circular reasoning and accepting a false hypothesis. Prior to the trial commencing, the person is placed into a region in the environment in accordance with the person's schedule profile. The person's exact position in the region is randomized, however it is restricted to being in an area navigable by the robot. Depending on the scenario, evidence markers (e.g. dirty dishes) are placed in the environment, to indicate the person's prior locations. For simulated experiments, these markers are in the form of meta-data in the simulated environment, while for robot experiments these markers take the form of QR codes. A map of the environment, and the boundaries for the associated regions, is also provided to the robot (Figure 22).

When the trial begins, the robot's person search planner is sent a request to search for the target person. The global search component generates a naïve search prioritization based on the person's identity to initiate the search. The local search component then begins searching the region with the highest priority. During the search process, if evidence of past activity is located by the robotic system, the global planner regenerates the search prioritization using the informed approach. This only affects the search priority of unexplored regions – previously explored regions are not re-explored even if their priority is increased. The trial continues until either the target person is located using the robot's perception system, indicating success, or the robot has explored all regions in the environment without locating the target person, indicating failure of the trial.

5.4.2. Simulated Trials

For the simulated trials, the physical robot systems are emulated using the simulator outlined in Section 4.3. This approach enables evaluating the person-search planning component with a large variety of scenarios due to the high feasible number of iterations - the simulator runs faster than real-time at a 20:1 ratio. For an initial trial set of experiments, 500 trials were run for each of the below scenarios:

i. Random: the global search planner is implemented with a random prioritization sequence generator. This serves as a baseline for comparison of the HMM prioritization.
ii. HMM 0: The HMM inference model is used to generate the search prioritization. No evidence is placed in the environment, so the model is always queried in naïve inference mode.

iii. HMM 1: The target person's immediate prior location has an evidence marker placed in the environment. The marker is placed in a random location within the region of interest, and must be observed by the robot before the information is used for inference.

iv. HMM 2: The target person's two immediate prior locations have evidence markers placed in the environment.

Collected results for this testing round are displayed in Table 6:

Table 6: First Round Simulation Results

Scenario	**Success after *x* rooms searched**					**Total Trials**	**Average Time (s)**	**Evidence per Trial**
	1	2	3	4	5			
Random	28.24%	45.15%	63.24%	79.12%	92.50%	500	225.98	-
HMM 0	54.16%	75.18%	83.21%	88.76%	92.12%	500	124.93	-
HMM 1	56.70%	80.47%	85.25%	90.89%	94.75%	500	103.97	0.63
HMM 2	55.67%	80.97%	87.66%	92.75%	94.48%	500	89.49	0.88

From Table 6, the final success rate of all search scenarios is an average of 93.4%, which indicates that regardless of the global planner's prioritization performance, the local search approach is highly effective at locating the target person in a single cycle of the person search.

The baseline Random global search planner has a higher performance than predicted during training. This is due to the person often being located near the border of one region, and being found by the robot while it is searching another region. When evaluating the HMM-based global search planner, the majority of the performance gains over the baseline planner are captured with the naïve inference approach – a 44.7% decrease in search time. This indicates that based on the collected schedule data, person location is predictable and generalizable based on the time of day only.

The HMM 1 and HMM 2 scenarios display performance improvement when compared to the HMM 0 scenario, due to their use of the informed inference approach. The presence of 1 and 2 evidence markers in the environment reduce overall search time 16.8% and 28.4% respectively over the naïve HMM planner, and 53.9% and 60.4% respectively over the randomized planner. Evidence markers can only affect the search prioritization after they are discovered, hence, the addition of markers impacts the success rate of person-search only in iterations where the robot does not immediately locate the person in the first and/or second search regions. The performance gains of the HMM planner in the 1 and 2 evidence scenarios are visible in the 'Success after 1 room searched' and 'Success after 2 rooms searched' columns of Table 6, respectively.

To simulate the effect of reduced performance in the perception system, an additional cycle of simulations was performed with an artificially reduced evidence detection rate. Prior to commencing an additional round of testing on the HMM 1 and HMM 2 scenarios, the simulated perception system was set to only report evidence detection 80% of the time. While this reduction does affect the overall search success rate, it creates a linear reduction in the amount of evidence detected during each trial, as well as a decrease in the early-success rate of the HMM-based search planner in the HMM 1 / 2 scenarios. The results for these scenarios are shown in Table 7, alongside the original Random and HMM 0 scenario results for comparison.

Table 7: Second Round Simulation Results (80% Evidence Detection)

Scenario	**Success after *x* rooms searched**					**Total Trials**	**Average Time (s)**	**Evidence per Trial**
	1	2	3	4	5			
Random	28.24%	45.15%	63.24%	79.12%	92.50%	500	225.98	-
HMM 0	54.16%	75.18%	83.21%	88.76%	92.12%	500	124.93	-
HMM 1	57.56%	79.04%	83.63%	90.94%	93.46%	500	109.69	0.51
HMM 2	55.35%	80.22%	86.81%	91.60%	94.65%	500	95.20	0.69

The reduced detection rate generated the expected effect of a linear reduction of average evidence detected per trial, as well as a minor increase in the average search time (5.21% and 6.3% for HMM 1 and 2 respectively). As expected, the overall success rate of the search scenarios is affected.

5.4.3. Robot Trials

After the initial system validation through simulation, further person-search experiments were performed at the TRI Homelab using the complete robotic system. Since both the naïve and informed inference approaches were validated in simulation, both methods are active from the outset in robot trials, eliminating the need for HMM 0 / 1 / 2 scenarios. This guides the following scenario configurations:

i. Random - the global search planner is implemented with a random prioritization sequence generator. This serves as a baseline for comparison of the HMM prioritization.
ii. HMM – The HMM inference model is used to generate the search prioritization. The person's two immediate prior locations have evidence markers placed in the environment. The markers are placed in random locations within the region of interest, and must be observed by the robot before being used for inference.

Since the robot trials are limited to operating in real-time (vs. compressed time in simulations), 50 trials of each scenario configuration were performed. The results of these scenarios are shown in Table 8:

Table 8: Robot Trials Results

Scenario	**Success after *x* rooms searched**					**Total Trials**	**Average Time (s)**	**Evidence per Trial**
	1	2	3	4	5			
Random	24.0%	44.0%	58.0%	70.0%	88.0%	50	264.03	-
HMM	46.0%	76.0%	84.0%	90.0%	92.0%	50	141.72	0.73

From Table 8, the robot trial results overall validate the design of the HMM planner, as well as the overall robotic person-search system. The final success rate of all methods average 90% success rate overall (Success after 5 rooms searched in Table 8), indicating that the local person search planner is effective at locating the target person in a single search cycle, regardless of the performance of the global search planner. The HMM-based global search planner performs 46.5% faster (Average Time in Table 8) than the random search baseline when using the physical robotic system. Directly evaluating the performance of the person- and evidence-detection components is made difficult due to the complex factors involved in the vision model, namely limited field of view and significance of the angle of robot approach. When compared to a linear estimation of the performance profile in Table 6 (100% success rate - 0.88 evidence markers detected per trial) and Table 7 (80% success rate - 0.69 evidence markers detected per trial), the implemented marker detection system to detects markers 84.2% (0.73 evidence markers per trial) of the time the robot enters a region.

5.5. Chapter Summary

This chapter presents an evaluation of the overall person-search approach. First, the method for collecting and processing labeled schedule data is outlined. Subsequently, the global planner's backing statistical model is tuned, trained and examined for patterns in user behaviour and location. Finally, the overall approach is validated in simulated trials, and live robot trials.

Chapter 6
Information Gathering Experiment

To determine whether the interaction system implementation and the user interface design would promote the use of the robotic system as a meal preparation guide, a series of activity sessions were performed with 15 participants drawn from our research group. In each session, the robot guided a participant through a meal preparation exercise, while providing guidance through the preparation steps using the available interaction capabilities. The participants then completed a questionnaire designed to evaluate multiple criteria evaluating the system's acceptability as an assistive agent.

The baseline technique for evaluating acceptance of a technological device is the technology acceptance model (TAM) [53], which defines several important constructs that serve as criteria in the system acceptance evaluation:

i. Perceived Ease of Use (PEOU), defined as "the degree to which a person believes that using a particular system would be free of effort".
ii. Perceived Usefulness (PU), defined as “the degree to which a person believes that using a particular system would enhance his or her job performance"

An important addendum to the original TAM model is proposed in [54] to be

iii. Perceived Enjoyment (PE), defined as “the extent to which the activity of using the computer is perceived to be enjoyable in its own right, apart from any performance consequences that may be anticipated".

Davis posits that the above factors contribute to the overall technology acceptance, measured as the Behavioural Intent to Use (BIU) [53]. A further study by Heijden [55] classifies individual systems as being either utilitarian or hedonic, and proposes that for hedonic systems, the PE construct is a “stronger predictor of behavioural intent to use then perceived usefulness”. The author distinguishes hedonic systems as those that “provide self-fulfilling value to the user”, while utilitarian systems are evaluated purely on the instrumental value they provide to the user. Heijden surveyed the users of a widely used film recommendation website, and found that the purpose of a system is pivotal in the distribution of which factors influence the acceptance of a system. The findings illustrate that for a hedonic system, PE and PEU outweigh PU as contributing factors to BIU (0.32 and 0.25, versus 0.15 respectively). Of interest in the experiment is the overall acceptance measure, as well contribution of the individual criteria to the overall BIU construct, which can help determine whether the participants view the system as primarily utilitarian or hedonic.

6.1. Questionnaire Design

The guided task questionnaire administered to the participants was designed based on the survey instrument used in the Hedonic Information Systems study [55]. The core questions are based on a modified TAM approach:

i. The PU, PEOU, and BIU constructs are each evaluated using a set of three correlated questions, with the participant marking a response along a 5-point Likert scale (In order from lowest to highest: Strongly Disagree, Slightly Disagree, Neutral, Slightly Agree, Strongly Agree).
ii. The PE construct is evaluated using a set of four prompts consisting of negative-positive emotional response pairs, with the participant responding on a 7-point Semantic scale representing the magnitude of his/her response.

Additionally, the questionnaire contains:

iii. General self-measure of the participant’s experience with computers and robotics, with each technology category presenting a choice ranging between *No Experience*, *Beginner*, *Intermediate*, and *Advanced*.
iv. Multiple-selection questions of which features of the robot the participants found most helpful and which characteristics most enjoyable.
v. Prompts for further long-form response by the participants, on the subject of enjoyed/disliked aspects of the design, and potential improvements/features.

6.2. Experiment Set-up and Procedure

The meal preparation exercise is designed to be representative of a typical meal preparation activity undertaken by a user of the complete proposed system. The experiments are set inside of a basement office repurposed as a mock kitchen, with the desk and cupboards representing countertop and cabinets (Figure 24 and Figure 25).

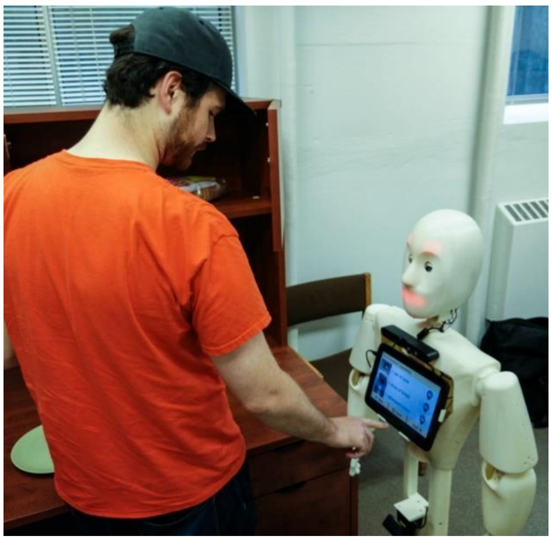

Figure 24: Casper displaying list of ingredients

Figure 25: Locating ingredients with Casper's assistance

Prior to each experiment, ingredients required for the meal preparation were stowed inside each cabinet. Throughout the interaction, the Casper robot is remotely controlled by an operator to advance through each interaction segment. The segments of the interaction are outlined below:

i. The participant is asked by the operator to wait outside the kitchen doorway.
ii. Casper drives through the doorway, and requests the participant to follow it into the room.
iii. Casper leads the robot into the room.
iv. Casper plays a short introduction speech introducing the activity, and prompts the participant to select the 'Tuna Sandwich' recipe from its tablet interface.
v. The participant proceeds through the finite state machine outlined in Section 4.3.5, using the tablet interface (Figure 24).

vi. If the participant requests help finding an ingredient for the meal preparation, Casper points to the appropriate cupboard where the ingredient is located (Figure 25**)**.

vii. Once the terminal state of the state machine is reached, Casper informs the participant that the meal preparation activity is complete, and thanks him/her for his/her time.

6.3. Results

Participant responses to the guided task questionnaire are shown in Tables 9 and 10:

:

Table 9: General Questionnaire Results

Participant	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**Mean**	**Var**
Experience with Computers	4	4	4	4	4	4	3	4	4	4	2	4	4	4	4	3.80	0.31
Experience with Robots	2	3	4	4	3	4	2	4	3	3	2	4	4	2	4	3.20	0.74
Helpful Features																	
Written instructions for each recipe step displayed on the tablet	0	1	1	1	1	1	0	0	1	1	0	1	1	1	1	0.73	0.21
Video recording of each recipe step displayed on the tablet	0	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0.80	0.17
Gestural prompts (robot pointing to ingredients and tools in the environment)	1	0	1	0	1	0	0	1	0	1	0	0	0	1	1	0.47	0.27
Ability of the robot to guide me to the kitchen area	0	0	1	0	1	1	0	0	1	1	1	1	1	1	1	0.67	0.24
Liked about the robot design																	
The robot’s overall physical appearance	0	0	1	0	1	1	0	1	1	1	0	1	1	0	1	0.60	0.26
The robot’s clear voice	0	1	1	1	1	0	1	1	0	1	1	1	0	0	1	0.67	0.24
The touchscreen user interface on the robot	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1	0.80	0.17
The robot expressing different emotions through facial expressions	0	0	1	0	1	0	0	1	1	0	1	0	0	0	1	0.40	0.26

Table 10: TAM Questionnaire Results

Participant	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Mean	Var
Perceived Usefulness (Likert 5)																	
Using the robot helps me prepare the recipe faster	5	3	1	4	5	4	4	4	4	4	5	5	4	4	3	3.93	1.07
I remember the meal preparation steps more precisely with the robot's help	5	5	3	5	5	5	5	5	5	3	5	5	4	3	4	4.47	0.70
The robot helps me complete the recipe from start to finish	5	5	4	4	5	5	5	5	5	4	5	3	5	5	4	4.60	0.40
Perceived Ease of Use (Likert 5)																	
It was easy to interact with the robot during the meal preparation activity.	5	5	5	4	5	5	5	4	4	5	2	5	5	5	3	4.47	0.84
The robot's instructions were clear and easy to understand	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	4.93	0.07
The robot's instructions were logical and easy to perform	4	5	4	5	5	5	5	4	5	5	4	5	5	5	5	4.73	0.21
Perceived Enjoyment (Semantic 7)																	
Disgusting (0) - Enjoyable (7)	6	5	6	6	7	6	6	6	6	5	5	5	6	7	6	5.77	0.42
Unpleasant (0) - Pleasant (7)	6	6	6	6	7	6	6	6	5	6	5	6	6	7	6	5.90	0.33
Boring (0) - Interesting (7)	7	7	4	6	7	7	5	5	6	6	6	7	5	7	6	5.97	0.98
Dull (0) - Exciting (7)	6	6	2	6	4	6	4	5	6	4	5	7	4	7	6	5.13	1.80
Behavioural Intent to Use (Likert 5)																	
I intend to use the robot as a meal preparation guide again	4	4	1	2	5	5	4	4	3	3	5	3	4	4	3	3.60	1.26
I would like to use the robot for different recipes	5	5	3	3	4	5	5	5	4	4	5	5	5	5	4	4.47	0.55
I would use the robot again	4	5	3	4	5	5	5	5	4	4	5	5	5	4	4	4.47	0.41

6.4. Analysis

From the multiple-selection questions, the prevalence among participants who found specific robot features helpful (Table 11) and characteristics enjoyable (Table 12) was determined:

Table 11: Summary of the Robot's Helpful Features

Robot Feature	Ranking	Prevalence Among Participants
Video	1st	80.00%
Written	2nd	73.33%
Guidance	3rd	66.67%
Gestural	4th	46.67%

Table 12: Summary of Liked Aspects in the Robot Design

Design Element	Ranking	Prevalence Among Participants
Interface	1st	80.00%
Voice	2nd	66.67%
Appearance	3rd	60.00%
Emotions	4th	40.00%

The descriptive statistics for the TAM constructs are summarized in Table 13, including a variance measure, and a Cronbach's alpha measure (α) [56]. The Cronbach's alpha is an indicator of internal consistency, measuring the extent to which a group of questions and their responses represent a single construct. From the results in Table 13, each individual construct demonstrated an acceptable level of internal consistency ($\alpha \geq 0.70$ [57]), however the high alpha measure ($\alpha \geq 0.95$ [57]) for the BIU construct indicates that the questions asked for that construct used may be overly similar.

Table 13: Descriptive Statistics for TAM Constructs

Construct	Min	Max	Mean	Std. Dev.	Alpha
Perceived Usefulness	1	5	4.33	0.60	0.87
Perceived Ease of Use	1	5	4.71	0.15	0.77
Perceived Enjoyment	1	7	5.69	0.90	0.83
Behavioural Intent to Use	1	5	4.18	0.77	0.96

Due to the high consistency results amongst the TAM constructs using the Cronbach's alpha measure, all constructs are included in the subsequent analyses. First, the correlation measure between the four TAM constructs was examined [55]; the summary of the correlation results are shown in Table 14.

Table 14: Correlation Coefficients

Construct	PU	PEOU	PE	BIU
PU	1			
PEOU	-0.060	1		
PE	0.321	0.181	1	
BIU	0.757	0.048	0.331	1

From this result, a strong positive correlation ($x > 0.5$) was found between only the PU and BIU constructs. The summary of the follow-up regression analysis is shown in Table 15.

Table 15: Regression Analysis

Construct	Coefficients	Standard Error	t Stat	P-value
PU	0.895	0.250	3.577	0.004
PEOU	0.145	0.365	0.396	0.700
PE	0.072	0.190	0.382	0.710

A high contribution to BIU for the PU construct ($p < 0.01$) is found, but high p-values ($p > 0.7$) for PEOU and PE prevent statistical comparison. The participant's long-form responses were grouped into recurring sentiments, summarized in Table 16.

Table 16: Long-form Responses

Question	Response	Prevalence Among Participants
What would you like to see improved on the robot?	Robot too still/silent during tablet segment	20.0%
	Voice interactivity during tablet segment	73.3%
Do you think this robot could be useful to care for elderly/cognitively impaired individuals?	Yes	93.3%

6.5. Discussion

Although the constructs and associated questions demonstrated high internal-consistency, the correlation and regression analyses failed to demonstrate results of statistical significance for the PE and PEOU constructs (low correlation in Table 14, $p > 0.7$ in Table 15), precluding further deliberation about the utility and acceptance of the system as in [55]. The limitations of this experiment can be extracted from responses to the long-form component of the questionnaire (Table 16). Several participants (20.0%) found that the interactivity of the system was limited during the tablet interaction component. One participant remarked that the robot served primarily as a "tablet carrier". Subsequent improvements to the level of integration between the robotic system and the tablet interface would address this sentiment. One such improvement could entail replacing the simple finite state machine used during the interaction with a more complex control architecture that incorporates user state in terms of emotion or affect [58][59]. A high proportion (93.3%) of participants commented that the robot would be useful in caring for elderly and cognitively impaired individuals, however the lack of statistical congruency in the TAM results suggests that the participant group was sufficiently disjoint from the target user demographic that they did not find the system useful or the meal preparation activity enjoyable. A further study drawing on participants sampled from the target population segment must be done to obtain statistically consistent results.

Chapter 7
Conclusion

7.1. Summary of Contributions

The contributions of this thesis are:

7.1.1. Person Search Approach

A two-tiered approach for person search has been developed, comprising of global and local person search planners. The global planner generates a search prioritization of the environment using a by estimating user location from a probabilistic model. The local planner controls the movement of the robotic system through the environment, using sensory information to locate the user. The novelty of the approach lies in the global planner's model, which incorporates both behavioural and temporal context. The model contains multiple discrete Hidden Markov Models (HMM) as sub-models, each associated with a time of day, and trained on sequences of user behaviour and location. Additionally, throughout the search process evidence of past user activity is detected by the robot, and used by the global search planner to generate an improved search prioritization.

7.1.2. Assistive Robotic System

A robotic system (Casper) has been prototyped to integrate the user-search approach and an assistive activity, a guided meal preparation. Primary contributions were the overall design, prototyping, and integration of the robotic system with both off-the-shelf components, and peripheral systems developed in cooperation with other students and researchers.

7.1.3. Experimental Results

The global person-search planner reduced overall search time by 46%, when compared to the baseline random planner. The local person-search planner located the target person 90% of the time. Using an information gathering experiment, the assistive robotic system was evaluated using a modified Technology Acceptance Model (TAM). Participant feedback was positive, with many users reporting a positive reaction to Casper's aesthetic features and assistive function.

While the TAM constructs demonstrated high internal consistency, correlation and regression analyses of the results proved inconclusive. Feedback was collected from participants regarding potential improvements to the system.

7.2. Discussion of Future Work

The prototype robotic system provides a high degree of modularity in its components; avenues of future work for each component are outlined below:

7.2.1. User Search Approach

Both the global and local planner modules warrant additional improvement and tuning, the results of which can be compared against the benchmarks established in this work. In general, the user search approach as a whole would benefit from introducing travel cost into the search region prioritization process, as there is currently no cost considered in the decision to travel from one distant region to another, rather than exploring the proximal area first.

7.2.2. Global Search Planner

The global planner stands to gain from an online training approach such as the Baum-Welch algorithm, which would ensure that the model improves through continued use. A re-examination of the backing model for the global planner may be performed, in favour of more complex approaches. The discrete HMM sub-models of the current implementation could be replaced by a time-varying parameter approach [60], or with a more inter-connected model such as the Dynamic Bayesian Network approach proposed by Lee [22]..

7.2.3. Local Search Planner

The local Frontier-based search approach, would benefit from an internal model of the sensor field-of-view, which can be extracted with an extension of the backing grid model to coverage maps [25].

7.2.4. Assistive Interaction

Due to the lack of consistency in the results obtained during the information gathering experiment, there is no statistical grounds for evaluating the assistive interaction design as a whole. Certain iterative improvements can be made to the robot's aesthetic features based on participant comments, and improvements in the integration of the user interface (tablet) and the rest of the system would enable experiments to be more convincing for participants, improving the statistical congruence of future studies. Following this, a user study with a sample drawn from the target elderly population would provide further validation of the design.

7.3. Final Concluding Statement

The assistive robotic system proposed in this work was successfully designed and prototyped, and provided positive first-pass results on both the user-search and user interaction components of the system. We hope that the development of the statistical person-search approach drives natural operation of assistive robotic systems in household environments. The modular design of the implemented system allows iterative improvement of each component, as well as the application of the components to other robotic systems of similar configuration in the ROS ecosystem.

References

[1] Canadian Institute for Health Information, "Health Care in Canada,2012: A Focus on Wait Times." Canadian Institute for Health Information (CIHI), 01-Dec-2012.

[2] S. Zimmerman, P. D. Sloane, C. S. Williams, P. S. Reed, J. S. Preisser, J. K. Eckert, M. Boustani, and D. Dobbs, "Dementia Care and Quality of Life in Assisted Living and Nursing Homes," *Gerontologist*, vol. 45, no. Supplement 1, pp. 133–146, Oct. 2005.

[3] D. M. F. Mahoney, P. H. Mutschler, B. Tarlow, and E. Liss, "Real world implementation lessons and outcomes from the Worker Interactive Networking (WIN) project: workplace-based online caregiver support and remote monitoring of elders at home.," *Telemed. J. E. Health.*, vol. 14, no. 3, pp. 224–234, 2008.

[4] J. E. Graham, K. Rockwood, B. L. Beattie, R. Eastwood, S. Gauthier, H. Tuokko, and I. McDowell, "Prevalence and severity of cognitive impairment with and without dementia in an elderly population.," *Lancet*, vol. 349, no. 9068, pp. 1793–1796, Jun. 1997.

[5] T. K. Tatemichi, D. W. Desmond, Y. Stern, M. Paik, M. Sano, and E. Bagiella, "Cognitive impairment after stroke: frequency, patterns, and relationship to functional abilities.," *J. Neurol. Neurosurg. Psychiatry*, vol. 57, no. 2, pp. 202–207, Feb. 1994.

[6] S. M. Albert, J. Bear-Lehman, and A. Burkhardt, "Lifestyle-adjusted function: Variation beyond BADL and IADL competencies," *Gerontologist*, vol. 49, no. 6, pp. 767–777, Dec. 2009.

[7] G. A. Marshall, D. M. Rentz, M. T. Frey, J. J. Locascio, K. a. Johnson, and R. a. Sperling, "Executive function and instrumental activities of daily living in mild cognitive impairment and Alzheimer's disease," *Alzheimer's Dement.*, vol. 7, no. 3, pp. 300–308, May 2011.

[8] H. H. Dodge, T. Kadowaki, T. Hayakawa, M. Yamakawa, A. Sekikawa, and H. Ueshima, "Cognitive impairment as a strong predictor of incident disability in specific ADL-IADL tasks among community-dwelling elders: the Azuchi Study.," *Gerontologist*, vol. 45, no. 2, pp. 222–230, Apr. 2005.

[9] E. Dishman, "Inventing wellness systems for aging in place," *Computer (Long. Beach. Calif).*, vol. 37, no. 5, pp. 34–41, May 2004.

[10] M. Hickson, "Malnutrition and ageing.," *Postgrad. Med. J.*, vol. 82, no. 963, pp. 2–8, Jan. 2006.

[11] K. Farid, Y. Zhang, D. Bachelier, P. Gilson, A. Teixeira, M. E. Safar, and J. Blacher, "Cognitive impairment and malnutrition, predictors of all-cause mortality in hospitalized

elderly subjects with cardiovascular disease," *Arch. Cardiovasc. Dis.*, vol. 106, no. 4, pp. 188–195, Apr. 2013.

[12] M. C. Mozer, "The neural network house: An environment that adapts to its inhabitants," *Am. Assoc. Artif. Intell. Spring Symp. Intell. Environ.*, 1998.

[13] S. Akoush and A. Sameh, "Mobile user movement prediction using bayesian learning for neural networks," *ACM Organ. IWCMC '07*, p. 191, 2007.

[14] S. Oh, "Using an adaptive search tree to predict user location," *J. Inf. Process. Syst.*, vol. 8, no. 3, pp. 437–444, Sep. 2012.

[15] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, "Show me how you move and I will tell you who you are," *Proc. 3rd ACM SIGSPATIAL Int. Work. Secur. Priv. GIS LBS*, vol. 4, no. 2, pp. 34–41, Aug. 2010.

[16] S. Gambs, M.-O. Killijian, and M. N. Del Prado Cortez, "Next place prediction using mobility Markov chains," in *Proceedings of the First Workshop on Measurement Privacy and Mobility MPM 2012*, 2012, pp. 1–6.

[17] M. Luber, G. D. Tipaldi, and K. O. Arras, "Place-dependent people tracking," *Springer Tracts Adv. Robot.*, vol. 70, no. STAR, pp. 557–572, Jan. 2011.

[18] G. D. Tipaldi and K. O. Arras, "I want my coffee hot! Learning to find people under spatio-temporal constraints," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2011, pp. 1217–1222.

[19] A. T. Ihler and P. Smyth, "Learning time-intensity profiles of human activity using non-parametric Bayesian models," in *Neural Information Processing Systems 19*, 2007, pp. 625–632.

[20] A. Asahara, K. Maruyama, A. Sato, and K. Seto, "Pedestrian-movement prediction based on mixed Markov-chain model," in *GIS '11 Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2011, pp. 25–33.

[21] W. Mathew, R. Raposo, and B. Martins, "Predicting future locations with hidden Markov models," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing - UbiComp '12*, 2012, pp. 256–269.

[22] S. Lee, K. C. Lee, and H. Cho, *A Dynamic Bayesian Network Approach to Location Prediction in Ubiquitous Computing Environments*, vol. 114. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[23] G. Singla, D. J. Cook, and M. Schmitter-Edgecombe, “Recognizing independent and joint activities among multiple residents in smart environments,” *J. Ambient Intell. Humaniz. Comput.*, vol. 1, no. 1, pp. 57–63, Dec. 2010.

[24] B. Yamauchi, “A frontier-based approach for autonomous exploration,” in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA’97. “Towards New Computational Principles for Robotics and Automation,”* 1997, pp. 146–151.

[25] C. Stachniss and W. Burgard, “Exploring unknown environments with mobile robots using coverage maps,” *IJCAI Int. Jt. Conf. Artif. Intell.*, pp. 1127–1132, 2003.

[26] C. Stachniss, G. Grisetti, and W. Burgard, “Information Gain-based Exploration Using Rao-Blackwellized Particle Filters.,” *Robot. Sci. Syst.*, pp. 65–72, 2005.

[27] H. J. S. Feder, “Adaptive Mobile Robot Navigation and Mapping,” *Int. J. Rob. Res.*, vol. 18, no. 7, pp. 650–668, Jul. 1999.

[28] F. Bourgault, a. a. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, “Information based adaptive robotic exploration,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002, vol. 1, pp. 540–545.

[29] D. Holz, N. Basilico, F. Amigoni, and S. Behnke, “Evaluating the efficiency of frontier-based exploration strategies,” *Isr/Robotik 2010*, 2010. [Online]. Available: http://www.vde-verlag.de/proceedings-en/453273006.html. [Accessed: 15-Mar-2015].

[30] J. Petzold, F. Bagci, W. Trumler, and T. Ungerer, “Next Location Prediction Within a Smart Office Building,” *Cogn. Sci. Res. Pap. Unversity Sussex CSRP*, vol. 577, p. 69, 2005.

[31] L. R. Rabiner and and B. H. Juang, “An introduction to hidden Markov models,” *IEEE ASSP Mag.*, pp. 257–286, 1986.

[32] C. Bishop and N. Nasrabadi, “Pattern recognition and machine learning,” *Pattern Recognition*, 2006. [Online]. Available: http://www.library.wisc.edu/selectedtocs/bg0137.pdf. [Accessed: 22-Mar-2015].

[33] C. Zhai and J. Lafferty, “A study of smoothing methods for language models applied to Ad Hoc information retrieval,” in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR ’01*, 2001, pp. 334–342.

[34] M. Quigley, K. Conley, B. Gerkey, J. FAust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Mg, “ROS: an open-source Robot Operating System,” *Icra*. 2009.

[35] Crosswing Inc., “Crosswing Inc.” Toronto, ON, 2014.

[36] Y. Zeng, “Arm Design for an Assistive Robot,” University of Toronto, 2013.

[37] Y. Li, “Head and Hand Design for an Assistive Robot,” University of Toronto, 2013.

[38] Hokuyo Automatic Co. Ltd., “URG-04LX-UG01,” 2009. [Online]. Available: https://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx_ug01.html. [Accessed: 24-Mar-2015].

[39] ASUSTeK Computer Inc., “Xtion PRO Specifications,” 2012. [Online]. Available: http://www.asus.com/Multimedia/Xtion_PRO/specifications/. [Accessed: 24-Mar-2015].

[40] Open Perception, “Point Cloud Library (PCL): Module filters,” 2015. [Online]. Available: http://docs.pointclouds.org/trunk/group__filters.html. [Accessed: 24-Mar-2015].

[41] C. Pannu, “Assisting the elderly in meal preparation,” University of Toronto, 2013.

[42] G. Grisetti, C. Stachniss, and W. Burgard, “Improved techniques for grid mapping with Rao-Blackwellized particle filters,” *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 34–46, Feb. 2007.

[43] V. Rabaud, “ROS Gmapping Library,” 2009. [Online]. Available: http://wiki.ros.org/gmapping. [Accessed: 24-Mar-2015].

[44] D. Fox, “KLD-Sampling : Adaptive Particle Filters and Mobile Robot Localization,” pp. 1–12, 2001.

[45] B. Gerkey, “ROS AMCL Library,” 2009. [Online]. Available: http://wiki.ros.org/amcl. [Accessed: 24-Mar-2015].

[46] E. Marder-Eppstein and E. Perko, “ROS Navigation Library,” 2009. [Online]. Available: http://wiki.ros.org/navigation. [Accessed: 24-Mar-2015].

[47] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 1997.

[48] M. Munaro and E. Menegatti, “Fast RGB-D people tracking for service robots,” *Auton. Robots*, vol. 37, no. 3, pp. 227–242, Feb. 2014.

[49] Intel Corporation, “Intel Core i7-3517UE Processor,” 2012. [Online]. Available: http://ark.intel.com/products/65713/Intel-Core-i7-3517UE-Processor-4M-Cache-up-to-2_80-GHz. [Accessed: 24-Mar-2015].

[50] É. Marchand, F. Spindler, and F. Chaumette, “ViSP for visual servoing: A generic software platform with a wide class of robot control skills,” *IEEE Robot. Autom. Mag.*, vol. 12, no. 4, pp. 40–52, Dec. 2005.

[51] M. Schwenk, T. Vaquero, and G. Nejat, “Schedule-Based Robotic Search for Multiple Residents in a Retirement Home Environment,” *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/viewFile/8254/8651. [Accessed: 24-Mar-2015].

[52] D. N. Philip and M. Aziz, “Science Outreach through the iDAPT Centre for Rehabilitation Research,” *Crucible*, vol. 44, no. 1, Sep. 2012.

[53] F. D. Davis, “Perceived Ease of Use, and User Acceptance of Information Technology,” *MIS Q.*, vol. 13, no. 3, pp. 319–340, Sep. 1989.

[54] F. D. Davis, R. P. Bagozzi, and P. R. Warshaw, “Extrinsic and Intrinsic Motivation to Use Computers in the Workplace1,” *J. Appl. Soc. Psychol.*, vol. 22, no. 14, pp. 1111–1132, 1992.

[55] H. Van Der Heijden, “User acceptance of hedonic information systems^,” *MIS Q.*, vol. 28, no. 4, pp. 695–704, Dec. 2004.

[56] L. J. Cronbach, “Coefficient alpha and the internal structure of tests,” *Psychometrika*, vol. 16, no. 3, pp. 297–334, Sep. 1951.

[57] M. Tavakol and R. Dennick, “Making sense of Cronbach’s alpha,” *Int. J. Med. Educ.*, vol. 2, pp. 53–55, Jun. 2011.

[58] J. Chan and G. Nejat, “Social intelligence for a robot engaging people in cognitive training activities,” *Int. J. Adv. Robot. Syst.*, vol. 9, 2012.

[59] D. McColl and G. Nejat, “Determining the affective body language of older adults during socially assistive HRI,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 2633–2638.

[60] Z. Wang, E. E. Kuruoglu, X. Yang, Y. Xu, and S. Yu, “Event recognition with time varying Hidden Markov Model,” in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 1761–1764.

[61] P. Bovbel, D. McColl, G. Nejat, and C. Pannu, “Casper - Socially Assistive Humanoid Robot,” 2013. [Online]. Available: http://asblab.mie.utoronto.ca/research-areas/assistive-robotics. [Accessed: 24-Mar-2015].

Appendix A Schedule Collection Template

```
# You are living in a small apartment (areas: kitchen, dining room, bedroom,
# living room, bathroom) similar to the floor layout found at
# http://i.imgur.com/eZzqisN.png

# Your daily activities include eating, meal preparation, sleeping, managing
# your hygiene, and enjoying leisure activities (such as reading, writing,
# tv, talking on the phone, playing video games, etc.)

# Schedule format:
# timestamp activity location
# timestamp activity location

# First activity denotes wakeup, last activity should be final sleeping time
# Valid timestamp format for one o'clock: 1:00, 100, 13:00, 1300

# Valid activities: mealprep, eat, sleep, hygiene, watchingtv, reading, writing,
# phone (add any other leisure-type activities yourself as necessary)

# Valid locations: kitchen, diningroom, livingroom, bedroom, bathroom

# Example schedule:
7:45 hygiene bathroom
8:00 mealprep kitchen
8:15 eat dining
9:00 watchingtv livingroom
10:45 reading bedroom
12:00 mealprep kitchen
12:30 eat livingroom
1:00 watchingtv bedroom
2:00 watchingtv livingroom
2:30 hygiene bathroom
5:00 mealprep kitchen
5:30 mealprep dining
5:45 eat dining
6:30 phone dining
7:00 reading livingroom
1:00 sleep bedroom
```

Appendix B User Interaction Flow

Figures 23-34 captured from demonstration video [61]. Figures 27, 28, 30-33 show tablet interface designed by [41].

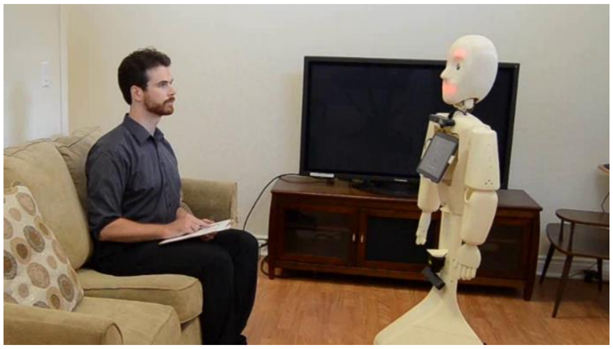

Figure 26: Casper Locates User

Figure 27: Casper Leads User to Kitchen

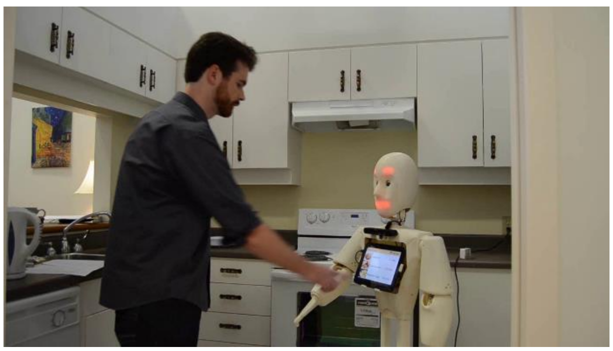

Figure 28: User Selects Recipe

Figure 29: Recipe Selection Screen

Figure 30: Ingredient List Screen

Figure 31: Casper Helps Locate Ingredient

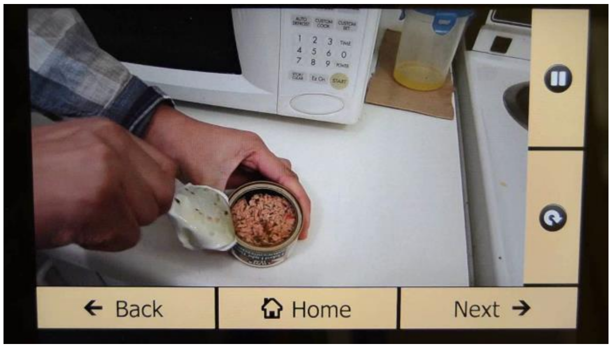

Figure 32: Preparation Step Video

Figure 33: User Prepares Recipe

Figure 34: Final Step Instructions

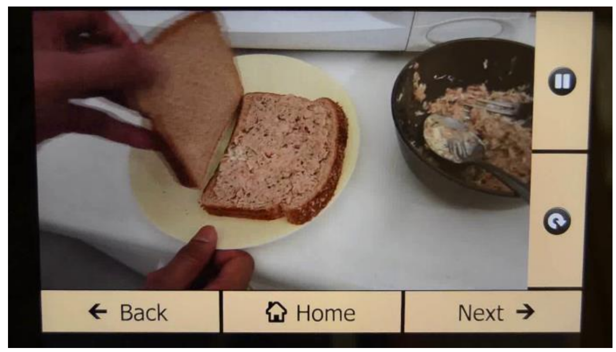

Figure 35: Final Step Video

Figure 36: User Finishes Preparing Meal

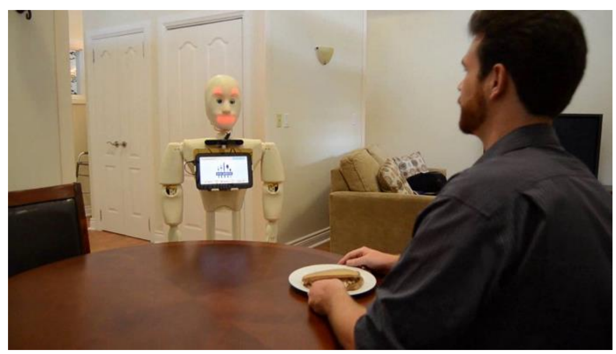

Figure 37: User is Nourished

Appendix C Robot Emotion Display

Figures 35-39 are of Casper's facial emotions, designed by [37].

Figure 38: Happy Emotion

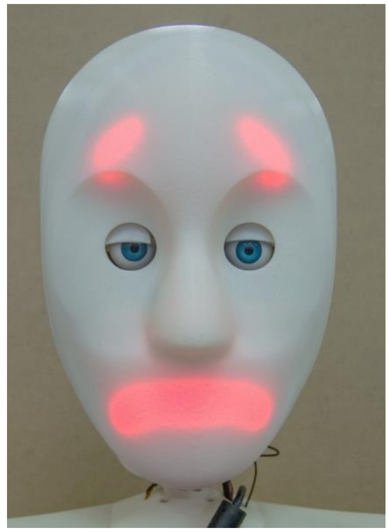

Figure 39: Sad Emotion

Figure 40: Surprised Emotion

Figure 41: Neutral Emotion

Figure 42: Angry Emotion