



Norwegian University of  
Science and Technology

# Intelligent Sliding Doors

Håvar Aambø Fosstveit

Master of Science in Computer Science

Submission date: Januar 2012

Supervisor: Anders Kofod-Petersen, IDI

Norwegian University of Science and Technology  
Department of Computer and Information Science



Håvar Aambø Fosstveit

# Intelligent Sliding Doors

Intelligent systems, master thesis, fall 2011

Faculty of Information Technology, Mathematics and Electrical Engineering  
Department of Computer and Information Science





## Problem Description

Truly smart systems need to interface with the behaviour of human and non-human actors in their surroundings and on their terms

This project aims to develop an intelligent sliding door, which responds to user intentions. The system is to be developed on a physical door using artificial vision.

*Assignment given by:* Anders Kofod-Petersen (supervisor)



## Abstract

You can see sliding doors everywhere, be it at the grocery store or the hospital. These doors are today mostly based on naive, motion sensing, and hence not very intelligent in deciding to open or not. I propose a solution by replacing the traditional sensor with the more sophisticated Microsoft Kinect depth mapping sensor allowing for skeletal tracking and feature extraction. I have applied *hidden markov models* to the behavioural features to understand the human intentions. Combined with a few simple rules, this solution proved to be accurate in 4 out of 5 times in understanding the user's intention in a controlled laboratory test.

**Keywords:** Sliding door, computer vision, Microsoft Kinect, behavioural features, hidden markov model, behaviour recognition, artificial intelligence.





## Preface

This thesis document outlines the work done in my master's project. The project continues from a specialization project done in cooperation with a fellow student, John-Sverre Solem. As such, some of the material in this thesis originates from the report delivered in the specialization project. Section 2.3 is in special written together with John-Sverre Solem. At delivery, this thesis will conclude my degree in Computer Science at the Department of Computer and Information Science at the Norwegian University of Science and Technology.



## Acknowledgments

This project would not be possible without the continuous help from my supervisor, Anders Kofod-Petersen. I would like to thank him for his invaluable advice and assistance throughout the entire course. I would also like to thank my co-supervisor, Richard Blake for all help in the field of computer vision.

Håvar Aambø Fosstveit  
Trondheim, January 22, 2012



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background and Motivation . . . . .	1
1.2	Goals and Research Questions . . . . .	1
1.3	Research Method . . . . .	2
1.4	Report Structure . . . . .	3
<b>2</b>	<b>Theory and Background</b>	<b>5</b>
2.1	Modeling Human Behaviour . . . . .	5
2.1.1	Model . . . . .	6
2.2	Computer Vision . . . . .	6
2.2.1	Computer Vision Theory . . . . .	7
2.3	Reasoning . . . . .	12
2.3.1	Bayesian Network . . . . .	12
2.3.2	Decision Network . . . . .	13
2.3.3	Hidden Markov Model . . . . .	14
2.3.4	Rule Based Reasoning . . . . .	14
2.3.5	Machine Learning . . . . .	15
<b>3</b>	<b>Research Results</b>	<b>19</b>
3.1	Modeling Human Behaviour . . . . .	19
3.1.1	Behavioural Features . . . . .	19
3.1.2	Intention . . . . .	21
3.2	Computer Vision . . . . .	21
3.2.1	Evaluation of Segmentation Methods . . . . .	21
3.2.2	Kinect . . . . .	23
3.3	Reasoning . . . . .	23
3.3.1	Hidden Markov Model . . . . .	23
3.3.2	Setting up the Model . . . . .	24
3.3.3	Data Analysis . . . . .	24
3.4	A Sliding Door . . . . .	30
3.5	Application . . . . .	30
3.5.1	Application Structure . . . . .	31
<b>4</b>	<b>Evaluation</b>	<b>33</b>
4.1	Testing . . . . .	33
4.1.1	Performance Measure . . . . .	33

<b>5 Conclusion and Future Work</b>	<b>39</b>
5.1 Discussion . . . . .	39
5.2 Future Work . . . . .	40
<b>Bibliography</b>	<b>41</b>
<b>A Manuscripts</b>	<b>43</b>

# List of Figures

2.1	Features for modeling intention . . . . .	6
2.2	Application of Otsu . . . . .	8
2.3	Application of Canny . . . . .	9
2.4	Application of image subtraction . . . . .	9
2.5	Application of HOG descriptor . . . . .	10
2.6	Steps of HOG descriptor algorithm . . . . .	10
2.7	Example image . . . . .	11
2.8	A simple Bayesian network . . . . .	12
2.9	A decision network . . . . .	13
2.10	A general Hidden Markov model . . . . .	14
2.11	Example rule base for project . . . . .	15
2.12	A simple decision tree . . . . .	16
3.1	The different levels of proximity (from Solem (2010)) . . . . .	20
3.2	Model of human skeleton with features of interest . . . . .	21
3.3	Location grid . . . . .	25
3.4	Example of location data . . . . .	26
3.5	Trained example hidden markov model . . . . .	27
3.6	Noise in feature data . . . . .	28
3.7	A trained HMM . . . . .	29
3.8	A sliding door . . . . .	30
3.9	Application structure . . . . .	31
3.10	Training application structure . . . . .	31





# List of Tables

1.1	Search engines used for literature survey . . . . .	2
1.2	Search terms used in literature survey . . . . .	3
3.1	Features, ranges and units . . . . .	20
3.2	Segmentation performance . . . . .	22
4.1	Test results . . . . .	34
4.2	Rates of the different result classes . . . . .	35
4.3	Statistical measures for results in percent. . . . .	36
4.4	Statistical measures for traditional sliding door (modified from Solem (2010)). . .	37
4.5	Comparing performances of different doors. . . . .	37
4.6	Comparing phi coefficient. . . . .	37



# Chapter 1

## Introduction

In the following sections I describe the background and motivation for doing the masters thesis. I define the goals and research questions for the work. I also provide a description of the research method used in the thesis work, including the literature survey and documentation method. Finally I give an overview of the report structure, the chapters and their content.

### 1.1 Background and Motivation

The background for this project is the extension of a specialization project, and the completion of my masters degree at the Norwegian University of Science and Technology. The main motivation for the work is given in the article by Kofod-Petersen and Cassens (2009). This article points out the weaknesses of today's automated sliding doors in the context of ambient intelligent systems, and outlines the challenges of interpreting human intentions. The work of feature extraction within the computer vision field is, although not trivial, a well-known task. The connection between these features and human intentions, however, is not. It requires a complete domain model of human behaviour, composed of different movements, together with an inference mechanism for intentions.

This challenge is now lifted from a sub-symbolic to a symbolic level. Making this lift, generalizes the process, giving a result that can easier be transferred to other similar tasks. Another motivational factor for this work is to lay down foundations, by exploring the possibilities within the challenge, for applying the methods in other fields like pedestrian surveillance. This will possibly allow the methods researched here to be applied in pedestrian crossings and in understanding the intentions of pedestrians.

### 1.2 Goals and Research Questions

**Goal 1** Design a model of features, human behaviour and intentions.

Define a set of features needed in order to describe human behaviour in the context of a door. Quantify the features, in a manner that the model is suitable both for feature extraction and intentional reasoning.

**Goal 2** Design a mechanism for capturing and extracting features according to the model.

Do a study in Computer Vision in order to find the required components and suitable tools for capturing and extracting the features as described in Goal 1.

**Goal 3** Design a reasoning mechanism for inference of intention.

Do a study within artificial intelligence theory in order to find a mechanism able to make the decision about opening a door based on the type of features as described in Goal 1. The mechanism must be able to conclude about the intentions of the person in front of the door.

**Goal 4** Implement the reasoning mechanism from Goal 3

Develop an application able to make the door reason about a humans intentions.

**Research question 1** What set of computer vision algorithms will meet Goal 2 efficiently?

Test the different algorithms in order to find a combination that performs well enough for real-time performance.

**Research question 2** What is a well suited reasoning mechanism for this task?

Test the different mechanisms found in Goal 3, in order to find the one with best accuracy regarding the actual intentions of a person in front of a door.

### 1.3 Research Method

The work done in this project can be divided into stages. The stages are described in the following list.

**Problem overview** In this stage I worked on the problem description, defining the problem area, what would and would not be included. I had meetings with my supervisor Anders Kofod-Petersen to find out where to focus my attention according to work previously done in the specialization project and work done by John Sverre Solem in his masters thesis.

**Literature survey** This stage was mainly performed in the specialization project, where I did research on similar problems, existing works and projects. For this purpose I made a table of search terms (see Table 1.2) relevant to the project, dividing it into categories corresponding to the parts as defined in the previous stage. I ran different combinations of these search terms in several digital libraries (listed in Table 1.1). Different combinations gave different levels of quality and relevancy in search results. The searches made in the work with the masters thesis can be seen below the line in the table.

Table 1.1: Search engines used for literature survey

Search engine	URL
IEEE Xplore	<a href="http://ieeexplore.ieee.org">http://ieeexplore.ieee.org</a>
SpringerLink	<a href="http://www.springerlink.com">http://www.springerlink.com</a>
ISI Web of Knowledge	<a href="http://www.isiknowledge.com">http://www.isiknowledge.com</a>
ScienceDirect	<a href="http://www.sciencedirect.com">http://www.sciencedirect.com</a>

**Component research and evaluation** Researching the different parts of the respective fields and evaluate according to the research goals.

Table 1.2: Search terms used in literature survey

Sensor	Computer Vision	Model/Reasoning	Human Behaviour
stereo vision	stereo vision	knowledge base	intention
camera	motion detection	learning	movement
motion sensor	segmentation	reasoning	anatomy
sensor fusion	kalman filter	retrieval	body language
	facial recognition	decision	posture
	vector	semiotics	hip
	proximity	syntax	pose
	marker-less	semantic	body alignment
	motion tracking		gaze
			gaze direction
			human behaviour
kinect	depth mapping	hidden markov model	behaviour recognition

**Development** This stage consisted of expanding on the framework laid out by Solem (2010). Later I had to try and develop a better reasoning mechanism.

**Testing** This included gathering data to train a *hidden markov model* and after gathering data from interactions with the door, testing had to be done to evaluate the performance of the reasoning mechanism.

**Thesis writing** This stage was done throughout the entire project. This consisted of refining documentation already attained from the specialization project and processing new documentation and research.

**Meetings** Meetings with my supervisor, Anders Kofod-Petersen discussing and planning current and future work.

## 1.4 Report Structure

This report documents the progress through my master's project. It first introduces the main problem with computers understanding human intentions, in this case sliding doors. It goes on to cover the theory and background of the research fields in question. Further it goes on to describe my solution and at last the evaluation of the solution.

The report goes through the general research in the fields of computer vision and artificial intelligence in Chapter 2 according to the goals and research questions. It goes on to describe the results of the research in Chapter 3 giving reasons for the choices made. It concludes with the evaluation of performance and discussion and future work in Chapters 4 and 5 respectively.



## Chapter 2

# Theory and Background

The project can roughly be divided into two parts. The first part is the data collection in the form of computer vision discussed in Section 2.2. Further, the second part is the reasoning and decision making from the collected data as explained in Section 2.3. For the two parts to work together, they need to agree on a basic understanding of the domain. This is the model that we elaborate in Section 2.1 as a basis for the two parts.

These parts come together to form the building blocks for a *more intelligent* sliding door. By more intelligent, we mean more intelligent than traditional sliding doors that rely purely on motion sensing.

### 2.1 Modeling Human Behaviour

An intelligent door must understand the intentions of human beings in order to know if it should open or not. Understanding the intentions of humans is by no means a trivial task. Human behaviour can be complex, and sometimes even irrational. A person walking towards a door can suddenly change to standing still, reading the newspaper in the stand beside the door. Perhaps the newspaper was the intended destination all along and not the door. Then again, the door could have been the initial target, but the front page of the newspaper made the person change his mind.

Martinec (2001) has done research on resources of movement focusing on interpersonal relations. He describes a model for actions, using parameters like body angle and distance. He also describes sign functions, mapping movements to meaning. His work can be used for developing a framework for interpreting body language. The work is based on previous work by Hall, following concepts posed by Halliday. Moore (2008) extends the works of Martinec by describing a context dependency, stating that the values (like distance and angle) valid for one context may not be valid for another context, using surgery as a point of reference. Guerra-Filho and Aloimonos (2006) take another approach, presenting a *Human Activity Language (HAL)* for symbolic non-arbitrary representation of visual and motor information. This language is based on the empirical discovery of a linguistic framework for the human action space. The described space has its own phonemes, morphemes and sentences. This approach uses learning algorithms for the different actions. Yet another approach is proposed by Amano et al. (2005). Here we are presented with a linguistic representation of human motion, based on the knowledge representation scheme proposed in *The Mental Image Directed Semantic Theory (MIDST)*. A formal language is defined, with syntax and semantics. The suggested application is interpretation of human motion data from a motion capture system. The movements in this approach is described

with *Locus formulas*. The approaches made by Guerra-Filho and Aloimonos and Amano et al. are similar, but while the latter one initially requires a full description of the modeled actions, the first one uses learning algorithms.

### 2.1.1 Model

One of the goals for the project is to lift the reasoning process from a sub-symbolic to a symbolic level. This implies the abstraction of the pixel stream from the cameras into symbols like position and speed. The symbols to use in this case are features extracted from the video stream.

The features that can be extracted from one single video stream are numerous. Adding an additional camera gives even more possibilities. Some of the features are more suited than others, and Kofod-Petersen and Cassens (2009) suggests the use of body alignment, proximity and visual target as features of human behaviour suited for modeling intention. The latter one is later discarded as being of low value concerning intention.

The body alignment feature is divided into two features, the orientation of the shoulders (shoulder angle) and the hips (hip angle), where the measured angles are relative to a point of origin. This point will be the door in most of the cases, but can also include other people, when more than one person is captured by the cameras.

The proximity feature gives a measure of closeness to the door. When adding the perspective of time, this feature can be used to extract another feature, motion. This is useful in distinguishing between people moving towards the door and moving away from the door. We now have the full model of features suggested by Kofod-Petersen and Cassens (2009) as shown in Figure 2.1.

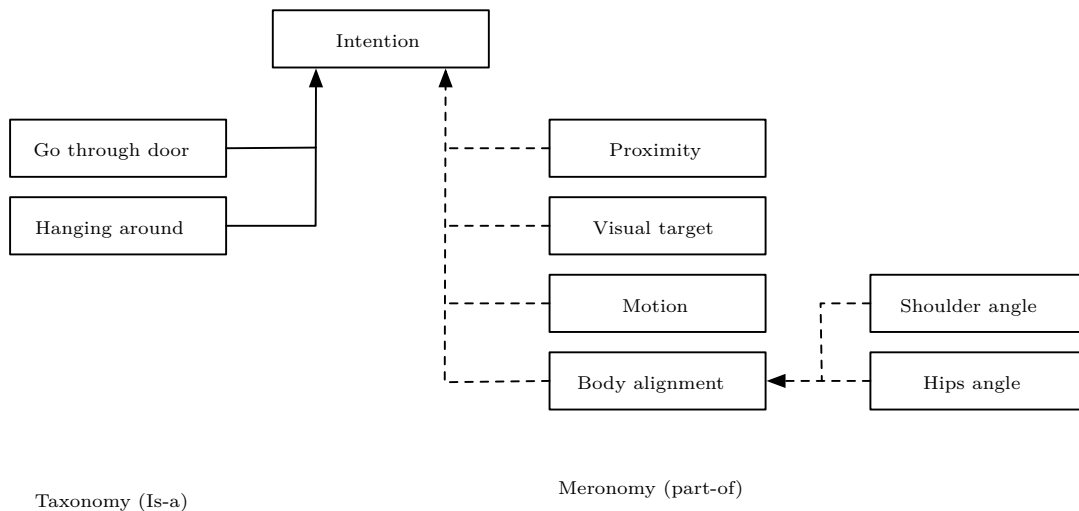


Figure 2.1: Features for modeling intention (from Kofod-Petersen and Cassens (2009))

## 2.2 Computer Vision

There has been much research into the low-level processing of image data. As a result of this, there are several different ways of segmenting and extract features from various image sources.



Computer vision is the science and technology of machines that see, where see in this case means that the machine is able to extract information from an image that is necessary to solve some task (Sonka et al., 2008).

The detection and tracking of humans are easy tasks for humans, but is difficult for a computer for a variety of reasons. The human body can be morphed into many different poses, be clothed in a myriad of different clothes and carry accessories. All this comes in addition to the problem with the scenery, weather and lighting conditions in which to do the detection.

In Giosan et al. (2009) we can see the use of stereo vision in marker-less pedestrian detection. This system uses full body contours when detecting humans. A result of this is that the human model used for comparison is a library of contours derived from many different poses. The use of stereo vision enables it to extract 3D information from the captured data, and this information is used in combination with simple 2D edge detection to provide better results regarding foreground/background separation.

Caillette and Howard (2004) uses a different strategy in which 3D models are extracted. This is achieved using several cameras capturing an object, in this case a human, and running calculations on the captured images. It produces a 3D voxel representation which is then matched to a kinematic representation in the same 3D space. Because of this kinematic model that has to fit to the object, the tracking can be very accurate, but requires the system to know the model beforehand.

A 3D representation is also used by Corazza et al. (2007). In this system they create the 3D representation by using the technique *visual hull*.

### 2.2.1 Computer Vision Theory

Early research suggested stereo vision as a valuable tool for data capturing. Stereo vision employs the use of two cameras placed above the door and slightly separated to simulate two eyes. This would give us a rough three dimensional representation similar to that which a human sees using stereopsis (Sonka et al., 2008).

Stereo vision requires the use of epipolar geometry to calculate the relation between points in the respective images. When two cameras view a 3D scene from two distinct positions, there are a number of geometric relations between the 3D points and their projections onto the 2D images that lead to constraints between the image points. These relations are derived based on the assumption that the cameras can be approximated by the pinhole camera model (Sonka et al., 2008).

As an alternative to stereo vision, a new opportunity opened with the release of the Kinect depth mapping camera. This approach uses an infrared laser grid to extract 3D information in front of the sensor.

There are several stages in the data collection pipeline. The first being the actual image capturing. Second, some sort of algorithm has to be applied to segment the image. Following this, features has to be extracted and finally converted to symbols. This is as we can see, a task of getting from a sub-symbolic level to a symbolic level which is non-trivial (Sonka et al., 2008).

### Segmentation

The main goal of the segmentation process is to divide an image into parts containing information of interest. To do this, several methods and algorithms may be used depending on the problem at hand. Taken what we know about the problem domain in the case of sliding doors, we see that we have some potentially very noisy data and difficult objects to track. This makes segmenting the image very hard to do in a consistent way.

The simplest form of segmentation is by applying a threshold, and examples of this include the widely used Otsu algorithm. Applying a threshold works well when there is a clear difference in contrast between the foreground and background. If this is not the case, you can have a hard time getting consistent areas of interest if any at all as seen in Figure 2.2. In the case of a camera pointing at an entrance from above a door there will in many cases be a cluttered background. As such, a threshold in its own will not suffice and must in any case be combined with another form of processing to be effective (Sonka et al., 2008). Shadings and lighting conditions further challenge the threshold application. One way to overcome situations like this, is to use an adaptive threshold. In this case, the threshold value can vary over the image. The variation can be achieved either by using a function of local image characteristics, or simply by dividing the image into smaller images, thresholding each one separately.



Figure 2.2: Application of Otsu: From left to right, original, gray scale and Otsu applied (Original image: Matt Banks / FreeDigitalPhotos.net)

Another form of widely used and easily implemented segmentation is edge based segmentation. Two of the most used methods are Sobel and Canny edge detection. These methods share much of the same shortcomings regarding the sliding door as applying a threshold as can be seen in Figure 2.3.

In Oral and Deniz (2007) we see the use of image subtraction. There are several methods of doing image subtraction, the most simple being *simple background subtraction*. This is done by comparing two frames, pixel by pixel, and marking the pixels with an absolute difference higher than a given threshold as shown in Figure 2.4. This method can be quite good in finding movements, and Oral and Deniz (2007) compares several image subtraction methods.

There are also other alternatives to the segmentation problem, and one of the candidates to run is some sort of object recognizer. By using a HOG descriptor<sup>1</sup> we can train the system to detect people in a given frame. This provides us with bounding boxes in which there ideally will be one person. We can then separate the image into these boxes for further analysis.

---

<sup>1</sup>Histogram of oriented gradient (HOG) descriptors are feature descriptors used in computer vision and image processing for the purpose of object detection.



Figure 2.3: Application of Canny

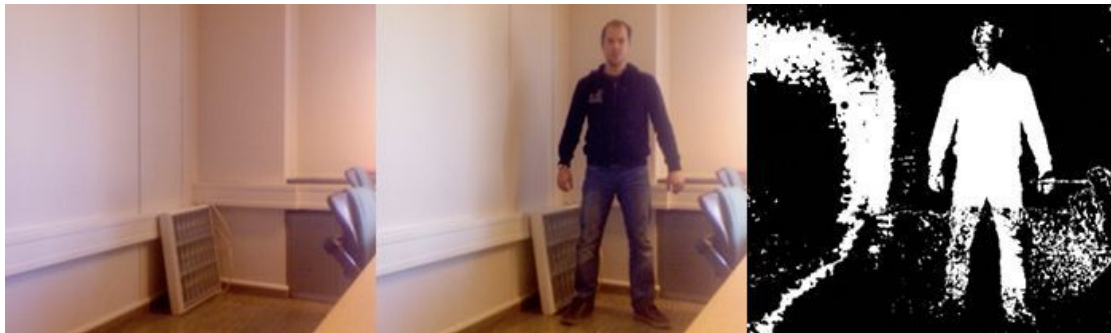


Figure 2.4: Application of image subtraction

### Histogram of Oriented Gradients Descriptors

Histogram of Oriented Gradients (HOG) descriptors are general feature descriptors used primarily for object recognition purposes. When trained correctly, HOG descriptors give a good detection rate for frames containing full body images of humans as can be seen in Figure 2.5. The algorithm laid out by Dalal and Triggs (2005), uses several steps to compare images to the HOG descriptors as can be seen in Figure 2.6. In general, this algorithm divides the image window into small spatial regions, called cells. For each cell it accumulates a local 1D histogram of gradient directions or edge orientations over the pixels of the cell. The combined histogram entries of all the cells form the representation. As can be seen in the first step in Figure 2.6 contrast-normalization can be useful to obtain better invariance to illumination, shadowing, etc. on the local responses before using them. The normalization can be done by finding a measure of local histogram "energy" over somewhat larger spatial regions called blocks, and using the results to normalize all the cells in the block. It is these normalized descriptor blocks that are called *Histogram of Oriented Gradient (HOG)* descriptors.

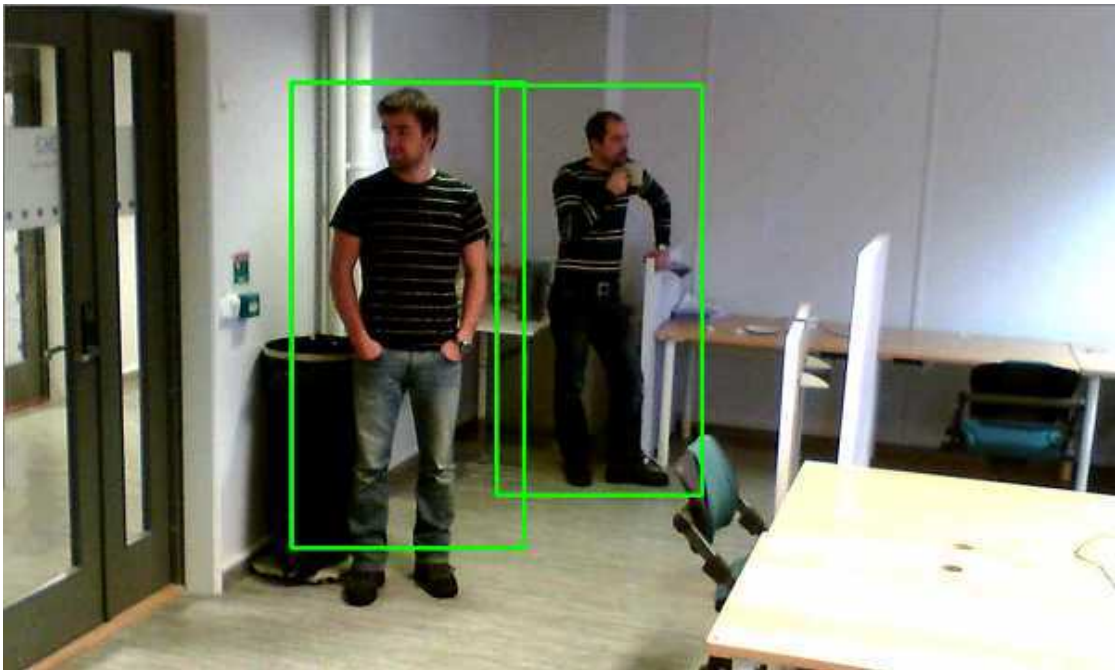


Figure 2.5: Application of HOG descriptor

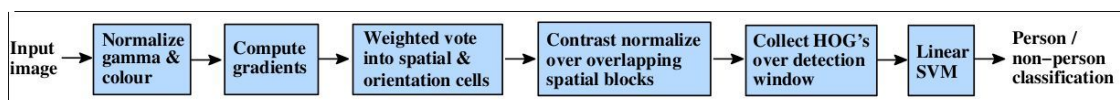


Figure 2.6: The steps from left to right in the HOG descriptor algorithm (Dalal and Triggs, 2005)

## Depth Mapping

Depth mapping is the use of lasers or other ranging equipment to get a 3D representation of the space in front of the sensor. Microsoft released an affordable solution for depth mapping with their Kinect sensor for the Xbox 360 gaming platform in November 2010. This sensor projects an infrared laser grid, and compares the resulting returned image with a base image to give a 3D representation of what is in front of the sensor as can be seen in Figure 2.7. This 3D representation is returned to the user in the form of a depth map ready for feature extraction.



Figure 2.7: Example of a depth image. Lighter gray means closer to camera.

## Feature Extraction

The segmentation process reduces the raw image data into a more manageable amount of relevant data. The ratio of information to data is still too low, the input data must be transformed into a reduced representation set of features, a process called feature extraction (Sonka et al., 2008).

Horaud et al. (2009) does this by first obtaining 3D data from several cameras pointed at the same spot. The images from these cameras are then segmented to subtract the background from the human. These segmented images showing silhouettes are then compared to an already constructed kinematic model of a human, consisting of connected ellipsoids. The best match then represents the pose made by the human.

In this project, the information computed in the segmentation step is not sufficient to have a complete system. Without feature extraction we would not be able to supply the features

required by the model described in Section 3.1. The reasoning process is dependent on these features, and as a result feature extraction is necessary for making a decision through reasoning.

## 2.3 Reasoning

When the wanted feature data has been collected, there are still some work to be done. The door must be given a command *open*, or simply ignore the activities in front of the door. Reasoning over the collected feature data gives us the possibility to automate the decision making process and thereby the power to control the door. The model contains sufficient information about the features for the reasoning process to conclude about intention enabling the system to send the command.

The reasoning process must be able to take some input parameters, validate them against the model, and output a response needed for a command to be sent to the door. There are several techniques that can be suited for this kind of work. We will discuss the ones most relevant for this project in the following sections.

### 2.3.1 Bayesian Network

A Bayesian network or belief network is a network of variables and their dependencies. It is constructed as a directed acyclic graph. The Bayesian network is a probabilistic model in the sense that each node is associated with a probability function. The inputs to this functions come from the parent nodes, representing observed events, and in case of missing observations, the probabilities of these events.

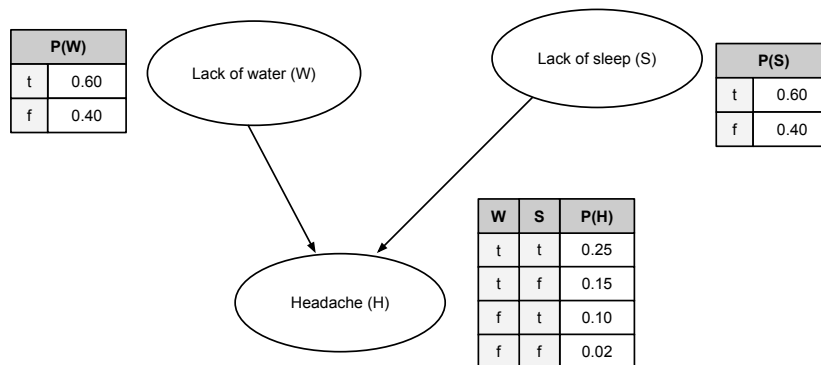


Figure 2.8: A simple Bayesian network

Figure 2.8 illustrates a simple Bayesian network with three nodes. The network models *headache* as an effect of lack water and/or lacking sleep. The probability distributions for each of the nodes are given. This model can answer questions like "What is the probability of getting a headache if I make sure I get enough sleep?" or "Why do I have a headache?".

Using a Bayesian Network for this project could model the features (see Table 3.1) as nodes, with possible dependencies between them, related to a child node *Intention*. The intention is then a probabilistic evaluation of all the features. This value is in other words the probability of a person wanting to go through the door. If it is higher than a set threshold we can open the door, if not, leave the door unopened.

A challenge related to this approach would be to set up accurate possibilities for each event. It is crucial to get a correct and close-to-reality model when dealing with the hard-to-observe intentions of human beings.

### 2.3.2 Decision Network

When making decisions, a useful tool can be decision networks<sup>2</sup>. This is a general mechanism for making rational decisions, following the principle of maximum expected utility (Russell and Norvig, 2010). The decision network is similar to a Bayesian network, but includes extra nodes for actions and utilities. The chance nodes (oval) represent the random variables, like in the Bayesian network. The decision nodes (rectangle) represents the choices to be made. The utility nodes (diamond) represents the utility function that describes the value of the results from a decision. The choices made can influence different parts of the network. By describing the utility of different states, it is possible to choose the decision that maximizes the utility.

Figure 2.9 shows how the choice of an *Airport Site* influences *deaths*, *noise* and *cost*. This influence is based on the nodes *Air Traffic*, *Litigation* and *Construction*. The utility node takes the deaths, noise and cost in consideration to conclude about the best Airport Site.

For this project the network would be the same as the Bayesian network, the only difference being the addition of a decision node *OpenDoor* and a utility node. The utility function must then evaluate the usefulness of the door opening or not in the different settings. This extension might not be useful as the value of the decisions would only reflect the probability of the intention, and most likely give the same answers as the plain Bayesian network would.

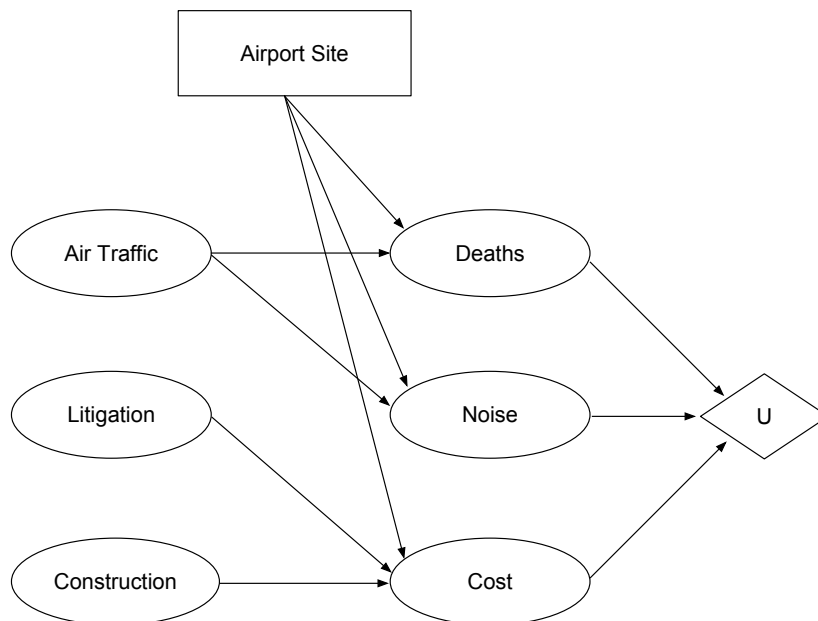


Figure 2.9: A decision network (recreated from Russell and Norvig (2010))

---

<sup>2</sup>Also known as influence diagrams

### 2.3.3 Hidden Markov Model

A Hidden Markov model (HMM) is a temporal probabilistic model in which the state of the process is described by a single discrete random variable (Russell and Norvig, 2010). HMMs are useful when working with an environment that changes over time. The model can utilize transition models and sensor models to predict future states based on current observations. Figure 2.10 illustrates an HMM with  $E$  as the evidence variable, and  $X$  as the hidden state variable.

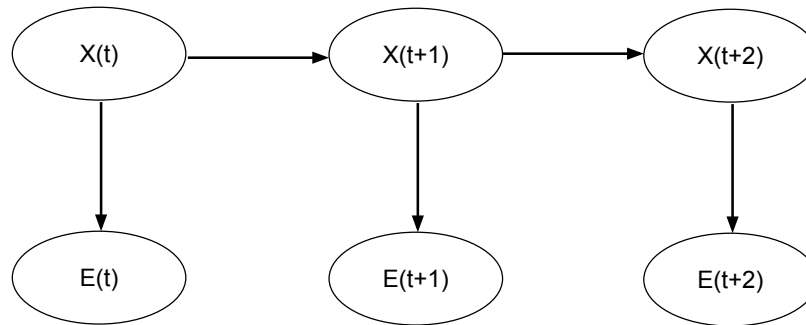


Figure 2.10: A general Hidden Markov model

Following this project, an HMM can model the intentions of people picked up by the sensors above the door. The intention can be seen as the future position of a person. If a person is outside but the model shows that the future position is inside, the intention is to go inside. The Hidden Markov model requires a single variable, but can model more complex environments by combining several variables into one big variable. Following this procedure, the features can be combined into one variable in order to model the intention.

### 2.3.4 Rule Based Reasoning

Rule based reasoning utilizes a set of predefined rules in a knowledge base combined with some input facts to find the solution to a problem. The approach is simple, but effective. It requires a set of rules to be formed that model the problem domain. When facts are given, the system can run through the rule base looking for rules that match the given conditions. On match the system performs the appropriate actions.

Rules are typically of the form *IF [condition] THEN [action]*. The system loops over the rules, or a subset of the rules until some condition is met, thus giving a control flow for execution of the different parts of the program. Figure 2.11 lists a set of example rules that could have been used for this project.

Working with rule based systems leaves little room for situations not captured by any rule at all. This again might lead to either an exhausting set of rules, capturing most parts of every thinkable situation, or fewer rules but with one or more all consuming rules that capture the situations not expected. For this project it is easy to define that when no intention is shown to go through the door, no opening of the door is necessary. This captures all the situations that are not defined by the rule base. The challenge, however, is to define rules that are good enough for all the situations where the door should open. Figure 2.11 demonstrates a small set of primitive rules that involve features for the door opening.



```
WHILE(RUNNING){
  IF detect = humanDetect() != TRUE
  THEN BREAK

  IF detect = TRUE
  THEN trackHuman()

  IF hipAngle + shoulderAngle = 0
  THEN orientation = door

  IF proximity < threshold
  THEN closeToDoor = TRUE

  IF speed > 0 && orientation == door
  THEN heading = door

  IF heading == door && closeToDoor
  THEN intention = walkThroughDoor

  IF speed == 0 && orientation == door && headAngle == 0
  THEN intention = walkThroughDoor

  IF intention == walkThroughDoor
  THEN openDoor()

  IF intention != walkThroughDoor
  THEN closeDoor()
}
```

Figure 2.11: Example of a rule base for this project

### 2.3.5 Machine Learning

Machine learning is an approach to reasoning where a complete model of the domain is not required. This approach aims at automatically building a knowledge base substituting the prerequisite of an omniscient model. The techniques described in this section is not what is primarily aimed for in the goals of the project, but will serve as a perspective of alternative approaches subject to future work for the reason of comparison. Machine learning can be divided into three cases: supervised, unsupervised and reinforcement learning (Russell and Norvig, 2010). Supervised learning requires someone or something giving feedback about the outcome of a choice. Unsupervised learning is learning in the case of lacking output, hence no feedback. Reinforcement learning is learning based on reinforcements rather than being told what to do. The reinforcements can be seen as rewards of different sizes given according to the choices made.

In this case, we have some domain knowledge, and the possibility of telling the agent what is right and wrong. The task of learning from scratch without feedback, would be impossible, since the door only action is to open or not. No knowledge about when to open, results in the door always being closed<sup>3</sup>. Since the door must be automated, the learning is better done in a training process, where the cases are classified by intentions (want to enter, does not want to enter).

#### Decision Tree Learning

A decision takes objects with a set of attributes, and returns a decision. Decisions are made by feeding the attributes to the non-leaf nodes of the tree, following the branches corresponding to

<sup>3</sup>The feedback in a case like this could have been provided by a door opening switch, letting the door know that it should have opened, when it did not. Perhaps an idea for future work.

the value of the attribute, ending in a leaf node that gives the result of the decision. A simple and illustrative decision tree is shown in Figure 2.12. The tree takes attributes that correspond to the features of interest.

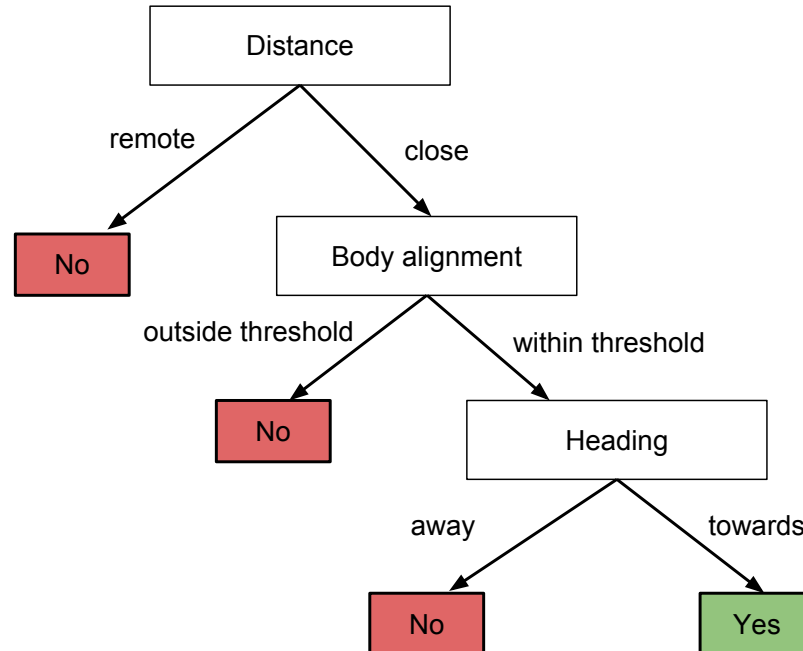


Figure 2.12: A simple decision tree for this decision problem; Leaf nodes annotate decisions. The Yes-node is the open door decision, No-node is the decision to leave the door closed.

Learning in the decision tree is mainly a classification problem. A set of training data must be used where the correct decision is supplied. Then following a decision tree learning algorithm, a tree can be built that classifies the training data.

The challenges in this approach is handling noise (decisions based on irrelevant attributes) and over-fitting (over-specified tree not addressing new cases). Good, representative training data, together with a well developed decision tree learning algorithm provide a robust decision tree, allowing for qualified decisions.

### Case-Based Reasoning

Case-based reasoning (CBR) is a method for problem solving and learning. According to Aamodt and Plaza (1994) CBR is *"to solve a new problem by remembering a previous similar situation and by reusing information and knowledge of that situation"*. The method can be divided into four main steps: Retrieve, Reuse, Revise and Retain (Aamodt and Plaza, 1994).

The four steps of CBR:

- Retrieve the most similar problems from the case-base
- Reuse the solutions that are applicable

- Revise the suggested solution
- Retain the new problem/solution for later use

This is a strong and adaptive technique, with its strength in the continuous build of a growing knowledge base. For this reasoning problem, this might not be the best approach. With few methods of providing continuous feedback, we are better off left with a more static learning phase.



# Chapter 3

## Research Results

### 3.1 Modeling Human Behaviour

To be able to detect differences in behaviour, we need a model to describe it. This model will aid us in the capturing of data, and later, the processing of it. This basic building block is essential to the reasoning engine to be able to observe human intentions.

The model mapping the features to intentions is based on the model suggested by Kofod-Petersen and Cassens (2009) and described in Section 2.1.1, and the further work by Solem (2010).

#### 3.1.1 Behavioural Features

Human motion comprises several features interesting for understanding human behaviour. First we have the pure motion vector that determines the *speed* and *current heading* of the person. In addition we have *acceleration* and *angular velocity* to determine future motion. Speed and heading will tell us if a person is moving towards the door at all. Acceleration and angular velocity can tell us something about the near future if the person is turning towards the door, or starting to move from a stand still close to the door.

Motion as a feature requires observation over time as a single frame will only give us one observation which is not enough to get a motion vector. To measure speed and heading we need at least two observations. Acceleration and angular velocity will need at least three observations as they measure the change in speed and heading. Because of this need for sequential data, the system needs to identify a person, and keep track of him or her as long as it is relevant for the door to make a decision.

Another feature not directly connected to human behaviour in this sense is proximity. This is simply the distance from the person to the door, and it gives us a measure of how much time we have to decide to open or not. A person far from the door is less important than a user close to the door. Although a person have all intentions of going through the door when he is 10 meters away, that can change as the person gets closer, and the proximity measure gives us time to postpone the decision. Proximity is divided into three regions: *close*, *nearby* and *distant* as seen in Figure 3.1.

The last two features as defined by Kofod-Petersen and Cassens (2009) are related to a persons body alignment. More precisely, we measure the angle of the shoulder and hip in relation to the door.

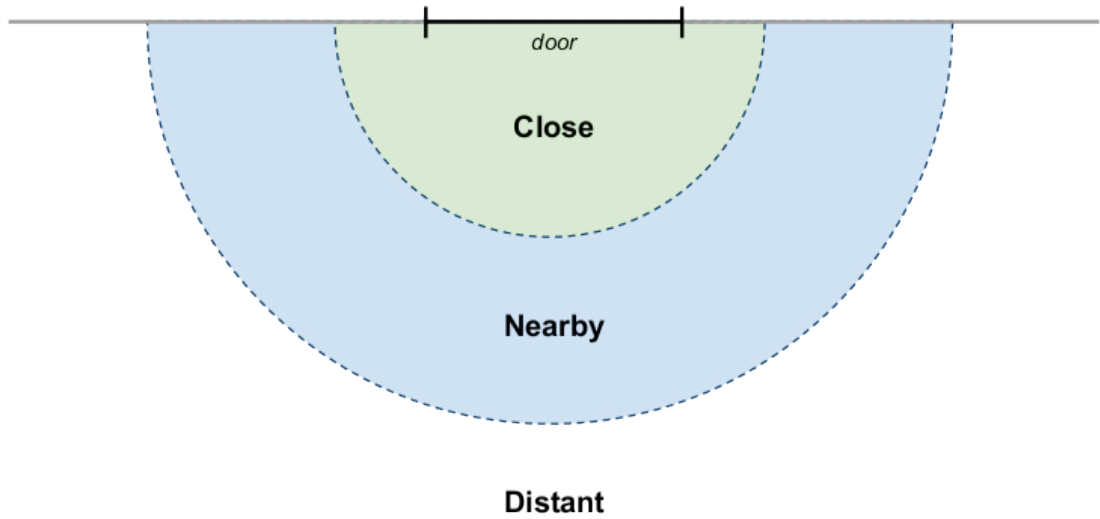


Figure 3.1: The different levels of proximity (from Solem (2010))

Table 3.1: Features, ranges and units

Main feature	Feature	Range/Unit
Motion	Speed	$[-10, 10] m/s$
	Acceleration	$[-5, 5] m/s^2$
	Heading	$[0, 360)^\circ$
	Angular velocity	$[-\frac{\pi}{2}, \frac{\pi}{2}] rad/s$
Proximity	Distance	$[0, 10] m$
Body alignment	Shoulder angle	$[0, 360)^\circ$
	Hip angle	$[0, 360)^\circ$

### 3.1.2 Intention

All these behavioural features can be analyzed in order to try and understand human intentions. The features have certain range values based on what a human is capable<sup>1</sup> of doing as seen in Table 3.1. The different combinations of values for each of these features will be fed into the reasoning engine to try and decide to open the door or not.

## 3.2 Computer Vision

We have found the features we need to know about a person in front of the door. This presents us with a list of requirements for what the computer needs to be able to detect. This being in a real environment and operating on real-time data requires the sensor and computer to be able to handle the incoming data in *real-time*. It needs to be able to recognize and keep track of a person over time in three dimensional space to be able to get the various motion and angular vectors required. Based on these requirements we can test the various computer vision tools presented in Section 2.2.

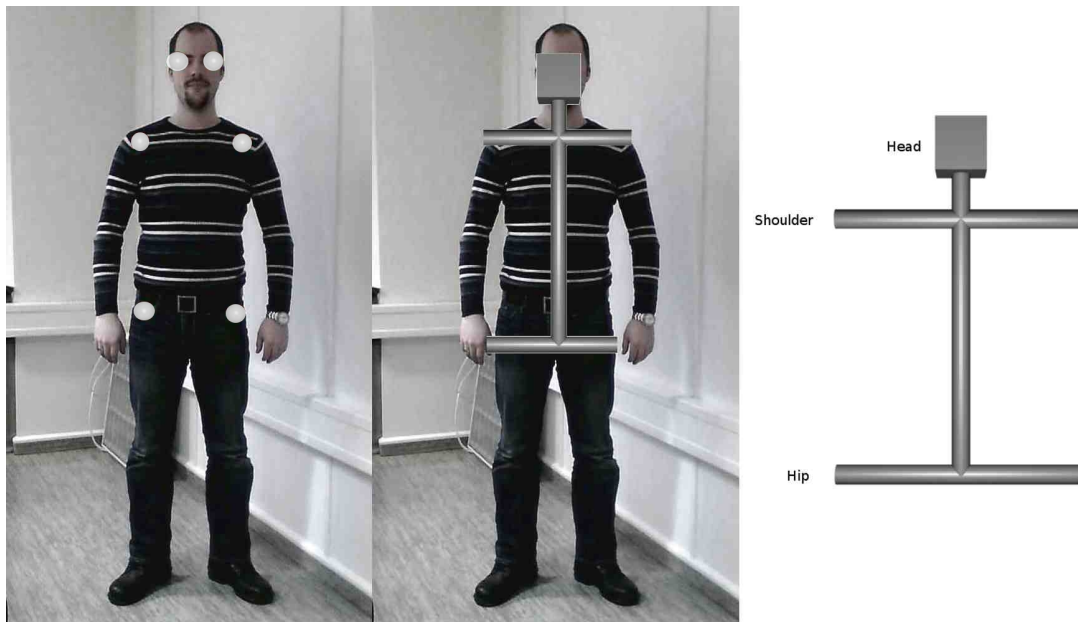


Figure 3.2: A model of a human skeleton simplified to have only the features of interest namely hip angle, shoulder angle and gaze direction

### 3.2.1 Evaluation of Segmentation Methods

All testing was done on a dedicated computer with an Intel Core 2 Duo 3.0 GHz processor and 4 GB of RAM. As this is going to be placed at a door, it may at a later stage have to be tested on other hardware more suited to be permanently placed in a real world environment. These are

<sup>1</sup>Values are a little exaggerated to be able to handle the odd drunk trying to ram the door.

also fairly powerful specifications compared to smaller and more mobile computational solutions. It should however provide the best possible scenario for the algorithms to be tested under in regards to real-time performance.

### Threshold and Edge Detection

These simple and quick algorithms as described in Section 2.2 are well suited for detecting changes in contrast in an image. This works well for statically lit environments, and can in those cases give good results. This however is not the case in situations where the edges between the foreground and background are not easily discernible, as they can be in real environments.

As seen in Table 3.2 these algorithms can be applied in real-time using OpenCV<sup>2</sup>. They do however neither detect humans in an image by themselves, nor give us a way of getting motion vectors for the respective humans, and have to be combined with other methods to complete these tasks.

Table 3.2: Segmentation performance

Image size	Segmentation	Average time
640x480	Sobel edge detection	1 ms
640x480	Canny edge detection	7 ms
640x480	Otsu threshold	2 ms
640x480	Image subtraction	1 ms
640x480	HOG detector	517 ms

### Histogram of Oriented Gradients

OpenCV provides the basic HOG descriptor algorithm described by Dalal and Triggs (2005). It also has a default human detector for testing performance. This algorithm works well, but is in its original form not fast enough for real-time application as can be seen in Table 3.2. In order to get the performance up to at least 15 frames per second, the image must be kept to a very low resolution. There has been later work in improving the HOG algorithm (Zhu et al., 2006; Jia and Zhang, 2007) for use in real-time applications showing promise. The HOG algorithm does however not provide any depth data, nor any tracking of features within a human. Because of this it does not fulfill the requirements for the computer vision.

### Depth Mapping

Depth mapping can be tested using the Kinect sensor provided by Microsoft. This sensor provides depth data in the form of 640x480 pixel depth maps at a rate of 30 frames per second. This satisfies the need for both depth data and real-time performance without any need to calculate anything on the computer. This leaves the processor free to perform feature extraction on the incoming data.

It seems based on this that depth mapping is the best alternative to the task of segmenting the data before the feature extraction.

<sup>2</sup>OpenCV is a library aimed mainly at real time computer vision. It was developed by Intel, but is now supported by Willow Garage. It is free under the open source BSD license.



### 3.2.2 Kinect

The kinect provides, in combination with third party libraries, the opportunity to track people in front of it. Work done by Solem (2010) gives a framework for using the kinect in this regard, and provides us with almost all the features we need.

In addition to the features provided by the framework laid out by Solem (2010) we need to find the angular velocity. This is done by taking the difference in two consecutive headings and gives us the rate of change in direction relative to the door.

$h_0$ : initial heading,  $h$ : current heading,  $t_0$ : initial time,  $t$ : current time

$$\omega = \frac{h - h_0}{t - t_0}$$

## 3.3 Reasoning

The computer vision provides the data needed to reason about the behaviour observed. As described in Section 2.3 we have several alternatives to reasoning methods. Solem (2010) suggested *rule-based reasoning* as a well suited mechanism for inferring intentions. This method however is very static, and dependent on a set of rules that may have to be changed for different environments.

Further research pointed at *hidden markov models* as a potential improvement on the *rule-based* approach. HMMs have been used extensively in the fields of gesture and speech recognition (Tanguay Jr, 1995; Juang and Rabiner, 1991), as well as human behavioural detection (Uddin and Kim, 2011).

The framework by Solem (2010) can provide the data to test, and if applicable, train and implement a reasoning mechanism using HMMs.

### 3.3.1 Hidden Markov Model

A HMM can be seen as a variant of a *finite state machine*. It has a set of hidden states denoted by  $Q$ . An output *alphabet* or observations denoted by  $O$ . Transition and output probabilities expressed with  $A$  and  $B$  respectively, and last, the initial state probabilities denoted by  $\Pi$ . The definitions of these properties can be seen in Equation 3.1, 3.2, 3.3, 3.4, 3.5.

$$Q = \{q_i\}, i = 1, \dots, N \quad (3.1)$$

$$A = \{a_{ij} = P(q_j \text{ at } t + 1 | q_j \text{ at } t)\} \quad (3.2)$$

$$B = \{b_{ik} = b_i(o_k) = P(o_k | q_j)\} \quad (3.3)$$

$$O = \{o_k\}, k = 1, \dots, M \quad (3.4)$$

$$\Pi = \{p_i = P(q_j \text{ at } t = 1)\} \quad (3.5)$$

There are three tasks to solve in the use of HMMs. One has to calculate the probability of a certain output sequence, i.e. find the probability for a sequence of observations belonging to the particular HMM. This task is done with the forward-backward algorithm Viterbi. Next one

has to find the most likely chain of hidden states that could generate a given observation chain. This too is solved by using the Viterbi algorithm. Last, one has to calculate the most likely set of state transitions and output probabilities. This last problem is solved using the Baum-Welch algorithm (Russell and Norvig, 2010).

### 3.3.2 Setting up the Model

The HMM must operate on one or more of the features described in Section 3.1.1. To do this we have to adapt the feature output from the framework by Solem (2010) to be able to use it. From the framework we get the raw data for each of the features for each time step as can be seen in Figure 3.4.

To be able to put this data into a HMM we have to combine all variables into one variable as described in Section 2.3.3. In the case of location data, where we originally have two variables<sup>3</sup> in the form of x and z coordinates, we have to combine these into one variable. To do this we divide the area in front of the door into a grid with each cell given a number as can be seen in Figure 3.3. This will give us the path of a person as a number of discrete observations (e.g. 10→18→19→27→35→43→51→59). This can also be seen as a form of smoothing, as we decrease the resolution of the data, or the *alphabet* to 64 possible observations.

If we are to add further features to the model we have to multiply the number of possible values of each of the features to get the number of possible discrete observations. After gathering a lot of data from interactions with the door, we can train a HMM using this data and an example of this training can be seen in Figure 3.5.

### 3.3.3 Data Analysis

An example output of location data can be seen in Figure 3.4. The data output from the feature extraction proved to be very prone to noise. Figure 3.6 is an example output from the body alignment feature. This proved to be the problem in all features but the location data, and reasons for this are discussed further in Chapter 5. Because of this, the pure location of a person was chosen as the feature to learn from and predict.

As described in Section 3.3.2, to further smooth the data from the location feature, the location was put into a 8x8 grid as seen in Figure 3.3. This gave us a series of discrete observations from each of the interactions with the door. From all of this data a HMM could be trained.

Before we use this trained HMM we must first lay down some rules that will be triggered to make the door open, i.e. the door will have to make an absolute decision when the person is a certain distance away in order to open in time. If we combine Figure 3.1 with Figure 3.3, we find that some of the grid cells are inside the area defined as *close* to the door. From this we define a rule that says if the next predicted state places the person inside this area the door shall open as it is most probable that the person has the intention of walking through.

From this we can see that if a person decides to change his mind inside this *close* area, we are passed the point of no return, and the door will open no matter what. This has to do with limitations in the form of accuracy in the data as well as mechanical limitations and is further discussed in Chapter 5.

---

<sup>3</sup>We ignore the y axis as we are only interested in the motion in the x and y plane.

1	2	3	...				

Figure 3.3: The location grid in front of the door. 8x8 cells with the door located at the bottom edge.

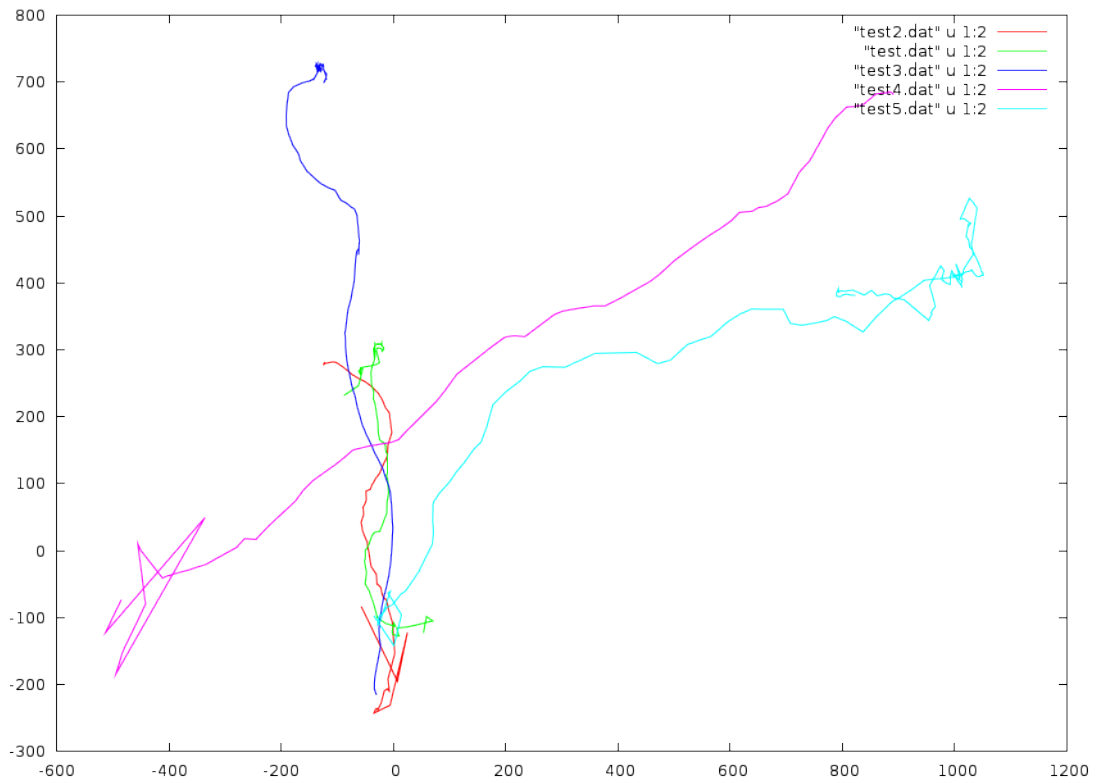


Figure 3.4: An example of the output of location data based on detected coordinates. The door is at the bottom here, and this is an example of four people walking through the door, and one person not going through.



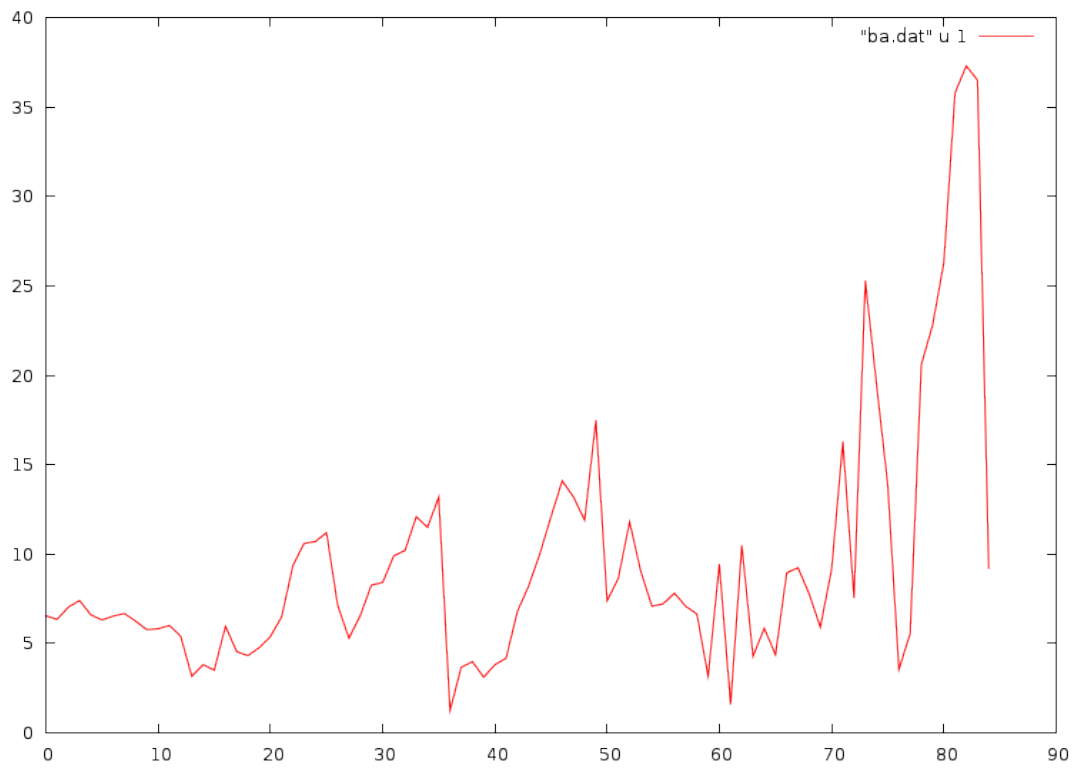


Figure 3.6: An example of the noisy data generated from the feature extraction, in this case the body alignment feature in a particular case.

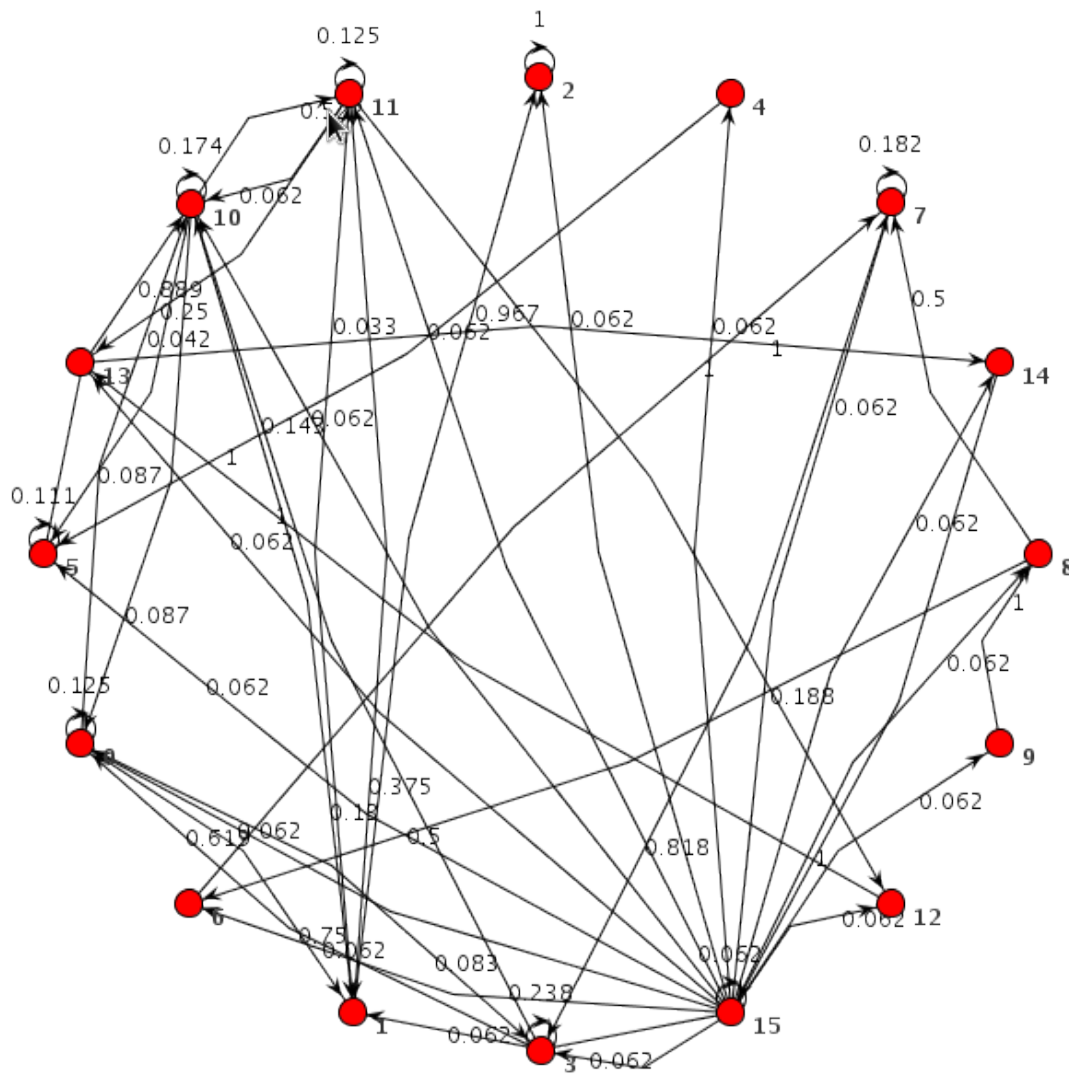


Figure 3.7: A 16 state, trained *hidden markov model* defining most probable state changes in front of the door.

### 3.4 A Sliding Door

Work done by Solem (2010) provided hardware to test on. The door can be seen in Figure 3.8. Solem (2010) made an application framework for operating the door which had to be modified in order to use it with HMM. This door has some limitations discussed further in Chapter 5.



Figure 3.8: The automatic sliding door built by Solem (2010)

### 3.5 Application

After modifying the framework by Solem (2010) to output sequences of data containing the features described in Section 3.1.1 and described further in Section 3.3.3, I chose to develop an application in the Java programming language. This was chosen because of ease of programming, and availability of good libraries. The application was built around Jahmm<sup>4</sup> in order to analyze and test the data.

---

<sup>4</sup>A library for training and testing hidden markov models



### 3.5.1 Application Structure

A general description of the structure of the application can be seen in Figure 3.9. This is of course the structure of the components working after we are done getting and applying the training data. The structure for training the system can be seen in Figure 3.10.

This application runs like a loop and keeps track of people and their respective paths in the grid. For each change in grid location the path of the person is updated. On each update of location a look-up is made to see if there is a high probability of the next state being within the threshold for opening the door. If the probability is high, the door opens, or else the location is just updated.

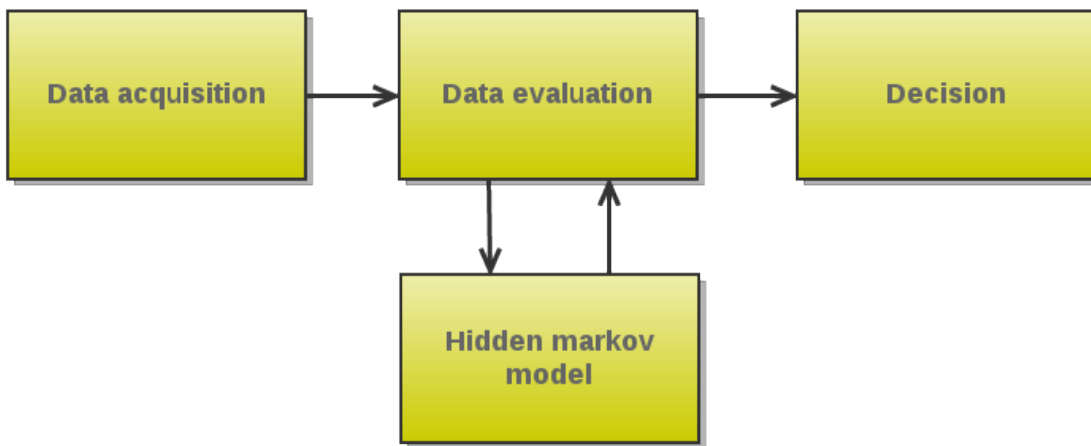


Figure 3.9: A diagram showing the structure of the application.

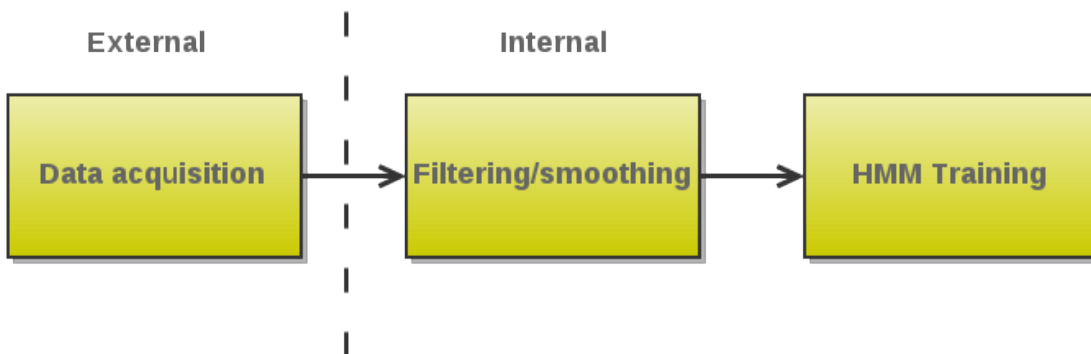


Figure 3.10: A diagram showing the structure of the training application.



# Chapter 4

## Evaluation

In order to test the usage of HMMs as a reasoning mechanism, we need a way of measuring how well it performs and why. To do this we can compare it to existing solutions and do tests under controlled conditions.

Solem (2010) proposed the usage of manuscripts for interacting with the door. This will give clear defined sets of data that we know should lead to the door opening or not. The raw data from these sets may also be tested to find out if the reasoning problem is solvable using HMMs before moving on to actually using it. Further, these manuscripts will let the results from this testing be compared to the original results obtained by Solem (2010). The manuscripts describe how a person should interact with the door, and the intended goal of the interaction. The manuscripts can be seen in Appendix A. Each manuscript was played out several times in order to get a good sample size to analyze.

### 4.1 Testing

The testing had to be done in three successive stages. First, all the manuscripts would have to be played out to first analyze the data to see if HMMs could be applied. After the first stage, if successful, the data would be used to train a *hidden markov model*. Last, a new round of playing out manuscripts to test against the trained reasoning engine and evaluate the final results.

Each of the 15 manuscripts were played out 20 times, resulting in 300 sets of data labeled either *open* or *not open* denoting the intended action of the manuscript. This data was then used to train the HMM, and the final trained HMM can be seen in Figure 3.7. Several combinations of number of states and observations were tried, and it was settled on a 16 state HMM with an *alphabet* with a total of 64 values.

After the HMM was done and trained, a new round of 20 runs for each of the manuscripts were done, and the results of this can be seen in Table 4.1.

#### 4.1.1 Performance Measure

The tests give us result states with two possible outputs, namely *correct* or *incorrect*. In a case like this, it is most common to use the following terms:

**True positive (TP)** The door opened when the user intended to walk through.

**False positive (FP)** The door opened when the user did not intend to walk through.

Table 4.1: Test results

Class	Manus.	Intention	Opened	-Opened	Total
Simple	1	Positive	20	0	20
	2	Positive	17	3	20
	3	Positive	7	13	20
	4	Negative	0	20	20
	5	Negative	3	17	20
	<b>SUM</b>			<b>47</b>	<b>53</b>
Intermediate	6	Positive	19	1	20
	7	Positive	16	4	20
	8	Positive	17	3	20
	9	Positive	17	3	20
	10	Negative	3	17	20
	11	Negative	4	16	20
	<b>SUM</b>			<b>76</b>	<b>44</b>
Complicated	12	Positive	17	3	20
	13	Positive	18	2	20
	14	Positive	10	10	20
	15	Negative	12	8	20
	<b>SUM</b>			<b>57</b>	<b>23</b>
<b>TOTAL</b>			<b>180</b>	<b>120</b>	<b>300</b>

**True negative (TN)** The door stayed closed when the user did not intend to walk through.

**False negative (FN)** The door stayed closed when the user intended to walk through.

The results can also be seen in Table 4.2 grouped by these classifications.

Table 4.2: Rates of the different result classes

Class	Manus.	TP	TN	FP	FN
Simple	1	20	-	-	0
	2	17	-	-	3
	3	7	-	-	13
	4	-	20	0	-
	5	-	17	3	-
	<b>SUM</b>	<b>44</b>	<b>37</b>	<b>3</b>	<b>16</b>
Intermediate	6	19	-	-	1
	7	16	-	-	4
	8	17	-	-	3
	9	17	-	-	3
	10	-	17	3	-
	11	-	16	4	-
	<b>SUM</b>	<b>69</b>	<b>33</b>	<b>7</b>	<b>11</b>
Complicated	12	17	-	-	3
	13	18	-	-	2
	14	10	-	-	10
	15	-	8	12	-
	<b>SUM</b>	<b>45</b>	<b>8</b>	<b>12</b>	<b>15</b>
<b>TOTAL</b>	<b>158</b>	<b>78</b>	<b>22</b>	<b>42</b>	

These different classifications can be used to evaluate different statistical measures relevant to determining the performance of the door (Guda et al., 2004). This will let us compare the performance to the original work by Solem (2010) as well as traditional sliding doors.

**Sensitivity (ST)** This value tells us the ability to identify positive results.

$$ST = \frac{TP}{TP + FN} \quad (4.1)$$

**Specificity (SP)** This value tells us the ability to identify negative results.

$$SP = \frac{TN}{TN + FP} \quad (4.2)$$

**Positive predictive value (PPV)** This value tells us how accurate the system is at recognizing positive intentions.

$$PPV = \frac{TP}{TP + FP} \quad (4.3)$$

**Negative predictive value (NPV)** This tells us how accurate the system is at recognizing negative intentions.

$$NPV = \frac{TN}{TN + FN} \quad (4.4)$$

**Accuracy (A)** The rate of true results compared to all the results.

$$A = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.5)$$

The statistical measures give us an indication of how good the door is at identifying the correct intention. The results can be seen in Table 4.3. Equal to the findings by Solem (2010) the different difficulties of the manuscripts were almost correct, meaning that from the system's standpoint, classes *simple* and *intermediate* were simplest and class *complicated* was most difficult to handle.

Table 4.3: Statistical measures for results in percent.

Class	Sensitivity	Specificity	PPV	NPV	Accuracy
Simple	73.3	92.5	93.6	69.8	81
Intermediate	86.3	82.5	90.8	75	85
Complicated	75	40	78.9	34.8	66.3
<b>TOTAL</b>	<b>79</b>	<b>78</b>	<b>87.8</b>	<b>65</b>	<b>78.7</b>

From Table 4.3 we see that the accuracy for class *complicated* is as low as 66.3 %. If we compare this to the traditional sliding door as seen in Table 4.4, we see that the accuracy of the intelligent sliding door is actually lower. This does not mean that the traditional door is better globally, but in that particular composition of cases. This can be explained by looking at the statistical measures. A traditional sliding door will always open if someone moves in its field of view. This means it can not measure negative intentions. The equations then collapse to only give a percent value of the ratio of positive and negative tests.

After all the testing is done, the door gets an accuracy of 78.7 % meaning it reads the user's intention correctly close to 4 out of 5 times. Like Solem (2010), the accuracy suffers because of the high rate of false negatives. As can be seen in Table 4.5, the accuracy is marginally better than the results obtained by Solem (2010). We also see that the sensitivity is higher, meaning that the door is opening when it should at a higher rate. The specificity is lower though, meaning that the door is opening too much when it shouldn't.

As we have a different sample size of positive and negative intention cases we do not get a good comparison with traditional doors in regards to accuracy. To address this we calculate the mean square contingency coefficient or the phi coefficient of the 2x2 contingency table of possible cases, i.e. TP, TN, FP and FN. The coefficient is given by the equation:

Table 4.4: Statistical measures for traditional sliding door (modified from Solem (2010)).

Class	Sensitivity	Specificity	PPV	NPV	Accuracy
Simple	100	0	60	0	60
Intermediate	100	0	66.7	0	66.7
Complicated	100	0	75	0	75
<b>TOTAL</b>	<b>100</b>	<b>0</b>	<b>66.7</b>	<b>0</b>	<b>66.7</b>

Table 4.5: Comparing performances of different doors.

Door	Sensitivity	Specificity	PPV	NPV	Accuracy
Intelligent door	79	78	87.8	65	78.7
Solem (2010) door	70.4	90	92.6	63	77.4
Traditional door	100	0	66.7	0	66.7

$$\phi^2 = \frac{X^2}{n} \quad (4.6)$$

Where  $n$  denotes the number of cases. This value will tell us how good the door's performance is compared to the real intentions of the people doing the tests. It ranges from -1 to 1, meaning total disagreement and total agreement respectively.

From Table 4.6 we see that the traditional sliding doors have a 0 phi coefficient meaning it is equal to a random prediction. In contrast we see that the intelligent door has 0.55 compared to the results found by Solem (2010) at 0.58. From this we see that the my proposed solution was a little less able to predict the true intention of people.

Table 4.6: Comparing phi coefficient.

Door	$\phi$
Intelligent door	0.55
Solem (2010) door	0.58
Traditional door	0





## Chapter 5

# Conclusion and Future Work

This project has made an automatic sliding door more intelligent by better understanding human intentions.

We have taken the raw data from a captured image stream and extracted the features of interest. In the context of the chosen decision mechanism, we have evaluated the feature extracted data to find the usable features. We have further adapted the output to make it fit into the framework of *hidden markov models*. A large set of data samples were gathered using manuscripts and fed into the learning algorithm of the HMM to generate the state and transition probabilities. Using the trained HMM, we ran all the manuscripts again, and evaluated the performance of the door using standard statistical tools.

### 5.1 Discussion

Through the results found in this project we see that doors can be made more intelligent. By this is meant that the door can be made better at understanding the intention of humans interacting with it.

From the results we can see some things of interest. A traditional door will always open if there are people moving within its field of view. This means that for all cases where the person did not intent to walk through the door, the door will still open, creating a false positive. From this we understand that the door will always open when a person has the intention of walking through<sup>1</sup> and always makes mistakes when the person has negative intentions. This means that as it now stands, a traditional door may be better in certain environments. If the door is placed at a location with almost exclusively interactions with positive intentions, a traditional door may actually perform better.

This project has focused on understanding a person's intention when interacting with a door. In doing so we have sacrificed the perfect positive intention reading of the traditional door with the lower ability of the intelligent door at doing the same. This sacrifice however, has made us able to better read negative intentions. One can argue if this is worth it. If we look at the different manuscripts we see that the door is very good at predicting the correct intention in the simplest cases.

The setup used in this project have some limitations that affected the results and design decisions. In several of the test cases the system was not able to detect the person in time or at all. These cases were then of course registered as false positives or true negatives and if removed

---

<sup>1</sup>In a best case scenario.

from the calculations would present a better result. In addition to the difficulties in detecting people, the data extracted was very prone to noise.

The decision to only use location data was made because of this problem with noisy data. There would have been problems in using the data extracted from the e.g. body alignment in an HMM as this would only teach the HMM the small movements made to generate the motion on a bigger scale. By this we mean the subtle changes in the features as a person moves e.g. one leg instead of the general motion in walking.

The location data was further reduced in complexity by dividing the area in front of the door into cells with designated numbers to be able to make it work with the HMM. This however did not preserve any good data about the motion, only current state or path and probable next state. In spite of this it worked very well.

## 5.2 Future Work

This project proves that it is possible to better understand human intentions using *hidden markov models*. In the previous discussion we looked at the relationship between traditional doors and the new intelligent door. From this we see that it could be interesting to change the view of when to open and not to preserve the true positives. By this we mean that the door should open unless there is a negative intention predicted instead of the other way around like it was made in this project.

It could prove beneficial to combine the findings in this project with the findings made by Solem (2010) to make a more robust reasoning mechanism. It could also be good to look into filtering and smoothing of the feature data to learn from the general motion instead of the small motions in between.

There are news about a new Kinect sensor that is more sensitive and accurate that, when it arrives, could be useful in getting more accurate and less noisy feature data.

The findings in this project could be transferred to other areas like pedestrian movement and understanding. This could be very beneficial in the case of pedestrian crossings to better handle the lights in the crossing. This could possibly ease traffic congestion and make the crossings safer by being able to adapt to both the traffic and person trying to cross.

# Bibliography

- Aamodt, A. and Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1):39–59.
- Amano, M., Hironaka, D., Oda, S., Capi, G., and Yokota, M. (2005). Linguistic interpretation of human motion based on mental image directed semantic theory. In *Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on*, volume 1, pages 139–144. IEEE.
- Caillette, F. and Howard, T. (2004). Real-time markerless human body tracking using colored voxels and 3d blobs. In *Mixed and Augmented Reality, 2004. ISMAR 2004. Third IEEE and ACM International Symposium on*, pages 266–267. IEEE.
- Corazza, S., Mündermann, L., and Andriacchi, T. (2007). A framework for the functional identification of joint centers using markerless motion capture, validation for the hip joint. *Journal of biomechanics*, 40(15):3510–3515.
- Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. Ieee.
- Giosan, I., Nedeveschi, S., and Bota, S. (2009). Real time stereo vision based pedestrian detection using full body contours. In *Intelligent Computer Communication and Processing, 2009. ICCP 2009. IEEE 5th International Conference on*, pages 79–86. IEEE.
- Guda, C., Fahy, E., and Subramaniam, S. (2004). Mitopred: a genome-scale method for prediction of nucleus-encoded mitochondrial proteins. *Bioinformatics*, 20(11):1785–1794.
- Guerra-Filho, G. and Aloimonos, Y. (2006). A sensory-motor language for human activity understanding. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 69–75. IEEE.
- Horaud, R., Niskanen, M., Dewaele, G., and Boyer, E. (2009). Human motion tracking by registering an articulated surface to 3d points and normals. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(1):158–163.
- Jia, H. and Zhang, Y. (2007). Fast human detection by boosting histograms of oriented gradients. In *Image and Graphics, 2007. ICIG 2007. Fourth International Conference on*, pages 683–688. IEEE.
- Juang, B. and Rabiner, L. (1991). Hidden markov models for speech recognition. *Technometrics*, pages 251–272.

- Kofod-Petersen, A., W. R. and Cassens, J. (2009). Closed doors—modelling intention in behavioural interfaces. pages 93–102. Tapir Akademisk Forlag.
- Martinec, R. (2001). Interpersonal resources in action. *Semiotica*, 2001(135):117–145.
- Moore, A. (2008). Surgical teams in action: a contextually sensitive approach to modelling body alignment and interpersonal engagement. *Interdisciplinary Perspectives on Multimodality: Theory and Practice, Campobasso, Italy*.
- Oral, M. and Deniz, U. (2007). Centre of mass model—a novel approach to background modelling for segmentation of moving objects. *Image and Vision Computing*, 25(8):1365–1376.
- Russell, S. and Norvig, P. (2010). *Artificial intelligence: a modern approach*. Prentice hall.
- Solem, J. (2010). Intention-aware sliding doors.
- Sonka, M., Hlavac, V., and Boyle, R. (2008). Image processing, analysis, and machine vision.
- Tanguay Jr, D. (1995). *Hidden Markov models for gesture recognition*. PhD thesis, Massachusetts Institute of Technology.
- Uddin, M. and Kim, T. (2011). Continuous hidden markov models for depth map-based human activity recognition. *Hidden Markov Models, Theory and Applications*, pages 225–247.
- Zhu, Q., Yeh, M., Cheng, K., and Avidan, S. (2006). Fast human detection using a cascade of histograms of oriented gradients. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1491–1498. IEEE.

# Appendix A

## Manuscripts

The manuscripts are based on work done by Solem (2010), and provides a systematic way of testing the sliding door. The manuscripts cover both the intention of walking through the door and not, and give instructions on how to accomplish this task. Each manuscript contains the following parameters:

**Category** The difficulty of the case for the system.

**Actors** The number of persons in the case.

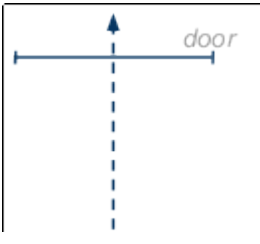
**Intention** The intended goal of the interaction.

**Speed** The walking speed.

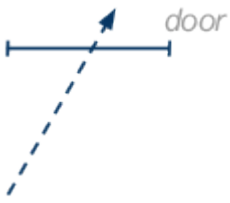
**Acceleration** Rate of change in speed of the person.

**Path** The persons heading.

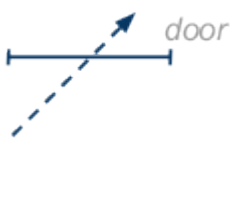
**Description** Detailed description.

Manuscript 1			
	Category	Actors	Intention
	Simple	1	Positive
	Speed	Acceleration	Path
	Normal	None	Straight

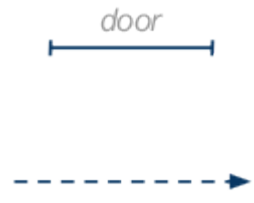
**Description:** A person walks with normal speed straight towards the door, intending to walk through it.

Manuscript 2			
	Category	Actors	Intention
	Simple	1	Positive
	Speed	Acceleration	Path
	Normal	None	Straight

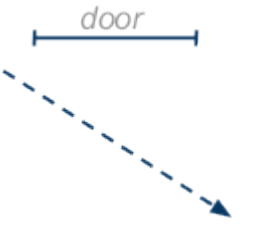
**Description:** A person is standing at the side of the scene. He faces the door and heads straight towards the door, intending to walk through it. The incoming path is straight but angled about  $30^\circ$  off the z-axis.

Manuscript 3			
	Category	Actors	Intention
	Simple	1	Positive
	Speed	Acceleration	Path
	Normal	None	Straight

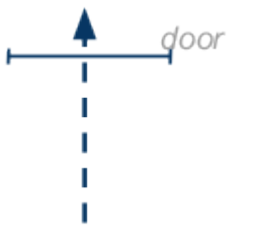
**Description:** A person is standing at the side of the scene. He faces the door and heads straight towards the door, intending to walk through it. The incoming path is straight but angled about  $45^\circ$  off the z-axis.

Manuscript 4			
	Category	Actors	Intention
	Simple	1	Negative
	Speed	Acceleration	Path
	Normal	None	Straight


**Description:** A person is standing at the side of the scene. He walks in a straight line in parallel with the door. Distance between the door and the person is about 2 meters.

Manuscript 5			
	Category	Actors	Intention
	Simple	1	Negative
	Speed	Acceleration	Path
	Normal	None	Straight

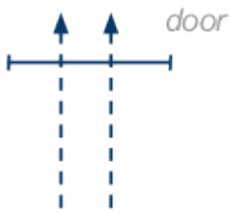
**Description:** A person is standing next to the door. He walks diagonally across the scene, away from the door.

Manuscript 6			
	Category	Actors	Intention
	Intermediate	1	Positive
	Speed	Acceleration	Path
	Fast	None	Straight

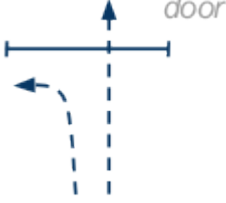
**Description:** A person walks quickly straight towards the door, intending to walk through it.

Manuscript 7			
	Category	Actors	Intention
	Intermediate	1	Positive
	Speed	Acceleration	Path
	Normal	Changing	Changing


**Description:** A person walks parallel to the door and is suddenly told to walk through the door, simulating changing one's mind.

Manuscript 8			
	Category	Actors	Intention
	Intermediate	2	Positive
	Speed	Acceleration	Path
	Normal	None	Straight

**Description:** A pair walks together in a straight line, at normal speed, along the z-axis, both intending to walk through the door.

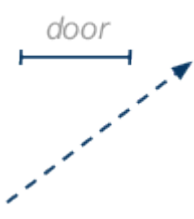
Manuscript 9			
	Category	Actors	Intention
	Intermediate	2	Positive
	Speed	Acceleration	Path
	Normal	Mixed	Mixed

**Description:** A pair walks together in a straight line, at normal speed, along the z-axis, both intending to walk through the door. One person is then told to exit the scene to the side.

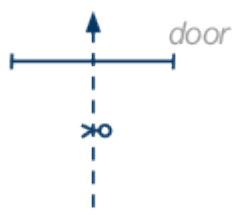
Manuscript 10			
	Category	Actors	Intention
	Intermediate	2	Negative
	Speed	Acceleration	Path
	None	None	Standing

**Description:** Two people are standing in front of the door, facing each other. They are not moving in any direction. They are talking together, gesticulating and moving at the spot. This scenario simulates a casual talk in front of a door.

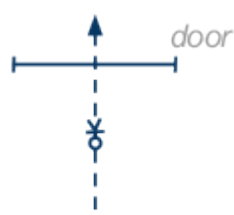


Manuscript 11			
	Category	Actors	Intention
	Intermediate	1	Negative
	Speed	Acceleration	Path
	Normal	None	Straight


**Description:** A person is standing at the side of the scene. He walks in a straight line, diagonally across the scene, aiming at the opposite side of the door, at a point about 1 m away from the door.

Manuscript 12			
	Category	Actors	Intention
	Complicated	1	Positive
	Speed	Acceleration	Path
	Normal	None	Straight

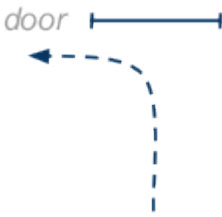
**Description:** A person walks sideways in a straight line, along the z-axis, with the intention of walking through the door.

Manuscript 13			
	Category	Actors	Intention
	Complicated	1	Positive
	Speed	Acceleration	Path
	Normal	None	Straight

**Description:** A person walks backwards in a straight line, along the z-axis, with the intention of walking through the door.

Manuscript 14			
	Category	Actors	Intention
	Complicated	1	Positive
	Speed	Acceleration	Path
	Low	None	Changing

**Description:** A person is standing close to the door. He is standing with his back to the door. He wants to go back inside again, and turns around.

Manuscript 15			
	Category	Actors	Intention
	Complicated	1	Negative
	Speed	Acceleration	Path
	Normal	Changing	Changing

**Description:** A person is told to walk through the door. Walking along the z-axis, he is suddenly ordered to turn left and go to the side.