NTNU
Norwegian University of
Science and Technology

Intention-aware Sliding Doors

John Sverre Solem

Master of Science in Computer Science
Submission date: June 2011
Supervisor: Anders Kofod-Petersen, IDI
Co-supervisor: Richard Blake, IDI
Rebekah Wegener, Macquarie University, NSW, Australia

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

Problem Description

Truly smart systems need to interface with the behavior 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: 17. January 2011, 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. Ignoring the user's intention, can result in a miscommunication between the door and the user, most often leading to erroneous openings of the door. I try to solve this problem by using a Kinect sensor capturing human activity in front of a door. The users are then detected and skeletal joints tracked using the OpenNI framework. Features are extracted according to a model of human behavior and intentions. A rule-based reasoning mechanism then makes a decision whether to open the door or not. In this project I have generalized a door user's behavior creating a model of symbols and events. I have implemented a program that can identify these events and operate a door based on the inferred intention. I have also built a door that is able to demonstrate the functionality of the program. The intention-aware, intelligent sliding door achieved an accuracy of 77-86 % in the performed test cases.

Keywords: Computer Vision, Artificial Intelligence, Intention recognition, Behavioral model, Rule-based reasoning, Intention-awareness, Sliding door, Xbox Kinect, OpenNI.

Preface

This master's thesis describes the study and work from my master's project. The project is the closure of my degree in Computer Science at the Department of Computer and Information Science at the Norwegian University of Science and Technology. The project is a continuation of a specialization project, performed by a fellow student, Håvar Aambø Fosstveit, and I. Some of the material presented in this thesis will therefore originate from the specialization project report. Section 2.2 is in special written together with Håvar Aambø Fosstveit. In connection with a submission to SCAI, Richard Blake also contributed in rewriting parts of this section.

Acknowledgements

This project would not be possible without the continuous help and support from my supervisor, Anders Kofod-Petersen. I would like to thank him for his invaluable advice and assistance throughout the entire project. I would also extend my gratitude to my co-supervisors; Richard Blake for many encouraging meetings and help in computer vision, and Rebekah Wegener for supplying me with resources and insight within the area of semiotics. Finally, I would like to thank my lovely wife, for standing by my side, and supporting me.

John Sverre Solem
Trondheim, June 20, 2011

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Overview

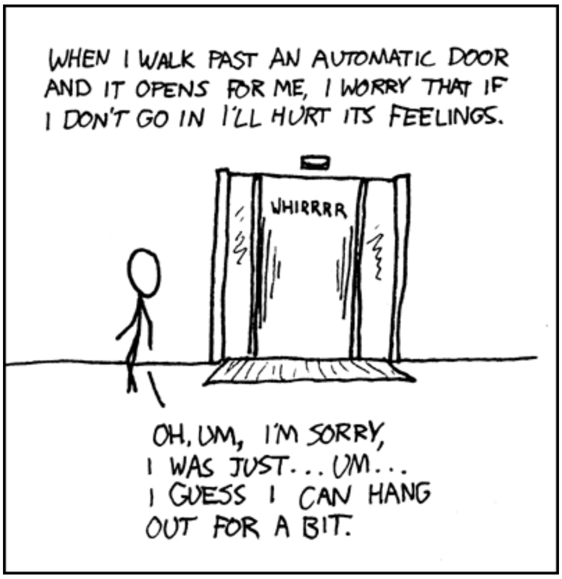


Figure 1.1: Automatic doors (`http://xkcd.com/175`)

In the following sections the background and motivation for this project is described. I define the goals and research questions for the work. I also provide a description of the research method used in the project work. Finally I present an overview of the report structure; the chapters and their content.

1.1 Background and Motivation

The background for this project is my master's project within the field of Artificial Intelligence at the Norwegian University of Science and Technology.

The main motivation for the work is given in the article by Kofod-Petersen et al. (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. Although the sliding doors are not the main concern, they point out the miscommunication between the two actors, human and computer. Sliding doors provide a simple case with a limited set of actions and make a great starting point for further work.

Can an automated sliding door respond to the user's intention? This challenge touches multiple areas: It requires a complete domain model of human behavior, describing the user's movement. Further, it requires data collection and feature extraction together with an inference mechanism for intentions.

Reasoning over intentions based on behavior requires work on the symbolic level, rather than the more common sub-symbolic level. Lifting the process to this level generalizes the process, giving a result that can be transferred more easily to other similar tasks.

Another motivational factor for this thesis is the work that was done in my specialization project. This project laid the foundations, by exploring the possibilities within the challenge. I now take the previous results further, and try to solve this task completely – with a functional door. Building an intelligent sliding door that might actually work, is in itself a motivation. This factor also includes the motivation for working with an actual door instead of simulating one. Using a real door alleviates some of the issues that arise when simulating human behavior. In order to evaluate a solution it is important to work with behavior that is as close to reality as possible, showing true, unaffected intentions.

1.2 Goals and Research Questions

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

Define a set of features needed in order to describe human behavior 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 AI[1] 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 draw conclusions regarding the intentions of the person interacting with the door.

Goal 4 Implement software comprising the results from Goal 1, 2 and 3

Develop a complete software application for the operation of a door equipped with sensors giving it the ability to reason.

Goal 5 Build a motorized sliding door

Build a door that can be operated by a computer running the developed software.

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

Evaluate 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?

Evaluate 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 Contributions

The 11th Scandinavian Conference on Artificial Intelligence was held during my thesis work. I participated in this conference with a *poster* and a *demonstration* of the door. The poster can be seen in Appendix D.

1.4 Research Method

The work done in this project can be divided into seven 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 be included and what would not be included. I had meetings with my supervisor Anders Kofod-Petersen, where we discussed the contents of the work in context of the specialization project and the

Table 1.1: Time budget listing main activities

Week	Date	Activity
3	17. Jan	Planning
4	24. Jan	”
5	31. Jan	Build door
6	7. Feb	”
7	14. Feb	”
8	21. Feb	Computer Vision
9	28. Feb	”
10	7. Mar	”
11	14. Mar	”
12	21. Mar	Reasoning/Decision
13	28. Mar	”
14	4. Apr	”
15	11. Apr	Testing
16	18. Apr	”
17	25. Apr	Easter vacation
18	2. May	Testing
19	9. May	Report writing
20	16. May	”
21	23. May	”
22	30. May	”
23	6. June	Extra
24	13. June	Extra (application needed).

previous work done through that project. I worked out a time budget, giving a rough overview of main activities (see Table 1.1).

During the project, I revised the time budget several times, as additional activities were needed and some of the activities took longer than initially planned.

Literature survey This stage was mainly performed in the specialization project, where we did a research on similar problems, existing works and projects. For this purpose we 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. We ran different combinations of these search terms in several digital libraries (listed in Table 1.3). Different combinations gave different levels of quality and relevancy in search results.

Table 1.2: Search terms used in literature survey

Sensor	Computer Vision	Model/Reasoning	Human Behavior
stereo vision camera motion sensor sensor fusion	stereo vision motion detection segmentation kalman filter facial recognition vector proximity marker-less motion tracking	knowledge base learning reasoning retrieval decision semiotics syntax semantic	intention movement anatomy body language posture hip pose body alignment gaze gaze direction human behavior
kinect	hog histogram of oriented gradients gpu body tracking		joint

Table 1.3: Search engines used for literature survey

Search engine	URL
IEEE Xplore	`http://ieeexplore.ieee.org`
SpringerLink	`http://www.springerlink.com`
ISI Web of Knowledge	`http://www.isiknowledge.com`
ScienceDirect	`http://www.sciencedirect.com`

To determine the relevancy of the resources, we looked at the search results as follows:

[1]AI is an abbreviation for artificial intelligence.

1. *Title* – If irrelevant, discard. If relevant, go to step 2.
2. *Abstract* – If irrelevant, discard. If relevant, go to step 3.
3. *Overview* – Read quickly through the article, if irrelevant, discard. If relevant, go to step 4.
4. *Catalog* – Read through article, save in database, comment and rate according to relevancy and usefulness.

It was necessary to expand the table of search terms in order to update it to the focus areas of the master project. This can be seen in the lower row of the table.

Component research and evaluation This stage was performed for all of the main parts of the project: computer vision, modeling and reasoning. The research results from the specialization project narrowed down the search field, and this stage was now mainly about evaluating the previous findings as well as choosing the technologies and mechanisms to use.

Implementation This stage involved the production of the complete system. It was not a separate work stage, but continued throughout the span of the project.

Work that belongs to this stage includes:

- Building a sliding door
- Building a door controller
- Building an intention-aware door operator
  - Modeling human behavior
  - Programming a human tracking mechanism
  - Programming a reasoning mechanism

Testing and evaluation The door was tested using multiple test subjects, instructed to interact in various ways with the door. Since this stage included monitoring third party people using cameras, it was necessary to contact NSD concerning privacy and the handling of personal data (see Section 3.4.2, page 49). The collected data was evaluated using standard performance measures.

Thesis writing This process occurred throughout the time span of the project, but was concentrated mostly in the end. Here I documented the work that was done, together with theory and background. The written material from other stages was revised and structured.

Meetings I had regular contact with my supervisor. In these meetings I gave an overview of the work being done, and got feedback and input for the work ahead.

1.5 Thesis Structure

The goal of this master's thesis is to document the work done in my master's project. The reader is introduced to a problem concerning the interaction between human beings and non-human actors in their environment. I then proceed with a presentation of the theory and background required to get an understanding of this problem and point out a solution. Further I describe my proposed solution and evaluate it.

The thesis is structured in chapters. *Chapter 1 (this chapter)* gives an introduction to the project, describing the background, motivation and goals. It also explains the work methods. *Chapter 2* presents the context of the problem, and provides background and theory according to the different parts of the project. *Chapter 3* describes the results of my work. It explains decisions and choices, as well as giving details about each of the solution components. *Chapter 4* shows how the results were tested and evaluated. It also sums up the project with a discussion and summary. Finally I describe future work; things that can be improved and ideas for others to pick up. A *Bibliography* lists the sources referred throughout the thesis. In the back, *Appendices* provide details and material that were left out of the main contents.

Chapter 2

Theory and Background

The work in this project can be separated into different categories, touching several disciplines within computer science. The motivation lies within semiotics[1] and the need for understanding and modeling human behavior in the context of reading intentions. The background for this is explained in Section 2.1. A primary part of managing to read intentions is related to computer vision. This is the capture and interpretation of raw data, giving the computer eyes. Section 2.2 elaborates this part. Finally the collected information must be brought to life. Using the modeled knowledge about human behavior one can reason over the information and infer an intention. The background for the reasoning part is described in Section 2.3. Figure 2.1 illustrates how the different parts come together to form an *intention-aware, intelligent agent*.

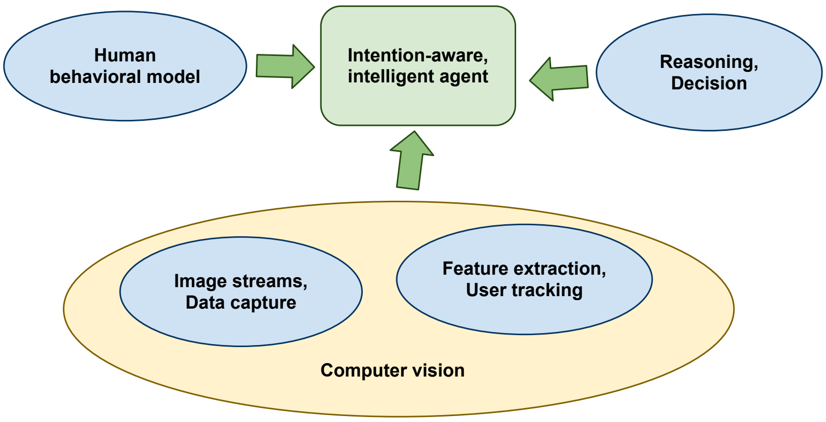


Figure 2.1: Different parts of an intelligent, intention-aware agent.

[1]The study of signs and symbols and their use or interpretation (Oxford American Dictionary).

2.1 Human Behavior and Intention

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 behavior can be complex, and sometimes even irrational. A person walking towards a door can suddenly change to standing still, reading the newspaper in the newspaper stand beside the door. Perhaps the stand 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 an 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 image stream.

The features that can be extracted from one single image stream are numerous. Adding an additional camera or sensor device gives even more possibilities. Some of the features are more suited than others, and Kofod-Petersen et al. (2009) suggest the use of body alignment, proximity and visual target as features of human behavior 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 et al. (2009) as shown in Figure 2.2.

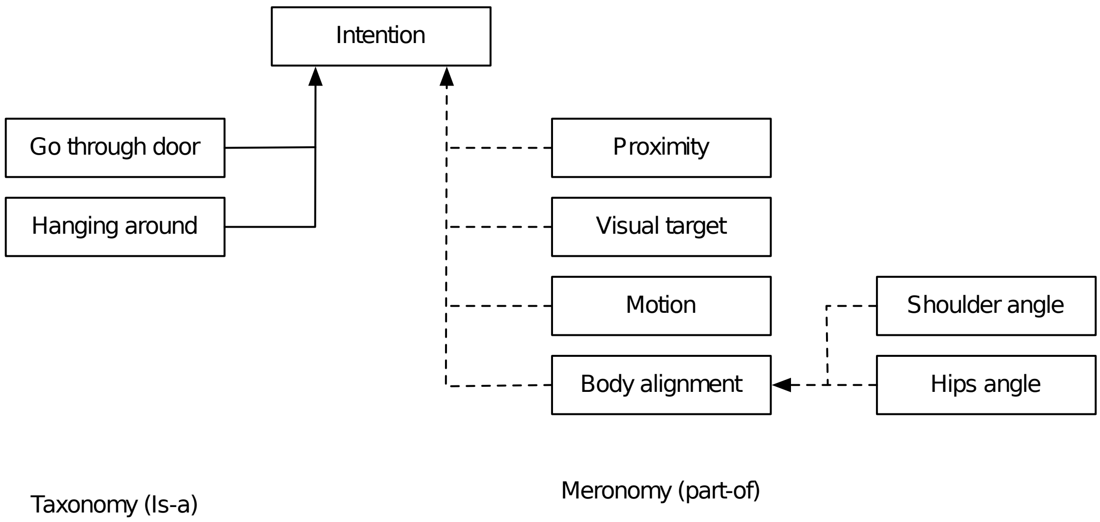


Figure 2.2: Features for modeling intention (from Kofod-Petersen et al., 2009)

The features described here are components of the perceived intention. If we have these features, we can derive a conclusion concerning the user's intentions.

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 extracting 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 simple tasks for humans, but 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 the works of Giosan et al. (2009) we can see the use of stereo vision in markerless 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) use 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 it 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*.

Another somewhat different approach to the human detection and tracking task is made by Viola and Jones (2001) where a rapid object detector is proposed using a boosted cascade of simple features. This solution uses Haar-like features and is specially useful for face detection. Further improvements to this has been done by Lienhart and Maydt (2002) strengthening rotational concerns.

Dalal and Triggs (2005) present a detector using Histograms of Oriented Gradients (HOG). It improves the accuracy of object detection, specially when it comes to pedestrians.

Further improvements have been suggested lately (Zhu et al., 2006; Jia and Zhang, 2007), trying to combine the discriminative power of HOG features (Dalal and Triggs, 2005) with the real-time performance of Viola's face detection framework (Viola and Jones, 2001). Work has also been done trying to boost the performance of the algorithm using the computer's GPU[2] (Lillywhite et al., 2009; Bilgic et al., 2010; Sugano et al., 2010).

[2]Graphics Processing Unit

2.2.1 Computer Vision Tools

Early studies pointed out *stereo vision* as a useful tool. This is basically due to the requirement of measuring distances to key points of the objects being tracked.

A stereo vision setup would use two cameras. They would be placed above the door, and be separated horizontally to give a base-line for the triangulation from which a distance can be calculated. The process, as explained by Sonka et al. (2008), is to find the coordinates in the two images of the "corresponding points". These are the two representations of easily identifiable structures in the real world such as corners found by the Moravec operator (Sonka et al., 2008).

The arrangement of the cameras, together with the disparity in the coordinate positions, allows the distance to be calculated. The equations are often over-determined and are solved using the SVD method[3], to give a least square error solution. The stereo vision is a module of the system that gives a distance image. This could be replaced by a different method if a suitable device becomes available.

There are several stages in the data collection pipeline. The first being the actual image capturing done with simple cameras. Second, the images must be segmented, separating the regions of interest from the rest. Following this, features have 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

Segmentation is the process of associating names from the outside world with collections of pixels. Segmentation is also a pattern recognition task: separating parts of the image that are of interest for further analysis from those that are not; often termed foreground and background. Segmentation is tuned to suit the goal of the system and is conditioned to tolerate data that is expected from the application domain.

The system for the sliding doors will receive image data that is rather unstable because of shadowing as people move about and automatic parameter changes in the camera. There is also a requirement to segment images to support an acceptable frame rate.

Edge based segmentation uses tools such as *Sobel masks*, giving thick edges, or *Canny edge detector*, giving unit width edges, to isolate intensity transitions in an image. Figure 2.3 shows the result of a Canny edge detection. The resulting edge image will be hard, and time consuming, to split into foreground/background.

Thresholding is a technique that is part of statistical decision theory. Essentially, a merit function is calculated and, depending on the value, one of two classes

[3] Singular Value Decomposition

Figure 2.3: Application of Canny

is chosen. The effectiveness of the method depends on the merit function and how much information it uses. The more complicated the function, then the slower the segmentation will run.

The simplest approach is to use the pixel value as the merit function and to choose a threshold value from the image statistics. Simple thresholding uses a threshold that is a local minimum in the image histogram. The choice is easiest when the histogram is clearly bi-modal, but this is no guarantee of good performance.

Preliminary experiments showed that images of people near the door contained a wide range of intensities and that a separation between foreground and background was probably going to be complicated.

Otsu's method for thresholding chooses a threshold that maximizes inter-class variance. It is calculated for each individual image from the image histogram. Preliminary experiments showed that Otsu's method did not reliably separate foreground from background as illustrated by Figure 2.4.

Anther useful tool is image subtraction, using more than one frame, in order to find the differences. There are several methods of doing image subtraction, the most basic 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 2.5. This method can be quite good in finding movements, but also for separating foreground from background. Oral and Deniz (2007) compare several image subtraction methods with the purpose of discovering movement.

As described initially, there are other alternatives to the segmentation problem. One of the choices is to use an object recognizer. This might be seen as a bridge between the segmentation and feature extraction process. By using a trained HOG descriptor (explained in detail in the following section) the system can detect people in a given frame. This will provide bounding boxes in which there ideally will be one person. The focus can then be on these boxes for further analysis.

Figure 2.4: Application of Otsu: From left to right; original, gray scale and Otsu applied.

Figure 2.5: Application of image subtraction: From left to right; background, background+foreground and difference mask

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 2.6. The algorithm as described by Dalal and Triggs (2005), uses several steps to compare images to the HOG descriptors (see Figure 2.7). 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. 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.This measure is used to normalize all the cells in the block. It is these normalized descriptor blocks that are called *Histogram of Oriented Gradient* descriptors.

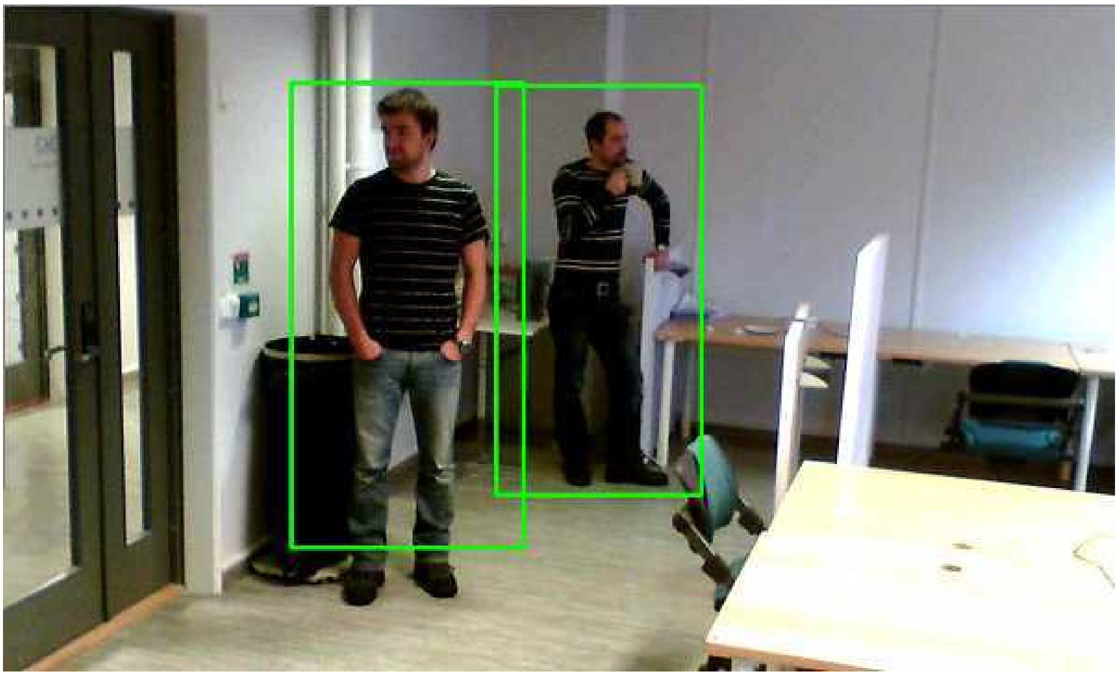


Figure 2.6: Application of HOG descriptor

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

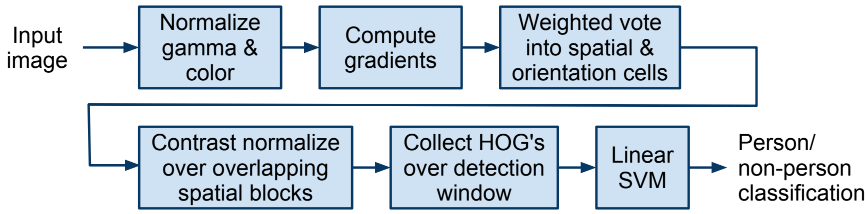


Figure 2.7: The steps from left to right in the HOG descriptor algorithm (Dalal and Triggs, 2005)

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) do 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 body. 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 person in the images.

In this project, the information computed in a segmentation step is not sufficient to give a complete system. Without feature extraction it would not be possible to provide the features required by the model of behavioral features discussed by Kofod-Petersen et al. (2009) for inferring human intentions. The reasoning process is dependent on these features, and as a result feature extraction is necessary for making a decision through reasoning.

The next step is then to utilize the 3D range model retrieved from the stereo vision joined with the detection rectangles from the HOG detector. A skeletal model can then be fitted in order to locate the points of interest in 3D space. This will give the measures needed for the feature model. Figure 2.8 shows how this overlay can provide the points of interest.

2.3 Reasoning

When the wanted feature data has been collected, there is still some work to be done. The door must either be given a command *open*, or simply ignore the activities in front of the door. Reasoning over the collected feature data makes it possible 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

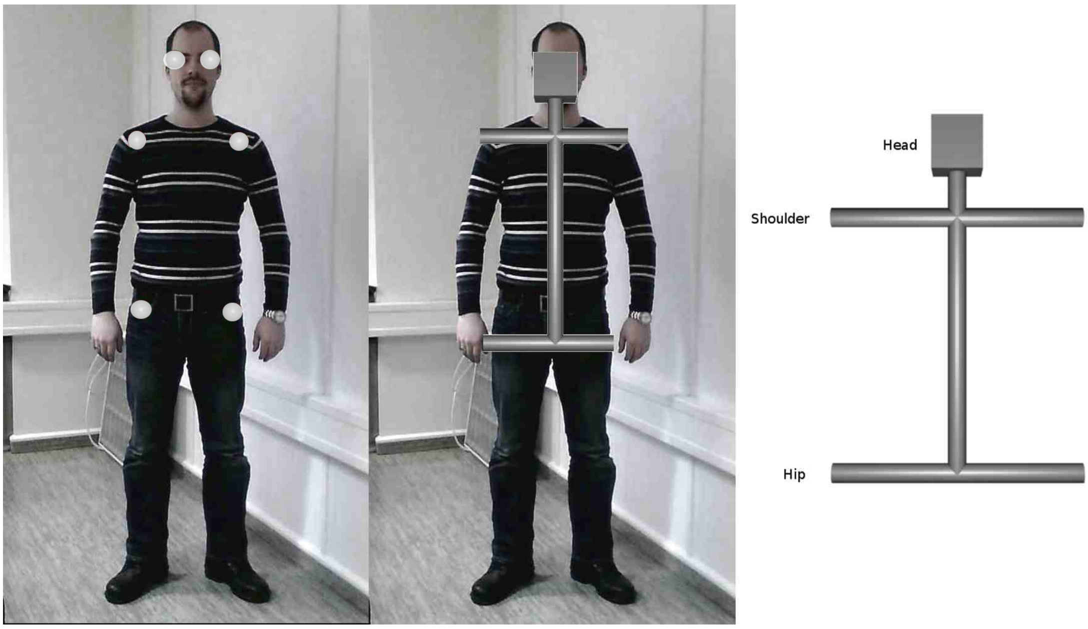


Figure 2.8: A model of a human skeleton simplified to have only the features of interest namely hip angle, shoulder angle and gaze direction

them against the model, and output a response needed for a command to be sent to the door. There are several techniques that are suited for this kind of work. The following sections present the ones most relevant to this project.

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 these functions come from the parent nodes, representing observed events, and in case of missing observations, the probabilities of these events.

Figure 2.9 illustrates a simple Bayesian network with three nodes. The network models *headache* as an effect of lacking 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?"*.

If using a Bayesian Network for this project, it could model the features (see Table 3.1, page 27) 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

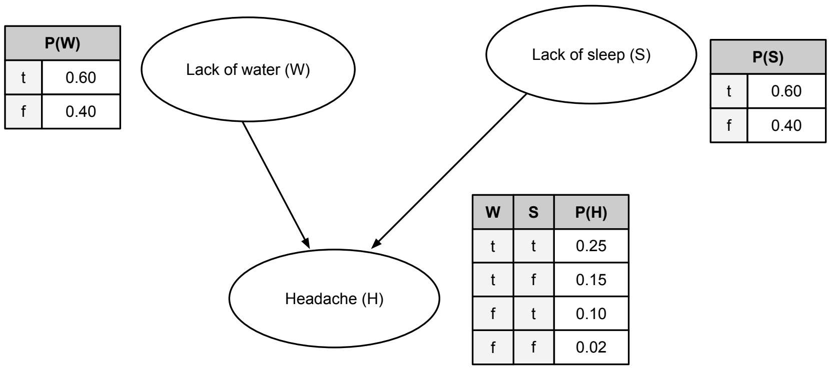


Figure 2.9: A simple Bayesian network

through the door. If it is higher than a set threshold the door can be opened, if not, leave it closed.

A challenge related to this approach would be to define 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. The possibilities for the different states leading to an intention can not easily be derived from my model of behavior (see Section 3.1, page 25), making this approach less desirable.

2.3.2 Decision Network

When making decisions, a useful tool can be decision networks[4]. This is a general mechanism for making rational decisions, following the principle of maximum expected utility (Russell and Norvig, 2009). 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) represent the choices to be made. The utility nodes (diamond) represent 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 2.10 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.

[4]Also known as influence diagrams

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.

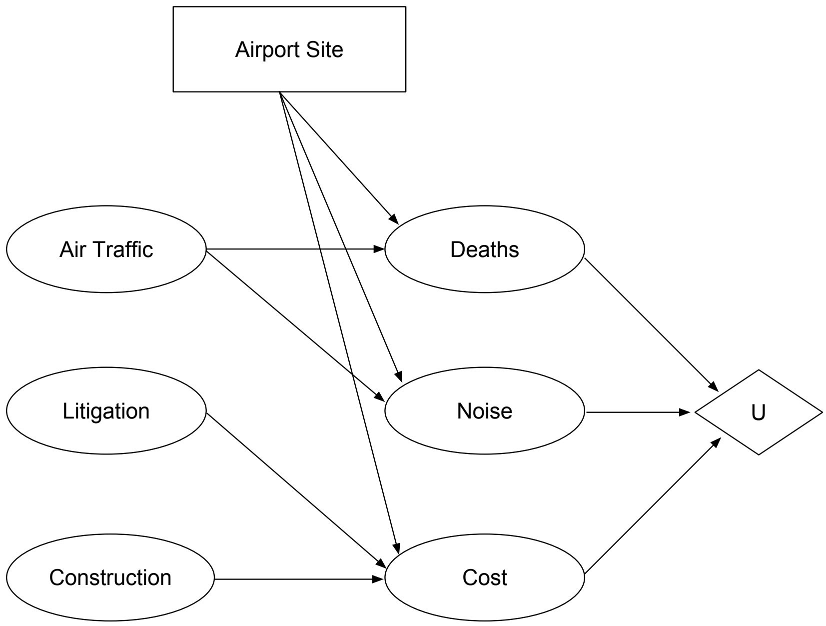


Figure 2.10: A decision network (recreated from Russell and Norvig, 2009)

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, 2009). 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 2.11 illustrates an HMM with E as the evidence variable, and X as the hidden state variable.

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. HMMs are probabilistic, and will create the same challenge for modeling as the Bayesian Network.

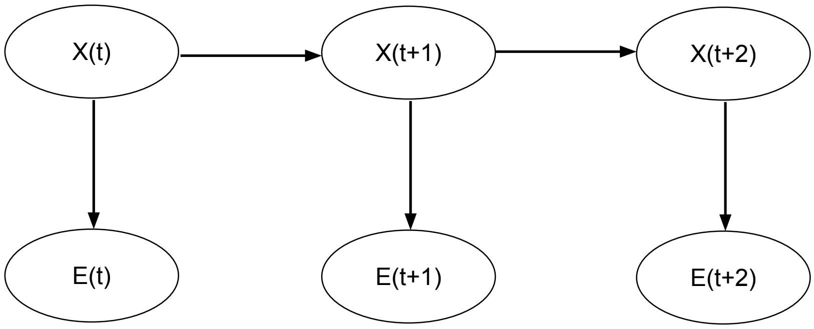


Figure 2.11: A general Hidden Markov model

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. This gives a control flow for execution of the different parts of the program.

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 see 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. My application has a limited context; the input parameters are few and the outcomes are limited. The rules for the outcome are also unchanging. This makes rule-based reasoning a viable candidate for the reasoning mechanism. Figure 2.12 demonstrates a small set of primitive rules that involve features for the door opening.

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 de-

```
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 2.12: Example of a simple rule base for this project.

scribed in this section are not what I primarily aim for in the project goals, but they 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, 2009). 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 the sliding door case, there are 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's only action is to open or not to open. No knowledge about when to open, results in the door always being closed[5]. 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 tree 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 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 2.13.

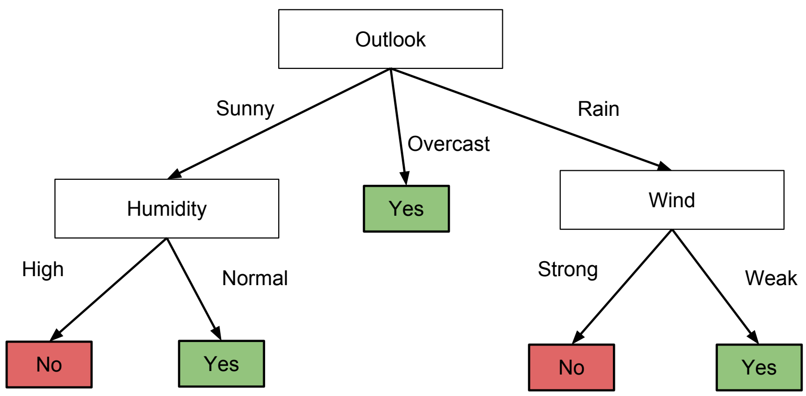


Figure 2.13: A simple decision tree for the decision problem *Should I play tennis*; Leaf nodes annotate decisions. (Recreated from Mitchell, 1997)

[5]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.

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 overfitting (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. A decision tree can be transformed into a rule set, and is in this way similar to the rule-based reasoning, where the main difference is the learning process.

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 the reasoning problem in this project, this might not be the best approach. With few methods of providing continuous feedback, it is better with a more static learning phase.

Chapter 3

Research Results

3.1 Modeling human behavior

The model of human behavior is the basis for the rest of the work. This model guides the data capturing process, defining the features to produce. It also provides the foundation for the rules in reasoning engine. Without a strong, precise model, it is unclear which data that is useful to collect, and furthermore the reasoning engine will have no knowledge about human behavior and intentions.

The work of building the model mapping the features to intentions, is based on the model suggested by Kofod-Petersen et al. (2009) and described in Section 2.1.1 (page 10). The previous work gives a starting point, but I need some further specifications of this model, in order to make it fit to the problem. When writing about intentions I will refer to the intention of walking through the door as a *positive intention* and the intention of not walking through the door as a *negative intention*.

3.1.1 Motion

Motion can be divided into more features. I suggest *speed*, *acceleration* and *heading*. Speed can be used to determine if a person is moving or not. Acceleration can be useful for predicting future motion, like slowing down for stopping and speeding up for leaving. Heading, or direction of movement, is obvious and necessary in order to read the intention of people.

The motion feature, with its specifications, requires a perspective of time. A single captured frame will only give parts of the information needed, thus a sequence of frames is required. As a result of this, a tracking mechanism is needed that follows the subjects in front of the door over time.

3.1.2 Proximity

Proximity is a simple feature defining the distance between the door and the user. Regardless of its simplicity, this measure is one of the most important in the model. A user that is far away from the door can easily be ignored, until the distance is shorter. The door should not open before it is necessary. Although the user is coming towards the door, and is displaying all the signs of a positive intention, the final destination might be another. For that reason it is better to delay the decision. The signs that indicate an intention will also change based on the distance to the door. In the model three levels of proximity are defined: *close*, *nearby* and *distant* (see Figure 3.1).

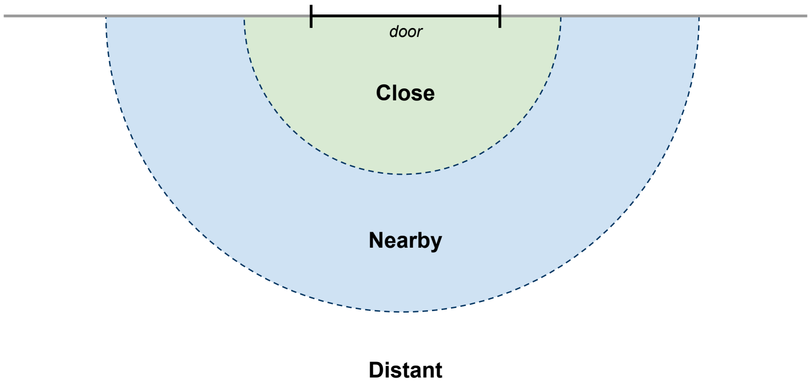


Figure 3.1: Different levels of proximity.

3.1.3 Body alignment

The body alignment describes *how* the user's body is positioned. Since my focus has mostly been on the interaction between the user and the door, this positioning is described only relative to the door and not other people. Kofod-Petersen et al. (2009) define this feature as a combination of shoulder angle and hip angle. The body alignment becomes more important when the distance to the door is small and the generation of movement might be limited.

3.1.4 From feature to intention

We now have six features that can be used to model intentions of the human behavior related to the interaction with an automated sliding door. Each of the features have different values, ranges and units. The domain has got a limited range of values, as this is for human beings. Table 3.1 gives an overview of the features together with the suggested range of expected[1], valid values and units of measure.

Table 3.1: Features, ranges and units

Main feature	Feature	Range/Unit
Motion	Speed Acceleration Heading	$[0, 10]$ m/s $[-5, 5]$ m/s^2 $[0, 360\rangle$ °
Proximity	Distance	$[0, 10]$ m
Body alignment	Shoulder angle Hip angle	$[0, 360\rangle$ ° $[0, 360\rangle$ °

Based on these features it is possible to put together symbols, that in combination will form different events. The events express the intention of the user. Table 3.2 shows events based on symbols that can be created using the features.

Table 3.2: Events, symbols and intention

Event	Symbol	Feature	Intention
Turning in	Close Facing door	Distance Body alignment	Positive
Incoming	Nearby Headed towards door Walking	Distance Heading Speed	Positive
Moving away	Nearby Headed away Walking	Distance Heading, acceleration Speed	Negative
Hanging around	Nearby Lingering	Distance Speed	Negative
Remote ignore	Distant	Distance	Negative

The features, symbols and events as described here will be the model for the human intentions. The features must be calculated from the raw features extracted from the video stream. This is then the input to the reasoning mechanism, for the decision to be made about what is happening.

[1]This is the expected values, bordering to extreme, as human beings can produce greater acceleration and speed than one would expect in front of a door.

3.2 Capturing the intention

With the model of features for human behavior in mind, the capturing process is given a direction. The features that need to be extracted are defined. In Section 2.2 different approaches were discussed, and a set of tools were presented. A capturing mechanism must obey the following requirements:

- Real-time handling
- Body recognition
- Body tracking
- Depth mapping

A sliding door operates in a real environment. Since the system must capture, analyze and respond to a user's behavior, it is very important with real-time handling. Time-consuming tasks could create unacceptable delays, resulting in a non-functional door. It is therefore important to know how the different tools perform.

3.2.1 Evaluation of computer vision tools

All testing was done on a computer with Intel Core 2 Duo 3.0 GHz and 3.7 GiB of RAM. The algorithms were tested using a video clip consisting of 114 consecutive frames, each frame treated separately. Based on the features as defined in Table 3.1 (page 27), I need frame rates of at least 10-15 frames per second (fps).

Traditional segmentation techniques

The traditionally used computer vision techniques build the foundation for many tasks. Although methods like Sobel and Canny for edge detection function well, some additional steps are required for detecting humans. The Otsu algorithm for separating foreground from background was tried out, as well as using image subtraction, finding the difference between two images (one serving as a background model) as raw input for human detection.

The complexity of the scenes and the changing lighting conditions, together with the auto adjustments of the camera made it hard to get any good results, without manually tweaking parameters for each separate frame. This is something that cannot be afforded when working real-time, so a more robust solution is needed. Table 3.3 shows that all these components in themselves performed very well and pose no threat to the real-time demand. The *average time* is the time used by the tool on the individual frames. The *frame rate* is calculated using the average time.

Table 3.3: Performance of different segmentation tools

Frame size	Segmentation tool	Average time	Frame rate
480x640	Canny edge detection	8 ms	125 fps
	Otsu threshold	2.1 ms	474 fps
	Image subtraction	1.8 ms	558 fps

Histogram of Oriented Gradients

Object detection using the oriented gradients overcomes some of the problems with the more traditional methods. It provides a robust mechanism for detecting objects, even in challenging environments. OpenCV[2] has got an implementation of this detector following the original description by Dalal and Triggs (2005). The original implementation suffers from low performance and might not deliver the performance needed for a real-time system, but improvements can be made. The tests were run using this implementation together with the supplied *defaultPeopleDetector*, an already trained people detector.

Testing the performance of the HOG-detector using different frame sizes gave the results as shown in Table 3.4. The initial tests were run on the consecutive frames from the video clip, all using the default parameters for the method. The video clip showed a person walking towards a door, turning around and walking back again. The person was present in all the frames. The *accuracy* is defined as frames where the person was detected. The *detection time* is the time that the algorithm needed for processing one frame (average). The frame rate is then calculated based on the average detection time. As can be seen, the performance is greatly increased by reducing the frame size. The accuracy of the detections is also kept down to the 50% reduction. The frame rate is not good for real-time with scales above 35 %.

Table 3.4: Detection time, using HOG on 114 consecutive frames

Frame size	Scale	Average detection time	Accuracy	Framerate
480x640	100 %	650 ms	100 %	1.5 fps
360x480	75 %	362 ms	99.1%	2.8 fps
240x320	50 %	151 ms	100 %	6.6 fps
168x224	35 %	65 ms	91.2 %	15.4 fps
120x160	25 %	22 ms	54.4 %	45.5 fps

Although the desired performance was not achieved using HOG descriptors,

[2]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.

work can be done to improve the implementation. Research has been done in delivering real-time performance with HOG (Zhu et al., 2006; Jia and Zhang, 2007). Further improvements are suggested where accuracy is improved and detection time reduced (Wang and Zhang, 2008). Using the computer's GPU for some of the workload may also increase the performance radically (Lillywhite et al., 2009; Bilgic et al., 2010; Sugano et al., 2010).

Attempts have been made in combining stereo vision and HOG descriptors, achieving better accuracy as well as real-time performance (Toya et al., 2008). This solution seems promising and might have had advantages for my project since information provided by stereo vision is needed anyway.

Recognizing human shapes in the images is just one component of the complete system. Getting the HOG-algoritms up to speed would only solve parts of the problem. With limited time and no good implementation of the algorithms at hand, it was better to look for alternatives.

3.2.2 Kinect and OpenNI

Following the development and the usage of the Kinect sensor device within the Open source community, new possibilities opened. This device is created for tracking humans, and can take on the depth calculations, giving more resources to other important tasks.

Kinect

The Kinect is a controller device produced for Microsoft's gaming console Xbox 360. It allows for motion tracking and gesture recognition through the built-in RGB-camera and depth sensor. Although the device is designed for use with a console, it is also possible to connect it to a computer. This however, requires a separate driver suitable for the Kinect. The device produces raw data in the forms of RGB data (color images) and depth data (depth map with distances in millimeters). The raw data must then be processed using software specially designed for the purpose. The device delivers its data well within real-time specifications, and can support an application that has real-time demands.

After experimenting with the Kinect, I found out that the vertical field of view is about 47°. When mounting the sensor above the door, one would have to tilt it downwards in order to cover the area close to the door. This will limit the depth of the field. However, this would also be a problem on most of the ordinary cameras available, the ones I tested included. Figure 3.2 illustrates limitations in the vertical field of view.

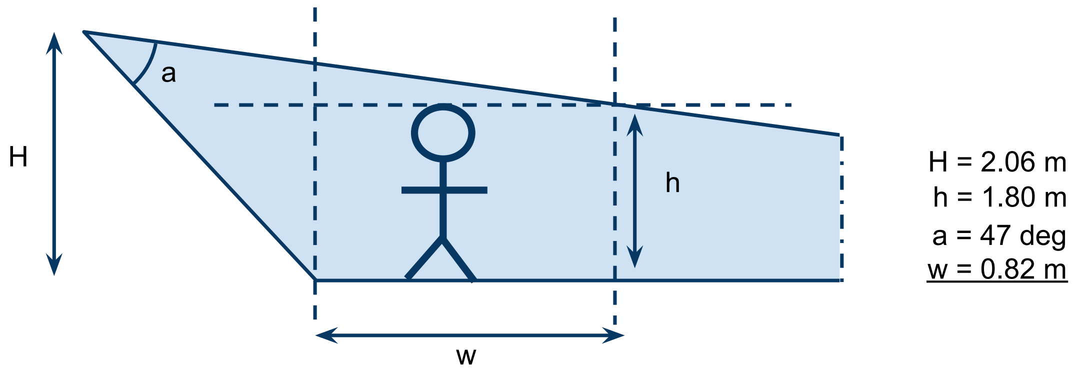


Figure 3.2: Vertical field of view. The figure illustrates the limited area that a person of height h would be visible in, in full height, when the sensor is in height H, and tilted downwards.

OpenNI

OpenNI is an open source framework that provides an interface for physical devices and for middleware components (OpenNI User Guide, 2010). The framework provides an API[3] for writing applications utilizing natural interaction. With OpenNI it is possible to write applications, that use sensor data, on a higher level, abstracting unnecessary details concerning different vendors. With the appropriate middleware, it is also possible to write applications that take advantage of third-party routines for different tasks like speech and command recognition, hand gesture recognition and body motion tracking.

Middleware - NITE

NITE is a middleware perception component released by PrimeSense intended for usage with OpenNI. The NITE Middleware focuses on two applications: *Control by Gesture* and *Games for All* allowing developers to track multiple users, handling movement through skeletal tracking and gesture recognition. NITE provides means for tracking humans and retrieving skeletal data, through a set of joint coordinates.

3.2.3 Calculating the features of interest

Using OpenNI in combinations with NITE and Kinect, I was able to track users of the door and retrieve the points of interest in world coordinates. This coordinate system uses X, Y and Z-axis, and has its origin in the sensor position, point (0,0,0). The unit used in the system is millimeters. With the points coordiantes available,

[3] Application programming interface

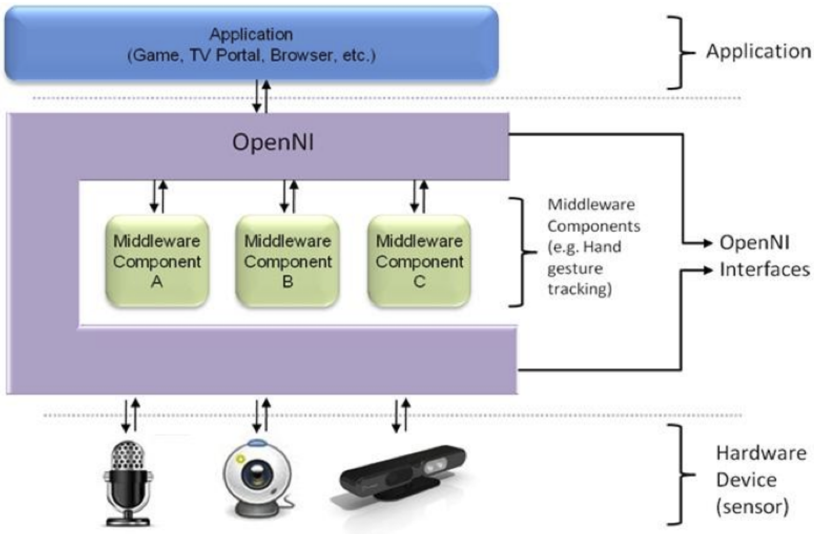


Figure 3.3: Layered view of the OpenNI concept (from OpenNI User Guide, 2010)

calculating the necessary features is reduces to a mathematical task. Figure 3.4 shows five points of interest, marked on a user, together with a visualization of the coordinate system.

The features can be divided into three groups, according to the number of data sets needed. The features distance, shoulder angle and hip angle, only need one set of points. Speed and heading need two sets of points and acceleration needs three sets of points. The calculations must be done in order, from single to multiple data sets, and can only be done when the sufficient number of data sets are retrieved.

Distance

In order to calculate the distance, I first need to define what the distance is. Here I choose the distance as the length between a representative point on the user (the torso) and the sensor, in the XZ-plane since the Y-direction is irrelevant. The length is found using vector formulas:

$$\text{Distance between torso and sensor: } \|\vec{t}\| = \sqrt{{t_x}^2 + {t_z}^2}$$

Body alignment

The body alignment in the model is defined as a combination of shoulder angle and hip angle, representing the alignment of the user's body. I choose to use the door as the point of reference and use the average of the two values, making the body alignment relative to the door. For this feature, the needed data are the

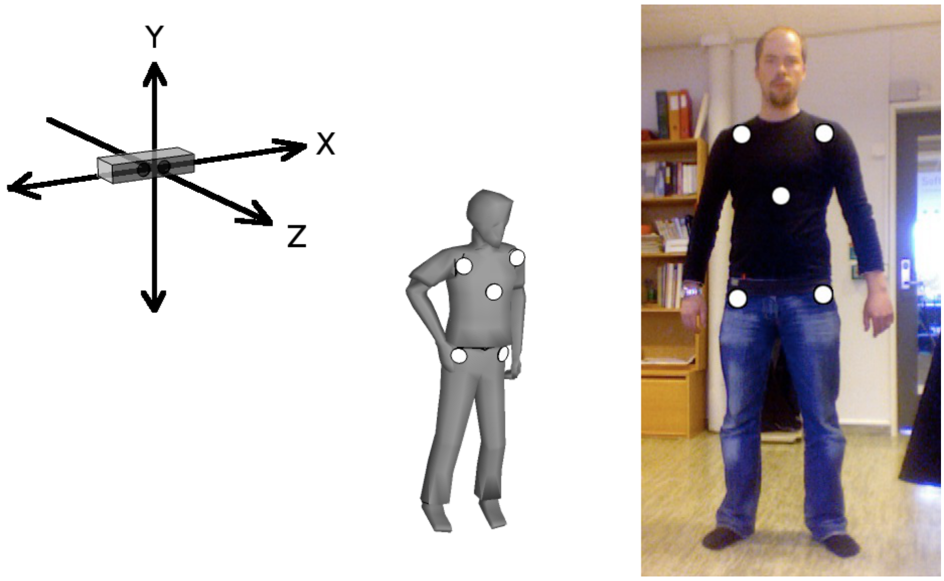


Figure 3.4: Five points of interest marked on a user: Left and Right shoulder, Torso and Left and Right hip.

shoulder points and the hip points (as shown in Figure 3.4). The Y-direction can also here be left out.

A simple solution would be to use the x-axis as the "wall" in which the door would be mounted. Then one could calculate the shoulder angle as the angle between the x-axis and the line intersecting the two shoulder points, and likewise with the hip angle. This, however, is not correct as it would neglect variations along the x-axis. Facing the wall is not the same as facing the door. This can easily be seen in Figure 3.5 comparing the two angles a and a'. The figure shows two different people in the coordinate system. While the first one (p_1, p_2) is situated on the z-axis , the other one (p'_1, p'_2) is further to the right. The angles are both 30°, but the body alignment relative to the door is different in these two cases.

To get this angle correct, one must take a different approach, involving some definitions: I define the door to be a point. The angle, either shoulder or hip, is relative to this point. I define the angle to be 0° when the body is aligned with the door (the user is facing the door). Further, I simplify the model by saying that the direction of the alignment, either left or right, is indifferent; giving a range from 0° to 180°.

Figure 3.6 illustrates the process of finding the wanted angle. Here p_1 and p_2 are the two shoulder points. First I find the middle point, m, between p_1 and p_2. Second I define an arc, with its center in the door point, intersecting m. The shoulder angle is then the angle between two lines, line P intersecting p_1 and p_2,

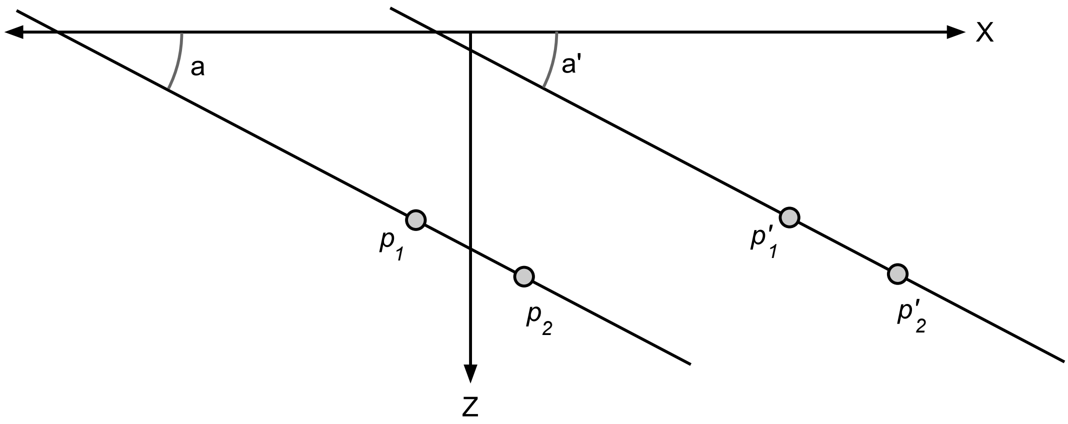


Figure 3.5: Failed attempt of modeling the body alignment.

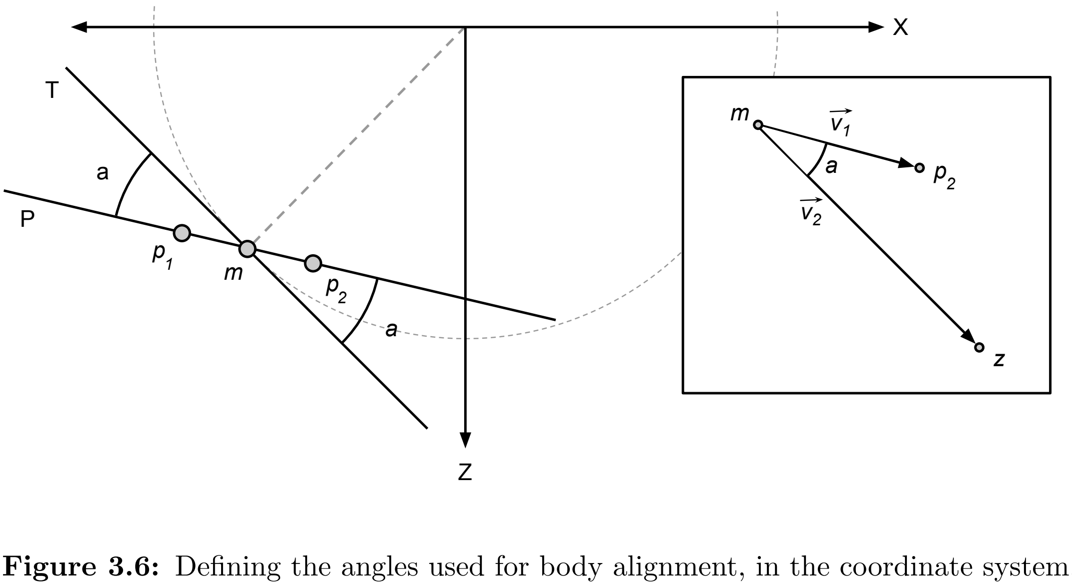


Figure 3.6: Defining the angles used for body alignment, in the coordinate system.

and line T being a tangent line to the arc, intersecting m.

Calculating the angle can be done using vectors and trigonometry:

$\vec{v_1}$: vector between m and the shoulder point closest to the z-axis.

$$\vec{v_1} = [p_x - m_x, p_z - m_z]$$

$\vec{v_2}$: vector between m and the the intersect with the z-axis (following T).

$$\text{Tangent line } T : m_z = am_x + b \Rightarrow m_z = -\frac{m_x}{m_z} + b$$

$$\text{Intersect: } b = m_z - \left(-\frac{m_x}{m_z}\right) m_x$$

$$\vec{v_2} = \left[-m_x, \frac{{m_x}^2}{m_z}\right]$$

$$\text{Shoulder angle: } a = \arccos\left(\frac{\vec{v_1} \cdot \vec{v_2}}{\|\vec{v_1}\|\|\vec{v_1}\|}\right)$$

The same calculations are then done for the hip angle, and an average of these two values give the body alignment.

Heading

Heading is simply expressed by the direction of the user's movement. The measure is chosen to be degrees away from a direction straight towards the door, and can be in the range from 0° to 180°.

Calculations are done by comparing the user's position at two different times, t_1 and t_2.

$\vec{v_1}$: vector defining straight line to door

$\vec{v_2}$: direction of movement

$$\vec{v_1} = -[Px_2, Pz_2]$$

$$\vec{v_2} = [\Delta X, \Delta Z] = [Px_2 - Px_1, Pz_2 - Pz_1]$$

$$\text{Heading: } h = \arccos\left(\frac{\vec{v_1} \cdot \vec{v_2}}{\|\vec{v_1}\|\|\vec{v_1}\|}\right)$$

Speed

The calculation of speed uses the same data as heading, as well as the time difference between t_1 and t_2. The speed as defined here leaves out direction.

$$\text{Speed: } v = \frac{s}{t} = \frac{\sqrt{(\Delta X)^2 + (\Delta Z)^2}}{t_2 - t_1} = \frac{\sqrt{(Px_2 - Px_1)^2 + (Pz_2 - Pz_1)^2}}{t_2 - t_1}$$

Acceleration

Calculating acceleration requires more data sets than the other calculations. Acceleration can be calculated by using the velocity at two different time points and the elapsed time between them.

v_0: initial velocity, v: final velocity, t: elapsed time

$$\text{Acceleration: } a = \frac{v - v_0}{t}$$

3.3 To open or not to open

The research done in the preparation work pointed out *rule-based reasoning* as a well suited mechanism for infering intentions (see Section 2.3.4, page 21). The model of human behavior and intention defined symbols and events; they can be seen as input to the reasoning process.

The user tracking process provides the points of interest that are used for the calculation of features. All points collected from one sensor update, produce the features belonging to this set of points. The features can then be grouped into *feature sets*, belonging to one specific user with one timestamp.

The feature sets are the input for the reasoning mechanism. Providing a stack of feature sets for each user, may strengthen the reasoning process, by allowing the reasoning to be based on events that form over time. Figure 3.7 shows how the feature sets are related to users.

The system has to be able to tell if a user wants to go through the door at any time. Rules that analyze the feature sets can do this.

3.3.1 Defining rules

Defining rules for reasoning is really a process of instantiating and grouping the symbols described in Section 3.1.4. The values as suggested in Table 3.5 are a product of intuitive measures together with some trial and error and limitations concerning the sensor equipment and setup.

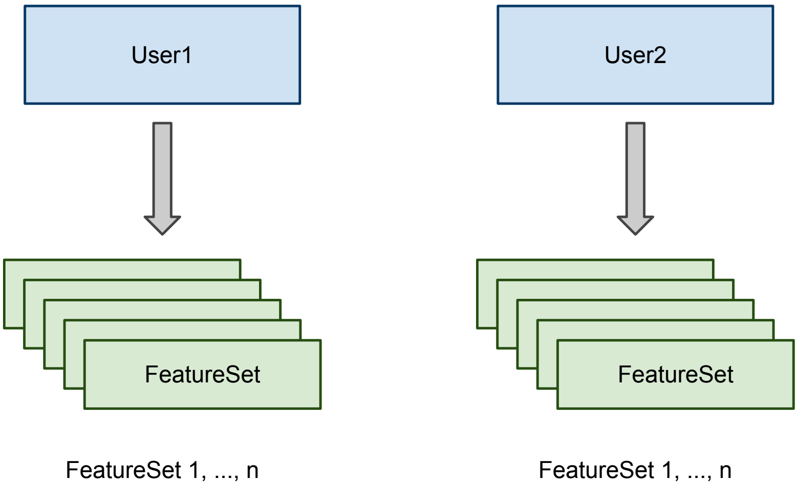


Figure 3.7: Each user being tracked, stores a stack of feature sets.

Table 3.5: Instantiating the symbols

Feature	Symbol	Value
Distance	Close	$d \in [0, 1.2]$ m
	Nearby	$d \in \langle 1.2, 1.8]$ m
	Distant	$d \in \langle 1.8, \rightarrow \rangle$m
Speed	Walking	$v \in \langle 0.8, 8]$ m/s
	Lingering	$v \in [0, 0.8]$ m/s
Body alignment	Facing door	$ba \in [0, 4]\,^\circ$
Heading	Headed towards door	$h \in [0, 4]\,^\circ$
	Headed away	$h \notin [0, 4]\,^\circ$

The reasoning process can be visualized using a decision tree. Figure 3.8 shows how the different symbols are used in order to make a decision (note that the tree is *not* a product of a learning algorithm).

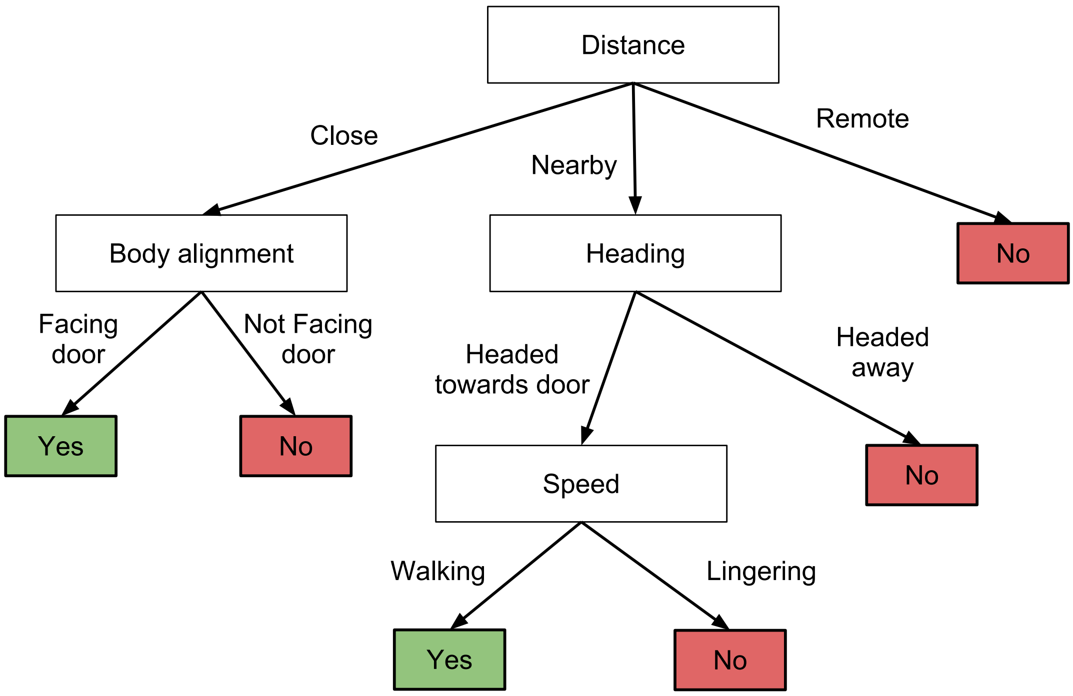


Figure 3.8: Decision tree visualization of rule base.

The transition from this tree to a rule base, using simple IF-statements, is trivial using the defined symbols.

3.3.2 Avoiding false positives

For every update from the sensor, new data is available, depending on the number of users being tracked at the moment, a number of feature sets are made available. Because of the high update rate it is possible to combine the information from several feature sets to get a better decision. In order to avoid false positives, I made the system count *three* positive intention indications before it would allow the door to open.

3.4 An automated sliding door

The most central components in the project are the sensors and the sensor readings together with the interpretations of these. All the main tasks are solved using these components. The sensors capture the movements of the users, and the computer analyzes the sensor readings, inferring an intention and finally decides the appropriate action. Although this is in its own a complete system, it is not enough. In order to evaluate the performance of the system altogether, some kind of feedback is necessary. Strictly speaking, this could have been limited to any binary output, symbolizing a door opening. The feedback could easily have been a LED or a light bulb, lighting up whenever the system chose that *open* was the appropriate action. However, this is not a very good idea. The system needs to analyze the behavior of people interacting with a door. A simulated door is harder to interact with. Walking through a simulated door is unnatural and might result in abnormal behavior. The need for a real door is therefore present.

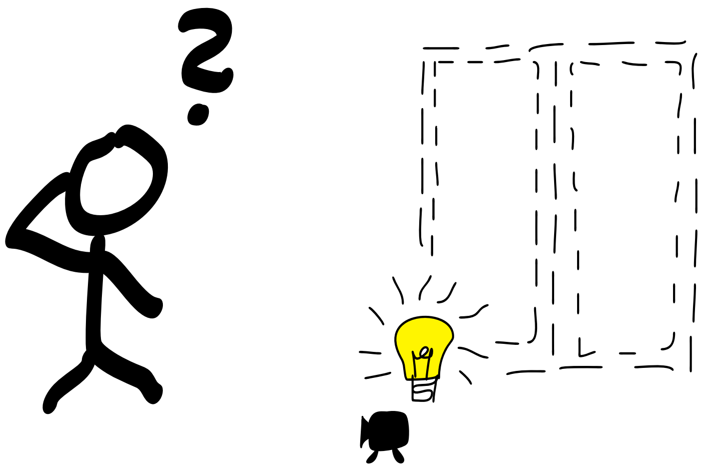


Figure 3.9: Abnormal behavior

An automated sliding door is a common device, and something that most people are accustomed to. Having an actual working sliding door will give the most desirable conditions and accurate results.

A viable solution would have been to equip an existing door with additional sensors. It would then be possible to evaluate the performance by comparing the system outputs (decisions made) with the user's behavior and the actual door outputs (door opening or not). This would also provide a comparison between my system and existing systems. This would however complicate the process, in several areas. First, a real door in a real environment, although more realistic, is harder to use both for development and testing. This is mostly because of the environmental factors that are out of my control, like the amount of people using the door and the lighting and weather conditions. Second, the feedback would be indirect, as the door's actions will not always correspond to the inferred intention. My system's output must then be checked in other ways. In other words, the communication that is going on is actually between the user and the existing system, rather than the user and my system. Third, using an actual door raises privacy issues, demanding alternative routes, as described in Section 3.4.2.

Having my own door that can be controlled directly, would be the best solution. Ready-made, motorized doors are very expensive and buying one was not an option. This left me with the option of building a door myself.

3.4.1 Building a door (the *do-it-yourself* way)

A sliding door can be very simple; take a door and put wheels on it. This is basically how the "household" type of sliding doors are built. One might think that these doors can be found everywhere. As it turns out, sliding doors are not so common, and can only be bought as complete sets. These type of doors are very hard to customize because of the compactness needed for built-in doors. A motor and a mechanism for pulling the doors would simply not fit in the sets. As a result of this, I needed to build the door from scratch.

Here are some of the choices for the specifications:

- *Double doors* – Wider access and lower speed needed in order to open.
- *Wood as building material* – Cost efficient and easy to work with.
- *Belt and cog wheels* – Reliable and easy drive system.

The proposed solution consists of different parts:

- *Door frame* – A self supported frame that can hold the doors, without the need of a wall.
- *Doors* – Double doors, with wheels for usage with rails in the door frame.
- *Drive system* – Belt and cogwheels mounted on top of the door frame, pulled by an electrical motor.
- *Door controller system* – Electronic component, computer controlled and able to run the motor, in both directions, at correct speed.

Figure 3.10 shows an image of the finished door. The sensor is attached and the computer controlling the door can be seen on the left side. Further descriptions of the design of the door frame and door can be found in Appendix B. The total cost of the door with sensor and door controller was less than 7000 norwegian kroner (computer excluded). This is about 15 % of the price one would have to pay for real automated sliding doors.

Figure 3.10: The door frame with the doors mounted

Door controller system

One of the more important features of the door, is its ability to be controlled by a computer, using plain programming languages. This requires some kind of communication between the door and the computer. There are many ways to do this, but one of the simplest ways is to use a relay controller board. A relay is an electrically operated switch. Using a relay has advantages, both when it comes to remote controlling a switch and separating the controlled circuit from the controller. The relay can be used to control a high voltage circuit, using a low voltage power signal.

There are different types of relays, as with switches, concerning the number of poles and throws. The purpose of my circuit is to offer three settings; *open*, *close* and *off*. Opening and closing is simply a matter of changing the direction of the current in order to change the direction of the motor. This can be achieved by using two SPDT relays (single pole, dual throw). The circuit must be constructed in such a way that no combination of the switches will short circuit the system. This requirement protects the system as timing errors or programming errors could result in unwanted combination of relay states.

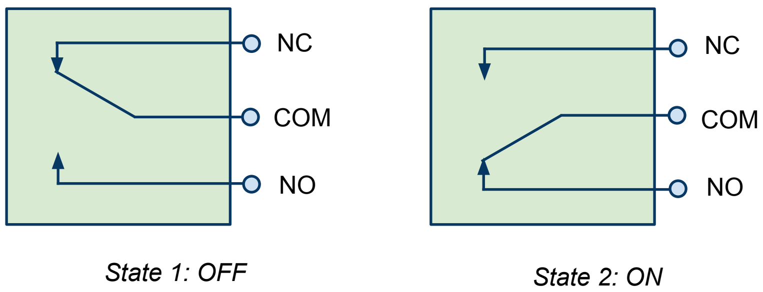


Figure 3.11: An SPDT switch with three terminals. To the left: *OFF-position*; *Normally closed* and *Com* are connected. To the right: *ON-position*; *Normally open* and *Com* are connected.

Operating a door is a precision task. Ideally, one would have sensors giving feedback to the door controller regarding the position of the door. The door controller can then easily stop the doors in the correct positions: fully open and fully closed. Adding two extra switches to the circuit, of the type *push-to-break* or *normally open*, can give a similar control, by cutting the engine power, preventing the doors from slamming or even breaking (also protecting engine and transmission) when in positions fully open or fully closed. These switches must be placed in the circuit in such a way that even if the circuit is broken for one action (open or close), it can not be broken for the other, otherwise the control over the doors is lost.

An electric motor is a type of inductive load because of the magnetic coils (Sheldon, 2008a). Inductive loads create voltage surges when the circuit is broken. These surges can cause sparks in the switches and electromagnetic interference. One effect of this is shortened lifetime of the relays and switches. Another effect is that the electromagnetic inference can disrupt the communication with the computer. Since USB is prone to errors, with no error checking or self correction, USB controllers are not recommended when controlling inductive devices (Sheldon, 2008b). A solution to this problem is to suppress the induction with a capacitor in the circuit, and to use a controller with another interface than USB.

The designed circuit is shown in Figure 3.12. In the figure we can see the switching circuit employing all three terminals of the relays in a fail-safe way. Turning on one of the relays will open the door, and turning on the other will close the door. Both relays on or off is the same as off or standby. Table 3.6 lists the possible relay combinations together with their actions. NO and NC in the figure refer to *Normally Open* and *Normally Closed*. The capacitor is placed in parallel with the motor, close to the com-terminals of the relays.

Table 3.6: Relay settings and action

Relay 1	**Relay 2**	**Action**
OFF	OFF	-
ON	OFF	**Open**
OFF	ON	**Close**
ON	ON	-

Drive system

The electric motor required to pull the belt and the doors, does not have to be very strong, and a moderately sized motor should suffice. However, the speed is more important. The doors must open in a time and speed that feels natural. With double doors, the speed can be reduced compared to a single door, in order to get the wanted time. A simple calculation gives the optimal rotational speed:

$$\text{Wheel diameter: } 5\,\text{cm}, \text{ Wheel circumference: } 2\pi r = 15.7\,\text{cm}$$

$$\text{Door width: } 60\,\text{cm}, \text{ Wanted opening time: } 1.5\,\text{s}$$

$$\text{Speed: } \frac{0.6\,\text{m}}{1.5\,\text{s}} = 0.4\,\text{m/s}$$

$$\text{Rotational speed: } \frac{0.4\,\text{m/s}}{0.157\,\text{m}} \approx 2.5\,\text{Hz}$$

The force needed to pull the belt in order to open the doors, is greatest in the beginning. Using a scale reveals the that the pulling force needed is $3.5\,\text{kg} \cdot 9.81\,\text{m/s}^2 = 34.3N$. Another calculation gives us the necessary torque:

$$\tau = F \cdot r$$

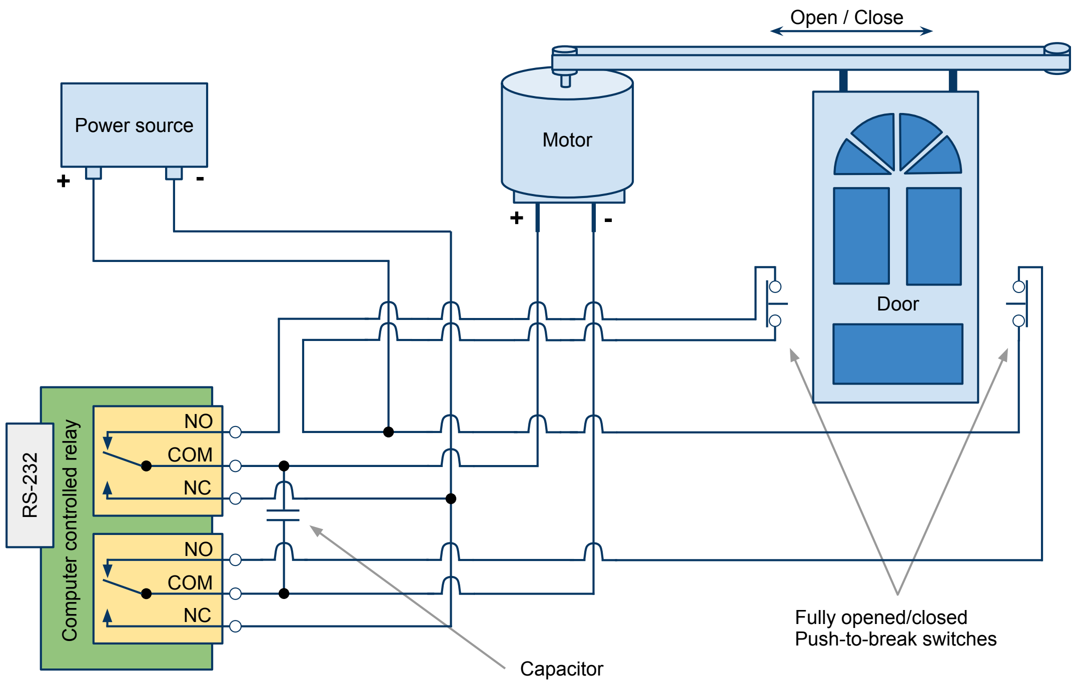


Figure 3.12: Circuit diagram for the door controller.

$$\tau = 34.3\,\mathrm{N} \cdot 0.025\,\mathrm{m}$$

$$\tau = 0.86\,\mathrm{N\,m}$$

Both speed and torque of these sizes can easily be supplied by a simple electric motor like the ones in a battery powered drill. The finished drive system can be seen in Figure 3.13.

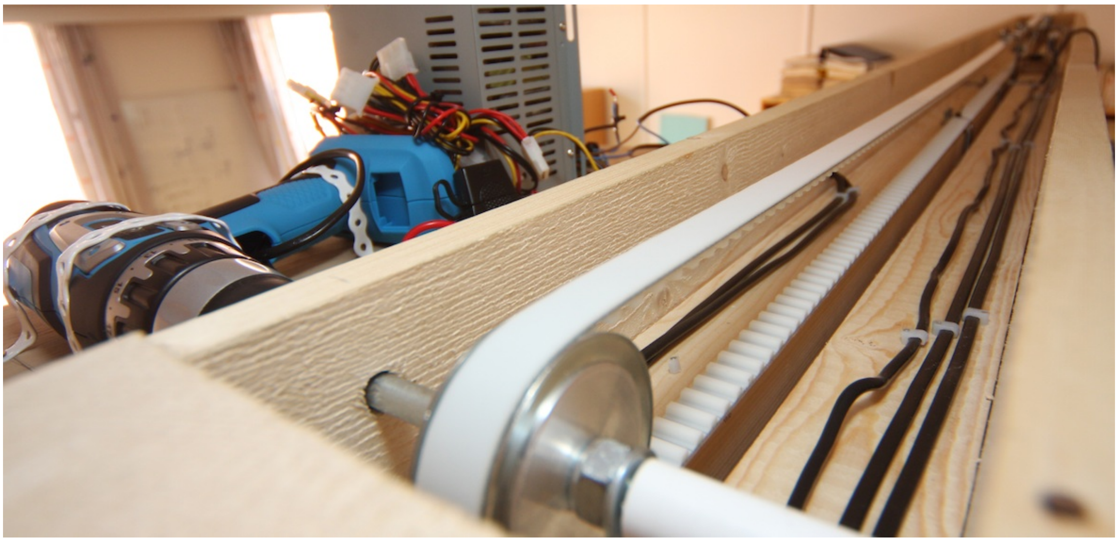


Figure 3.13: Picture showing the drive system. The drill is on the left; the belt stretches along the door and the wheels are in each end.

Testing and redesigning

The drill that I got for the setup provided faster rotational speed than necessary, so the circuit was expanded with a resistor. With a battery that gave 12 V, the ideal door speed was obtained by inserting a resistor of $0.5\,\Omega$. Because of the high current (about 8 A) and the generated heat, it was necessary to use multiple resistors of higher resistance in parallel instead of one smaller. Equations (3.1) and (3.2) show that using three resistors of $1.5\,\Omega$ in parallel give the wanted total resistance. To protect the electronics and the circuit, I also installed a fuse rated to 10 A.

Resistors in parallel:

$$\frac{1}{R_{tot}} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} \tag{3.1}$$

$$\frac{1}{0.5\,\Omega} = \frac{1}{r} + \frac{1}{r} + \frac{1}{r} \Rightarrow r = 1.5\,\Omega \tag{3.2}$$

When using the door for a longer period of time, I found that the battery did not supply a current that was stable enough for controlling the door; the voltage dropped and the door did not fully open and close. The solution to this problem was to replace the battery with a wall connected power supply. The power needed for my circuit is higher than the ones provided by simple power supplies where a transformer is plugged into the wall socket. A better alternative is to use an ATX computer power supply unit (PSU). This unit can provide a stable voltage supporting higher currents. The power needed can easily be calculated:

$$\begin{aligned} P &= U \cdot I \\ 12\,\mathrm{V} \cdot 8\,\mathrm{A} &= 96\,\mathrm{W} \\ &\approx 100\,\mathrm{W} \end{aligned}$$

A suited power supply was rated to 300 W with max current of 16 A on a 12 V channel. The design guide for ATX power supplies (Intel, 2007), explained the layouts for the different connectors and proved helpful when putting the PSU to use.

Since the PSU is not controlled by a computer, no signal is sent to power up the unit. This is usually controlled with the *PS_ON#*-pin (pin 16) on the main power connector. The PS_ON# is an active-low, TTL-compatible[4] signal. When PS_ON# is pulled to low, the power supply turns on the output rails. When PS_ON# is pulled to high or open-circuited the output rails deliver no current (Intel, 2007). Connecting PS_ON# to a COM (ground) pulled the signal to low, providing a constant supply of power. The layout of the connectors together with the used pins can be seen in Figure 3.14 and Table 3.7.

Any one of the power connectors with a +12 V line could have been used to provide the power for the controller circuit. I chose to use one of the many peripheral power connectors. The connector layout and the usage is shown in Figure 3.14 and Table 3.8. The finished door controller can be seen in Figure 3.15.

[4]TTL – Transistor-transistor logic, digital signaling that uses voltage levels representing logical states.

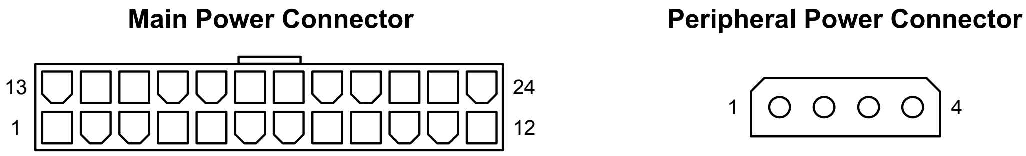


Figure 3.14: Connector layout for an ATX power supply unit

Table 3.7: Pin layout for ATX Main Power Connector. Connecting pin 7 and 16 turned the power supply on.

Pin	Signal	Color	Pin	Signal	Color
1	+3.3VDC	Orange	13	+3.3VDC	Orange
2	+3.3VDC	Orange	14	-12VDC	Blue
3	COM	Black	15	COM	Black
4	+5VDC	Red	**16**	**PS_ON#**	**Green**
5	COM	Black	17	COM	Black
6	+5VDC	Red	18	COM	Black
7	**COM**	**Black**	19	COM	Black
8	PWR_OK	Gray	20	Reserved	N/C
9	+5VSB	Purple	21	+5VDC	Red
10	+12VDC	Yellow	22	+5VDC	Red
11	+12VDC	Yellow	23	+5VDC	Red
12	+3.3VDC	Orange	24	COM	Black

Table 3.8: Pin layout for ATX Peripheral Power Connector. Pin 1 (yellow wire) and 2 (black wire) were connected to the controller circuit, replacing the battery.

Pin	Signal	Color
1	**+12VDC**	**Yellow**
2	**COM**	**Black**
3	COM	Black
4	+5VDC	Red

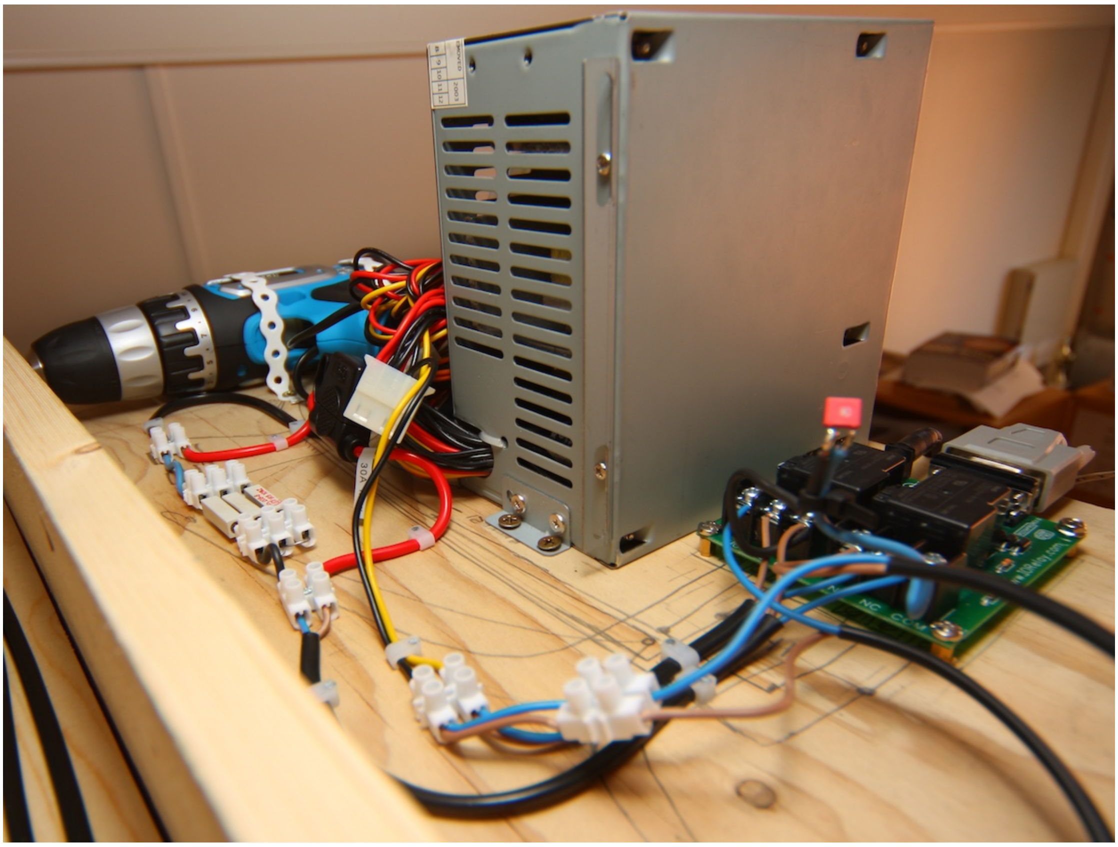


Figure 3.15: Picture of the finished door controller. The relay card is on the right side, the power supply in the middle and the drill on the left. In front of the drill the speed control and the fuse box can be seen.

3.4.2 Personal Data and Privacy

The operation of the door requires recording video sequences by camera, mostly for processing, but also for storing if necessary for testing and evaluation. The users of the door are persons. This brings forward the need for correct handling of personal data. NSD[5] is the organ for norwegian universities, that handle some of the responsibility delegated by the Data Inspectorate concerning these issues. A notification is needed in a case like this, describing the project, its means and purposes, as well as the handling of the information that may be considered personal. A concession was given, provided that I would follow the requirements as described. See Appendix A for details concerning the issue.

In short, here are some of the requirements:

1. Written notification is posted next to the door, explaining the recording of data, the handling of the data as well as the purpose of the project.
2. A consent is given by the user. This is given by using the door.
3. People that do not want to participate can choose alternative routes.
4. Collected data must be deleted at the end of the project (20.06.2011).

3.5 Developing an application

In the start-up process I had to choose which technologies and platform to use for the development of the software components. Research results from the specialization project proved OpenCV to be a useful tool when developing imaging software. OpenCV is a library providing functions for computer vision with focus on real-time applications. It was originally written in C, but it has a full C++ interface. The initial testing and evaluation of tools and algorithms were done using OpenCV. OpenNI is a more specialized framework, better suited for my project. It is also written in C, but a C++ wrapper is provided. This led me to the following choices:

- *C++ as programming language* – Easy integration of libraries and frameworks
- *Linux (Ubuntu) as platform* – Less trouble when working with open source packages, compiling and building releases.

My background as a Java developer required me to take a two week intensive course in C++.

[5]Norwegian Social Science Data Services

3.5.1 Application components

The application can roughly be divided into three components:

Door controller Component that handles communication with the door. It provides commands for opening, closing and stopping the door.

Inference engine Component that handles users, calculates features and infers intentions.

Sensor framework Component that communicates with sensors, interprets the raw input, and supplies routines for body recognition and skeletal tracking.

Figure 3.16 shows these components in relation to the hardware. The sensor framework consists of external components while the two others had to be developed.

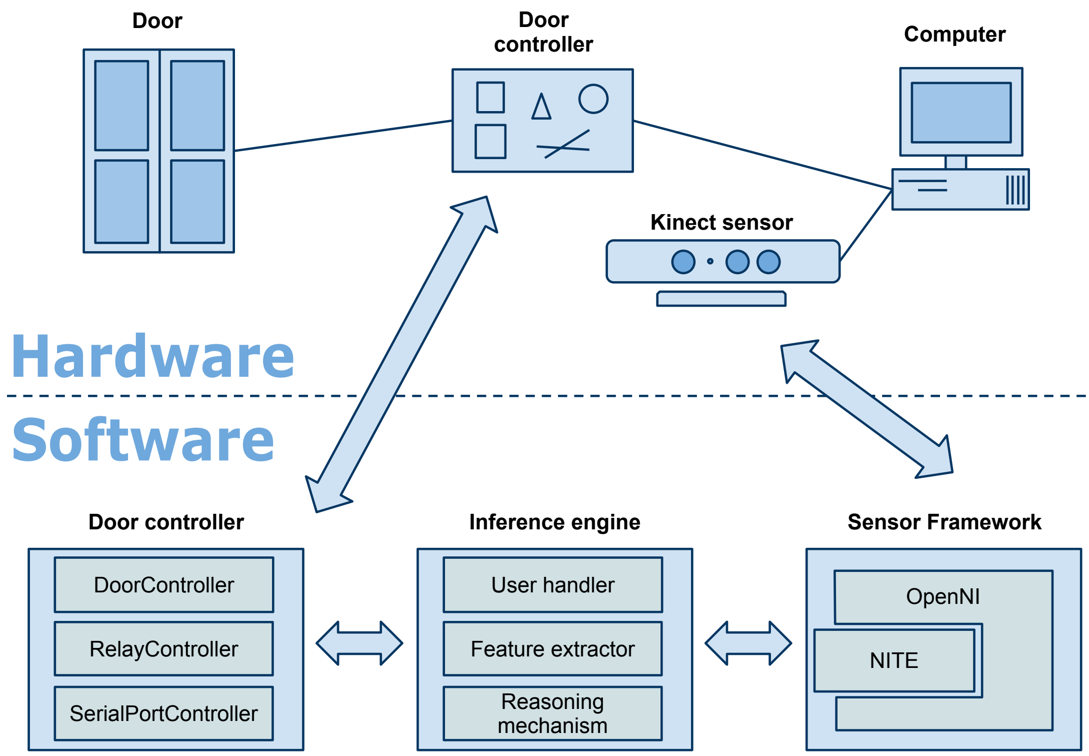


Figure 3.16: Component diagram showing implementation.

In Figure 3.17 a class diagram is presented. This diagram gives an overview of how the implementation was done, including attributes, methods and the relation between classes. Another view is presented in Figure 3.18 where the different steps of the inference process are shown.

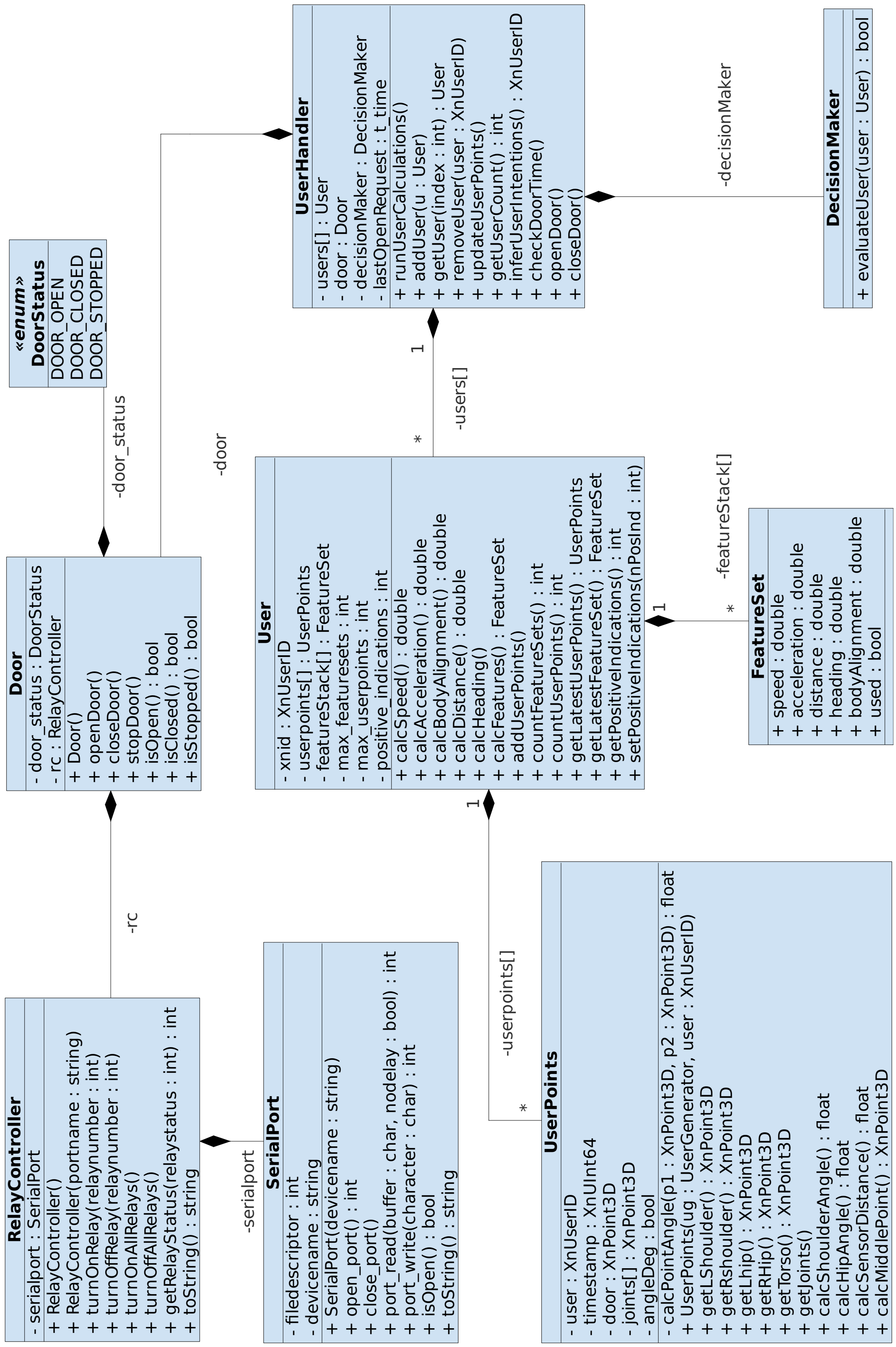


Figure 3.17: Class diagram

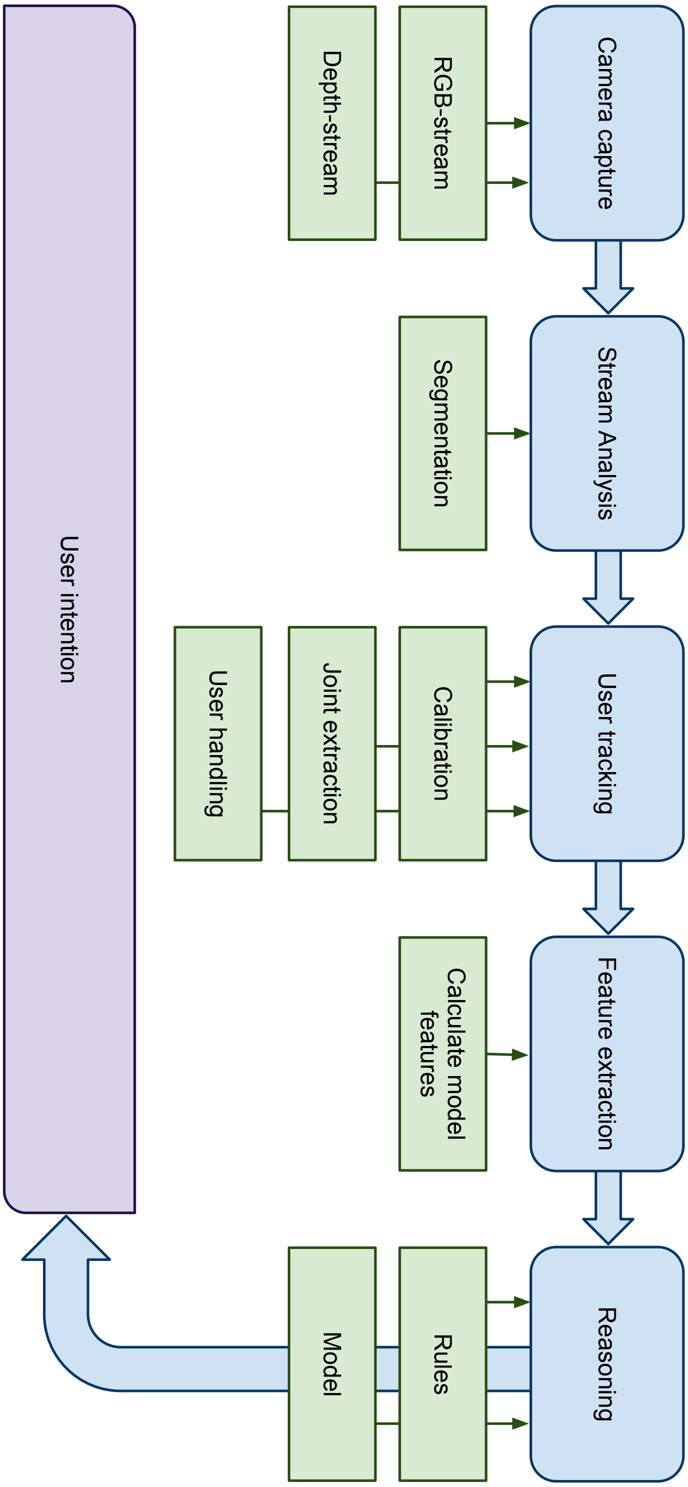


Figure 3.18: Pipeline illustrating the inference process

3.5.2 Program flow

The main program basically runs as a loop where the sensor data is updated in each step. It is then possible to retrieve the updated coordinates for each of the tracked points. For each update, the program calculates new features, adding feature sets to the corresponding user. The users are then evaluated individually by the *DecisionMaker*; if a user indicates a positive intention, an *open-signal* is sent to the door. Figure 3.19 lists the calling order of the primary functions in the program.

OpenNI requires a calibration of the new users in order to get accurate skeleton tracking. I generated a general calibration, and saved it to file. This way the users did not have to make a calibration pose to initiate tracking. Callbacks were registered for actions like *new user* and *lost user*, calling the appropriate methods in the user handler.

The operation of the door had to be processed in separate threads in order to get a non-blocking timing functionality for the door controller. Both timers and thread handling were implemented using *boost*[6].

```
while(true){
    context.WaitAndUpdateAll();
    userhandler.updateUserPoints();
    userhandler.runUserCalculations(); // calculate features
    userhandler.inferUserIntentions();
    drawOutput() // update screen
}
```

Figure 3.19: Example code: Main program loop

[6]Boost is a collection of C++ libraries speeding up development by providing portable functionality that basic C++ is lacking. See `www.boost.org` for more details.

Chapter 4

Evaluation and Conclusion

It is necessary to do some practical testing to evaluate the intention-aware sliding doors. Performance comes as a natural choice for evaluation. The performance can be judged by different criteria, but there are two aspects that must be covered:

1. *How well does the door perform compared to a traditional sliding door?*
2. *How well does the door perform compared to the real intention?*

In order to evaluate these aspects one can allow people to use the door and observe the results. However, random usage of the door might not cover the useful and interesting cases, giving low quality results. A simple and effective solution to this problem is to write a set of manuscripts that cover the cases that are necessary for a good evaluation. This includes cases where the door should open and where the door should remain closed. It is also important to explore the cases that are more unusual, where the outcome is uncertain. The manuscripts are written according to these criteria and grouped into categories. The categories group the manuscripts by how they challenge the system: *simple*, *intermediate* and *challenging*. The simple manuscripts cover basic usage and include cases that should work. The intermediate manuscripts are considered to be more complex and might pose a challenge to the system. The challenging manuscripts cover abnormal usage and cases where the outcome is uncertain. The manuscripts define specifics concerning *speed*, *acceleration*, *path*, *number of actors* and *intention*. Appendix C includes all manuscripts together with a more detailed presentation of the specifics.

In Langdridge (2006) this type of observation is described as a laboratory test, with structured observation. This kind of testing makes it easier to control the environmental variables affecting the results. It also gives a systematic collection of data. However, as Langdridge points out, it may lead to low ecological validity, meaning that the observed actions are not as natural and spontaneous as it would be in a normal environment. Using a camera and giving instructions will certainly also affect the validity. People behave differently when observed. In my situation this may lead to *overplay*, where the test subjects exaggerate their movements. In order to overcome these effects, I let the persons do each test several times. I also tried to introduce spontaneity, by changing the instructions during the tests.

4.1 Tests

Testing was done by playing out the manuscripts. Multiple people participated and they each played the scenarios several times. For every test, the results were written down; *door opened* or *door did not open*. Table 4.1 gives a summary of all the test results[1].

Table 4.1: Summary of test results. Categories denoted by roman numerals; I - Simple, II - Intermediate, III - Complicated.

	Manuscript	**Intention**	**Opened**	**¬Opened**	**Total**
I	#1	Positive	26	4	30
	#2	Positive	23	7	30
	#3	Positive	13	17	30
	#4	Negative	1	29	30
	#5	Negative	0	30	30
	SUM		**63**	**87**	**150**
II	#6	Positive	27	3	30
	#7	Positive	19	11	30
	#8	Positive	16	4	20
	#9	Positive	15	5	20
	#10	Negative	1	19	20
	#11	Negative	0	30	30
	SUM		**78**	**72**	**150**
III	#12	Positive	19	11	30
	#14	Positive	18	12	30
	#15	Negative	12	18	30
	SUM		**49**	**41**	**90**
	TOTAL		**190**	**200**	**390**

[1] Tests with Manuscript #13 were excluded, since it proved to be too difficult to perform by the test participants.

4.1.1 Measuring success

The tests are binary classification tests, giving two possible outputs, which can be correct or incorrect. In this setting it is common to use the following terms:

True positive (TP) Correctly opening; the door opened when the user wanted to go through.

True negative (TN) Correctly not opening; the door did not open when the user did not want to go through.

False positive (FP) Incorrectly opening; the door opened, even though the user did not want to go through.

False negative (FN) Incorrectly not opening; the door did not open when the user wanted it to.

A more structured view of the terms are provided by Table 4.2.

Table 4.2: Matrix displaying all possible outcomes of the test, using the defined terms. This matrix is also known as a confusion matrix

		Intention	
		Positive	Negative
Test results	Opened	**True positive**	**False positive**
	¬Opened	**False negative**	**True negative**

Based on these output classes, one can derive several quality attributes (Guda et al., 2004):

Accuracy (ACC)
Comparing correct actions to the total number of actions.

$$ACC = \frac{TP + TN}{TP + FN + TN + FP} \tag{4.1}$$

Sensitivity/True positive rate (TPR)
Is it opening when it should? TP compared to all positive intentions.

$$TPR = \frac{TP}{TP + FN} \tag{4.2}$$

Specificity/True negative rate (TNR)

Is it opening when it should not? TN compared to all negative intentions.

$$TNR = \frac{TN}{TN + FP} \tag{4.3}$$

Positive predictive value (PPV)

How good is the system at predicting positive intentions? Precision rate, TP compared to all test positives.

$$PPV = \frac{TP}{TP + FP} \tag{4.4}$$

Negative predictive value (NPV)

How good is the system at predicting negative intentions? TN compared to all test negatives.

$$PPV = \frac{TN}{TN + FN} \tag{4.5}$$

These measures are all ratios, comparing the system performance against the optimal performance. The values produced are numbers between 0 and 1 where 0 is total failure and 1 is optimal. Table 4.3 gives another view of the test results, grouped into output classes using the defined terms.

Table 4.4 lists the calculated measures. The values are presented in percent (1 is 100 %). In general, it is possible to see that the grouping by difficulty was correct; group III, the challenging manuscripts, score lower than the other two groups. However, group II has the same or better scores than group I. This might be related to some scripts being put in the wrong category.

In total, the door has an accuracy of 77.4 %, indicating that the door finds the user's intention successfully about 3 out of 4 times. The loss of accuracy is mainly because of the high rate of false negatives (FN), meaning that the door did not always open when it should have. This is also confirmed by the sensitivity and the negative predictive value (NPV), at 70.4 % and 63.0 %. The specificity and positive prediction value (PPV) are higher, at 90.0 % and 92.6 % (all at 98 % in group I and II). This indicates that the door is good at keeping closed when it should not open.

A normal sliding door suffers from the weakness that it cannot read negative intentions. In this context it means that it cannot produce any true negatives and that it will produce false positives instead. An advantage is that it produces close to no false negatives giving a high rate of true positives.

Table 4.5 lists the calculated measurements for how a traditional sliding door would perform on the manuscripts. A best/worse case is assumed, where the door

Table 4.3: Results ordered by positives and negatives.

	Manuscript	TP	TN	FP	FN
I	#1	26	-	-	4
	#2	23	-	-	7
	#3	13	-	-	17
	#4	-	29	1	-
	#5	-	30	0	-
	SUM	**62**	**59**	**1**	**28**
II	#6	27	-	-	3
	#7	19	-	-	11
	#8	16	-	-	4
	#9	15	-	-	5
	#10	-	19	1	-
	#11	-	30	0	-
	SUM	**77**	**49**	**1**	**23**
III	#12	19	-	-	11
	#14	18	-	-	12
	#15	-	18	12	-
	SUM	**37**	**18**	**12**	**23**
	TOTAL	**176**	**126**	**14**	**74**

Table 4.4: Measures of success. Values in percent.

Category	Accuracy	Sensitivity	Specificity	PPV	NPV
I	80.7	68.9	98.3	98.4	67.8
II	84.0	77.0	98.0	98.7	68.1
III	61.1	61.7	60.0	75.5	43.9
TOTAL	**77.4**	**70.4**	**90.0**	**92.6**	**63.0**

Table 4.5: Measures of success for a traditional sliding door. Values in percent.

Category	Accuracy	Sensitivity	Specificity	PPV	NPV
I	60.0	100.0	0.0	60.0	0.0
II	66.7	100.0	0.0	66.7	0.0
III	66.7	100.0	0.0	66.7	0.0
TOTAL	**64.1**	**100.0**	**0.0**	**64.1**	**0.0**

always opens, as long as there is movement within the sensor range. The results give the expected outcome with a sensitivity of 100 % and specificity of 0 %. With no true negatives, the negative predictive value is also 0 %. Since the door cannot produce neither true nor false negatives, the accuracy measure becomes equivalent with the positive predictive value. The accuracy of 64.1 % is then basically an indication of the distribution of positive/negative tests rather than the door's quality and is thereby not so useful as a general measure $\left(\frac{9 \text{ positive tests}}{14 \text{ tests}} = 64.3\ \%\right)$. However, it is possible to compare the two accuracy values, expressing that for *these tests* an intention-aware door will have a higher accuracy.

Some of the tests gave negative outputs because the system either failed to locate a person in the scene or to locate the points of interest. These results can be related to limitations in the hardware and the software library components that are out of my control, and thereby not related to the reasoning process. An *alternative view* can then be provided by removing these outcomes (both TN and FN). Tables 4.6 and 4.7 are updated to this view.

Table 4.6: Modified result table, alternative view. Negatives not related to reasoning process removed.

	Manuscript	TP	TN	FP	FN
I	#1	26	-	-	0
	#2	23	-	-	3
	#3	13	-	-	7
	#4	-	22	1	-
	#5	-	23	0	-
	SUM	**62**	**45**	**1**	**10**
II	#6	27	-	-	0
	#7	19	-	-	5
	#8	16	-	-	1
	#9	15	-	-	2
	#10	-	19	1	-
	#11	-	30	0	-
	SUM	**77**	**49**	**1**	**8**
III	#12	19	-	-	7
	#14	18	-	-	8
	#15	-	15	12	-
	SUM	**37**	**15**	**12**	**15**
	TOTAL	**176**	**109**	**14**	**33**

By removing the false negatives and the true negatives produced by limitations to the system, the values of the quality measures change. A drop in false nega-

Table 4.7: Measures of success, alternative view. Values in percent.

Category	Accuracy	Sensitivity	Specificity	PPV	NPV
I	90.7	86.1	97.8	98.4	81.8
II	93.3	90.6	98.0	98.7	86.0
III	65.8	71.2	55.6	75.5	50.0
TOTAL	**85.8**	**84.2**	**88.6**	**92.6**	**76.8**

tives (FN) results in a substantially better sensitivity and also a better negative predictive value (NPV). This drop means that the door is opening correctly more often than indicated earlier. A drop in true negatives (TN) results in a slightly worse specificity, meaning that some of the true negatives actually were produced by accident, and that the door might actually produce more false positives (FP) than indicated initially. The changes result altogether in an increased accuracy, going from 77.4 % to 85.8 %. Table 4.8 and Figure 4.1 gather the measures from the original test results, the alternative view and the traditional sliding doors.

Table 4.8: Comparing the differences. Values in percent

	Accuracy	Sensitivity	Specificity	PPV	NPV
Intention-aware	77.4	70.4	90.0	92.6	63.0
Alternative view	85.8	84.2	88.6	92.6	76.8
Traditional doors	64.1	100.0	0.0	64.1	0.0

A balanced measure (Matthews correlation coefficient)

In the test cases that I performed (provided by the manuscripts), there was an overweight of positive classifications. This is due to the fact that most of the interesting cases lie within this group. When the classes are of different sizes it is harder to get good measures for comparison. Matthews correlation coefficient is a measure considered to be more rigorous (Guda et al., 2004). Matthews (1975) estimates the quality of the prediction by calculating the correlation between the prediction and the observation[2].

The coefficient is given by (Guda et al., 2004):

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4.6}$$

The correlation coefficient gives an indication to how much better the prediction is than a random one. $MCC = 1$ indicates perfect agreement; $MCC = 0$ is

[2]In my case prediction is equal to the test result and observation is equal to the real intention

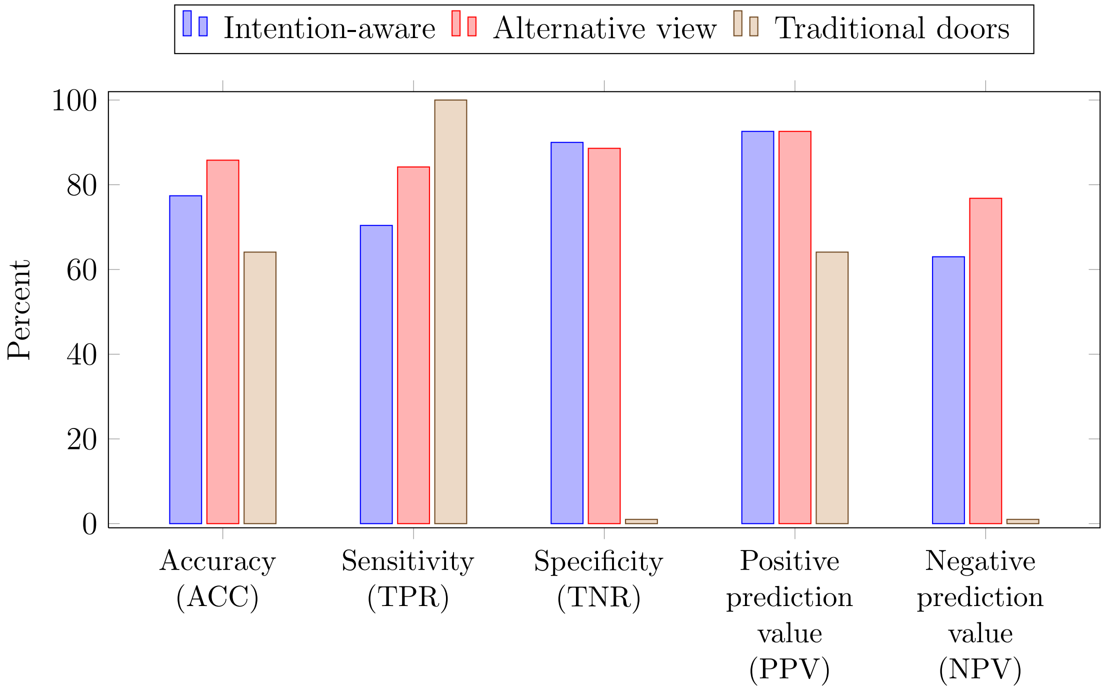


Figure 4.1: Bar chart presentation of quality measures.

expected for a random prediction and $MCC = -1$ indicates total disagreement between prediction and observation (Matthews, 1975). Table 4.9 lists the calculated coefficients[3].

Table 4.9: Matthews correlation coefficient

	Matthews correlation coefficient
Intention-aware doors	0.58
Alternative view	0.71
Traditional doors	0

Expressed with this coefficient, the quality of the traditional sliding doors is equal to a random prediction. The intention-aware sliding doors are closer to a perfect prediction with coefficients of 0.58 and 0.71.

[3]The coefficient for the traditional sliding doors could not be calculated directly using (4.6), since $TN + FN = 0$, leading to a division by zero. However, it is easy to show that MCC approaches 0 if any of the sums approaches 0 (Baldi et al., 2000).

4.2 Summary

In this project I have designed a *model of features, human behavior and intentions*. The model suggests a set of features that can be used to describe the interaction between a human being and an automated sliding door. The model also defines symbols representing value sets for the features. The symbols are then combined in order to describe different events, mapping features to intentions. This model provides a framework guiding the capturing process as well as the reasoning process.

Further, I have designed a mechanism for capturing human movement and extracting the features as suggested by the model of features, human behavior and intentions. The solution components are based on research done within computer vision, where different tools and algorithms were reviewed and evaluated. Parts of the suggested solution are provided as software libraries, while others had to be implemented. The solution includes using an Xbox Kinect as a sensor device, and the OpenNI framework together with the middleware NITE for Human body tracking and skeletal joint extraction.

A reasoning mechanism was designed, that utilizes the designed model in order to reach a conclusion about the intention of a human interacting with the door. Different reasoning techniques were reviewed in context of the sliding doors problem. Based on the review I suggest using *rule-based reasoning*. By using the events described in the model and by giving values to the different symbols I was able to form the rules for the reasoning process.

The designed mechanisms were put together in an implementation in C/C++ comprising depth and RGB image capture, body tracking, user handling and feature extraction, rule-based reasoning and door control.

A motorized sliding door was built, together with a door controller allowing a computer to interface with the door, giving open and close commands.

Finally, the door was tested both through a live demo and a laboratory style, structured observation. The door proved a superior performance to the traditional sliding doors when it came to identifying negative intentions, thus reducing the number of false positives drastically. However, both false positives and false negatives occurred, leaving room for improved accuracy.

With my solution I have managed to interpret the intention of a user interacting with an automated sliding door. I have lifted the reasoning process to a symbolic level, dealing with symbols and events easy to understand. Although the model is limited to a very specific domain, and the solution has got some limitations and weaknesses, this is a good starting point for further work.

4.3 Discussion

The final results showed that it is possible to improve the non-verbal communication between humans and computers. By interpreting human intentions the automated environment can understand its user in a better way. An automated sliding door is just one example where miscommunication is present. By giving the door controller the capability of understanding the behavior of its users, it is possible to reduce the number of cases where the door opens when it is not supposed to.

Allowing the door controller to reason, will however introduce the possibility of the door wrongly deciding not to open. Looking at the test results from my door one can see that the sensitivity was not very good (70.4 %), meaning that the door was not always opening when it should. This is also confirmed by the negative predictive value, stating that just 63 % of the cases where the door did not open were correct, and that 37 % were incorrect.

Some of the reasons for the low rates are related to limitations in the solution components and the setup. Together with the presented test results I also presented an alternative view. This view excluded the cases where the system either did not manage to detect any person in the scene or did not manage to start skeletal tracking of the person. In all these cases the door did not open because of lacking information.

The positioning of the sensor device (above the door) causes some of these limitations. The Kinect provides a field of view of 47° vertically, and in order to cover the close range, the sensor had to be tilted downwards, limiting the maximum depth (see Figure 3.2, page 31 for details). In cases where the head was out of range or arms were partly or fully occluded, the system often failed initiating tracking of the skeleton. I also observed that the skeletal tracking often failed initiating when the sensor did not detect both of the user's arms in a more or less normal position. Carrying bags or using cell phones often created problems.

If I had lowered the sensor, both the depth and height of the field of view could have been improved. This would probably have eased the skeletal tracking initiation problems. However, this would not in any way be an ideal position for a sensor as it now would be in the middle of the door opening. A solution to this could have been to put the sensor next to the door. This creates a blind spot on the opposite side of the door, but by combining multiple sensors one could expand the field of view.

The alternative view gave more promising results. With a set of false negatives (but also true negatives) removed, the accuracy increased. In the breakdown of the test results (Table 4.7, page 61) it is possible to see that the accuracy was lowest in Category III. This category includes some test cases that are difficult to determine (late change of mind) as well as giving few relevant feature sets (short time before user is in the sensor's blind spot). The decision about opening the door must be taken at a certain time; if the user changes his mind after this, there is no way of avoiding a false positive. In addition, if the sensor was able to cover the whole area in front of the door, without any blind spots under the camera, the tracking could have provided more feature sets, and the reasoning in the close cases could have been strengthened.

It is hard to prove whether better results can be achieved only by adjusting the values of the symbols, or if it is necessary to add extra symbols, rules, and possibly even extra features. It would be interesting to see if a machine learning process could find better values (and rules) for the reasoning, i.e. using decision tree learning.

The tests revealed a bug in the third-party software libraries, occurring when some of the tracking points was occluded (especially when the user walked through the door), leading to a segmentation violation, crashing the program. Since the middleware is only provided as binaries, I was not able to track down this bug. Improving the routines for skeletal tracking under occlusion is also impossible without having source code available. These aspects make it less desirable to use off the shelf components.

Regardless of what components that are used, it is possible to improve the communication between humans and computers by analyzing intentions. One of the main challenges is to provide the system with "perfect" data, covering every angle and every movement, even under partial occlusion. When testing the system with multiple actors I used two persons. Dealing with crowds of people provide even harder challenges, in this case it would be almost impossible to track each and every joint, inferring the underlying intentions.

4.4 Future Work

The solution proposed in this thesis is by no means the answer to the complex problem of interpreting human intentions. It is more of a simple, conceptual approach of merging two disciplines within computer science and artificial intelligence.

As mentioned in the Discussion, it could have been useful to explore the values of the symbols using machine learning. By learning the classification one could try to find an optimal set of values that might improve the accuracy, in general and especially in the more challenging cases. In my work the values were hard coded, and adjusted manually according to the limitations of the equipment and by trial and error.

Another interesting approach would be to use probabilistic tools, i.e. Markov models and Kalman filters, in order to predict the user's next move, but also in order to deal with noise and occlusion.

OpenNI provides a good framework, well suited for human-computer interaction. Using this, it is possible to add multiple sensors. This could remove some of the limitations that I experienced using only one Kinect, being bound to positioning it above the door, tilted downwards. Experimenting with the positioning of the sensor(s) or even adding different types of sensors could give valuable results providing better tracking abilities. It would also be possible to develop alternative implementations to the proprietary NITE middleware. This will, as well as removing bugs, allow you to control the tracking on a deeper level. This implementations can then more closely handle intentions and improve the occlusion issues described earlier.

In the last days of my thesis writing, Microsoft released their official development kit (SDK) for use with the Xbox Kinect sensor. It might be desirable to look into this if planning to continue working with a Kinect sensor. This however requires the use of Windows 7.

It is also desirable to transfer this solution to other problem areas. My solution is just a small step towards what could be much larger. Knowing the user's intention, without explicitly asking for it, is very helpful. Transferred to other tasks, this can open a wide range of possibilities. Systems can improve their functionality and efficiency greatly by reading the user's intentions, and acting upon them. Operating on a symbolic level also allows the systems to give reasonable explanations for their choices.

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.

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A. F., and Nielsen, H. (2000). Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424.

Bilgic, B., Horn, B., and Masaki, I. (2010). Fast human detection with cascaded ensembles on the GPU. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 325–332.

Caillette, F. and Howard, T. (2004). Real-Time Markerless Human Body Tracking Using Colored Voxels and 3-D Blobs. *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 266–267.

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., Nedevschi, S., and Bota, S. (2009). Real time stereo vision based pedestrian detection using full body contours. *2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing*, pages 79–86.

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.

Horaud, R., Niskanen, M., Dewaele, G., and Boyer, E. (2009). Human Motion Tracking by Registering an Articulated Surface to 3D Points and Normals. *Journal of Pharmacy and Pharmacology*, 61(7):27–31.

Intel (2007). *Power Supply Design Guide for Desktop Platform Form Factors*. Intel Corporation. `http://www.formfactors.org/developer%5Cspecs%5CPSU_DG_rev_1_1.pdf` (accessed 07.06.2011).

Jia, H.-X. and Zhang, Y.-J. (2007). Fast human detection by boosting histograms of oriented gradients. In *Image and Graphics, 2007. ICIG 2007. Fourth International Conference on*, pages 683–688.

Kofod-Petersen, A., Wegener, R., and Cassens, J. (2009). Closed doors – modelling intention in behavioural interfaces. In Kofod-Petersen, A., Langseth, H., and Gundersen, O. E., editors, *Proceedings of the first Norwegian Artificial Intelligence Symposium*, pages 93–102, Trondheim, Norway. Tapir Akademiske Forlag.

Langdridge, D. (2006). *Psykologisk forskningsmetode*. Tapir Akademisk Forlag. Norwegian translation of Introduction to Research Methods and Data Analysis in Psychology 01.

Lienhart, R. and Maydt, J. (2002). An extended set of Haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages 900–903.

Lillywhite, K., Lee, D.-J., and Zhang, D. (2009). Real-time human detection using histograms of oriented gradients on a GPU. In *Applications of Computer Vision (WACV), 2009 Workshop on*, pages 1–6.

Martinec, R. (2001). Interpersonal resources in action. *Semiotica*, 2001(135):117–145.

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451.

Mitchell, T. (1997). *Machine Learning*. McGraw Hill.

Moore, A. R. (2008). Surgical teams in action: a contextually sensitive approach to modelling body alignment and interpersonal engagement.

OpenNI User Guide (2010). *OpenNI User Guide*. `http://www.openni.org/images/stories/pdf/OpenNI_UserGuide.pdf` (accessed 13.04.2011).

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. (2009). *Artificial Intelligence: a Modern Approach*. Prentice hall.

Sheldon, R. (2008a). Controlling inductive devices. `http://www.relaycontrollers.com/Relay/Device/A0009` (accessed 06.04.2011).

Sheldon, R. (2008b). Important notice. `http://www.relaycontrollers.com/i.htm` (accessed 06.04.2011).

Sonka, M., Hlaváč, V., and Boyle, R. (2008). *Image processing, analysis, and machine vision*. Thomson.

Sugano, H., Miyamoto, R., and Nakamura, Y. (2010). Optimized parallel implementation of pedestrian tracking using HOG features on GPU. In *Ph.D. Research in Microelectronics and Electronics (PRIME), 2010 Conference on*, pages 1–4.

Toya, A., Hu, Z., Yoshida, T., Uchimura, K., Kubota, H., and Ono, M. (2008). Pedestrian recognition using stereo vision and histogram of oriented gradients. In *Vehicular Electronics and Safety, 2008. ICVES 2008. IEEE International Conference on*, pages 57–62.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages 511–518.

Wang, Q. J. and Zhang, R. B. (2008). LPP-HOG: A new local image descriptor for fast human detection. In *Knowledge Acquisition and Modeling Workshop, 2008. KAM Workshop 2008. IEEE International Symposium on*, pages 640–643.

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.

Appendices

Appendix A

NSD

NSD (Norwegian Social Science Data Services) is the organ for norwegian universities, that handles some of the responsibility delegated by the Data Inspectorate on issues concerning personal information and privacy. Because of my usage of a camera in a situation that involved other people, I had to notify this organ and get an approval that allowed me to collect the type of data I needed in order to test the performance of my door.

A.1 Notification

This notification is a print out of the electronically submitted notification form. The notification gives detailed specification of the personal information related activities.

MELDESKJEMA

Meldeskjema (versjon 1.1) for forsknings- og studentprosjekt som medfører meldeplikt eller konsesjonsplikt (jf. personopplysningsloven og helseregisterloven med forskrifter).

1. Prosjekttittel

Tittel	Intention-based sliding doors	

2. Behandlingsansvarlig institusjon

Institusjon	NTNU	Velg den institusjonen du er tilknyttet. Alle nivå må oppgis. Ved studentprosjekt er det studentens tilknytning som er avgjørende. Dersom din institusjon ikke finnes på listen, ta kontakt med personvernombudet.
Avdeling/Fakultet	Fakultet for informasjonsteknologi, matematikk og elektroteknikk	
Institutt	Institutt for datateknikk og informasjonsvitenskap	

3. Daglig ansvarlig (forsker, veileder)

Fornavn	Anders	Før opp navn på den som har det daglige ansvaret for prosjektet. For studentprosjekt er daglig ansvarlig vanligvis veileder.
Etternavn	Kofod-Petersen	
Akademisk grad	Høyere grad	Veileder og student må være tilknyttet samme institusjon. Dersom studenten har ekstern veileder, kan biveileder eller fagansvarlig stå som daglig ansvarlig.
Stilling	Førsteamanuensis II	
Arbeidssted	Norges teknisk-naturvitenskapelige universitet	Arbeidssted må være i tilknytning til behandlingsansvarlig institusjon, f.eks. underavdeling, institutt etc.
Adresse (arb.sted)	IT-bygget, Sem Sælands vei 9	
Postnr/sted (arb.sted)	7491 Trondheim	
Telefon/mobil (arb.sted)	73591717 /	NB! All korrespondanse går via e-post. Det er derfor viktig at du oppgir korrekt e-postadresse. Det bør være en adresse som du bruker aktivt over tid. Husk å gi beskjed dersom den endres.
E-post	anderpe@idi.ntnu.no	

4. Student

Studentprosjekt	Ja ● Nei ○	NB! All korrespondanse går via e-post. Det er derfor viktig at du oppgir en korrekt e-postadresse. Det bør videre være en adresse du bruker aktivt over tid. Husk å gi beskjed dersom den endres.
Fornavn	John Sverre	
Etternavn	Solem	
Akademisk grad	Høyere grad	
Privatadresse	[redacted]	
Postnr/sted (privatadresse)	7031 Trondheim	
Telefon/mobil	[redacted]	
E-post	johnsvs@stud.ntnu.no	

5. Formålet med prosjektet

Prosjektets formål	Mastergradsprosjekt. Formålet med prosjektet er å utvikle en "intelligent" automatisert dør. Denne døra skal ikke åpnes ved hjelp av tradisjonell bevegelsessensor, men basert på intensjonen til brukeren av døra. Dette innebærer å utstyre en dør (eksisterende eller konstruert) med kamera for automatisk analyse av personbevegelser i dørområdet.	Redegjør kort for prosjektets formål, problemstilling, forskningsspørsmål e.l. Maks 750 tegn.

6. Prosjektomfang

Velg omfang	● Enkel institusjon ○ Nasjonal multisenterstudie ○ Internasjonal multisenterstudie	Med multisenterstudier forstås her forskningsprosjekter som gjennomføres ved flere institusjoner samtidig, som har samme formål og hvor det utveksles/deles personopplysninger mellom
Oppgi øvrige institusjoner		

Hvordan foregår samarbeidet mellom institusjonene? Hvem har tilgang til personopplysninger og hvordan reguleres tilgangen?		deltakende institusjoner. Les mer om hva personopplysninger er

7. Utvalgsbeskrivelse

Beskrivelse av utvalget	Tilfeldig utvalg av personer som bruker automatiserte skyvedører	Med utvalg menes dem som deltar i undersøkelsen eller dem det innhentes opplysninger om. F. eks. et representativt utvalg av befolkningen, skoleelever med lese- og skrivevansker, pasienter, innsatte.
Rekruttering og trekking	Rekruttering av studenter/ansatte ved universitetet. Aktuelt både ved å hente inn folk for å bruke dør, og ved å la forbipasserende bruke dør.	Beskriv hvordan utvalget trekkes/rekrutteres. Utvalget kan trekkes fra registre, f. eks. folkeregisteret, NAV, pasientregistre, eller rekrutteres gjennom f.eks. en bedrift, skole, idrettsmiljø, eget nettverk. Oppgi hvem som foretar trekkingen/rekrutteringen.
Førstegangskontakt	I utgangspunktet er det	Oppgi hvem som oppretter førstegangskontakt med utvalget og beskriv hvordan den opprettes. Les mer om førstegangskontakt
Alder på utvalget	□ Barn (0-15 år) □ Ungdom (16-17 år) ■ Voksne (over 18 år)	
Antall personer som inngår i utvalget	Ikke spesifisert	
Inkluderes det myndige personer med redusert eller manglende samtykkekompetanse?	Ja ○ Nei ●	Redegjør for hvorfor det er nødvendig å inkludere myndige personer med redusert eller manglende samtykkekompetanse. Les mer om inklusjon av myndige personer med redusert eller manglende samtykkekompetanse
Hvis ja, beskriv		

8. Metode for innsamling av personopplysninger

Kryss av for hvilke datainnsamlingsmetoder og datakilder som skal benyttes	□ Spørreskjema □ Personlig intervju □ Gruppeintervju □ Observasjon □ Psykologiske/pedagogiske tester □ Medisinske undersøkelser/tester □ Journaldata □ Registerdata ■ Annen innsamlingsmetode	Personopplysninger kan innhentes direkte fra den registrerte og/eller fra ulike journaler (NAV, PPT, sykehus, bofellesskap og lignende) eller eksisterende registre (f.eks. Statistisk sentralbyrå, Kreftregisteret).
Annen innsamlingsmetode, oppgi hvilken	kameraovervåkning	
Kommentar til metode for innsamling av personopplysninger	I utgangspunktet skal ikke videomateriale lagres. Midlertidig lagring kan allikevel være nødvendig for manuell analyse.	

9. Datamaterialets innhold

Gjør rede for hvilke opplysninger som samles inn	Område foran dør videoovervåkes ved hjelp av kamera montert i overkant av døra.	Spørreskjema, intervjuguide/temaliste, m.m. legges ved meldeskjemaet til slutt.
Samles det inn direkte personidentifiserende opplysninger?	Ja ○ Nei ●	Les mer om hva personopplysninger er NB! Selv om resultatene i den endelige publikasjonen vil være anonymisert, må det krysses av her dersom direkte eller indirekte personidentifiserende opplysninger registreres i datamaterialet underveis i prosjektet.
Hvis ja, hvilke?	□ Navn □ Fødselsdato □ 11-sifret fødselsnummer □ Adresse og/eller e-postadresse og/eller telefonnummer	
Spesifiser hvilke		

Samles det inn indirekte personidentifiserende opplysninger?	Ja ● Nei ○	En person vil være indirekte identifiserbar dersom det er mulig å identifisere vedkommende gjennom bakgrunnsopplysninger som for eksempel bostedskommune eller arbeidsplass/skole kombinert med opplysninger som alder, kjønn, yrke, diagnose, etc.
Hvis ja, hvilke?	Ansikt kan til tider være synlige på videomateriale	
Samles det inn sensitive personopplysninger?	Ja ○ Nei ●	
Hvis ja, oppgi hvilke	□ Rasemessig eller etnisk bakgrunn, eller politisk, filosofisk eller religiøs oppfatning □ At en person har vært mistenkt, siktet, tiltalt eller dømt for en straffbar handling □ Helseforhold □ Seksuelle forhold □ Medlemskap i fagforeninger	
Samles det inn opplysninger om tredjeperson?	Ja ○ Nei ●	Med opplysninger om tredjeperson menes opplysninger som kan spores tilbake til personer som ikke inngår i utvalget. Eksempler på tredjeperson er kollega, elev, klient, familiemedlem.
Hvis ja, hvem er tredjeperson og hvilke opplysninger registreres?		
Hvordan blir tredjeperson informert om behandlingen?	□ Skriftlig informasjon □ Muntlig informasjon □ Blir ikke informert	
Blir ikke informert, redegjør hvorfor		

10. Informasjon og samtykke

Oppgi hvordan informasjon til utvalget gis	■ Skriftlig informasjon □ Muntlig informasjon □ Ingen informasjon	Som hovedregel skal det gis informasjon og innhentes samtykke fra den registrerte. Dersom informasjon gis skriftlig, legg ved kopi av informasjonsskriv. Dersom det ikke skal gis informasjon, må dette redegjøres for. Les mer om hvilken informasjon som bør gis til utvalget
Redegjør	Skilt som redegjør for prosjektet samt for videoovervåking synlig ved dør	
Oppgi hvordan samtykke innhentes	□ Skriftlig samtykke □ Muntlig samtykke ■ Innhentes ikke samtykke	Dersom det benyttes skriftlig samtykke, anbefales det at dette følger i teksten etter informasjonen. Dersom det ikke skal innhentes samtykke, må dette redegjøres for. Les mer om krav til gyldig samtykke
Innhentes ikke samtykke, redegjør	Personer gir samtykke ved å gå inn i dørens område	

11. Informasjonssikkerhet

Direkte personidentifiserende opplysninger erstattes med et referansenummer som viser til en atskilt navneliste	Ja ○ Nei ●	Direkte personidentifiserende opplysninger bør ikke registreres sammen med det øvrige datamaterialet.
Hvordan lagres listen/koblingsnøkkelen og hvem har tilgang til den?		
Direkte personidentifiserende opplysninger lagres sammen med det øvrige materialet	Ja ○ Nei ●	
Hvorfor er det nødvendig med oppbevaring av direkte identifikasjonsopplysninger sammen med det øvrige datamaterialet?		
Lagres direkte personidentifiserbare opplysninger på andre måter?	Ja ○ Nei ●	

Spesifiser		
Hvordan registreres og oppbevares datamaterialet?	■ Fysisk isolert PC tilhørende virksomheten □ PC i nettverkssystem tilhørende virksomheten ■ PC i nettverkssystem tilknyttet Internett tilhørende virksomheten □ Fysisk isolert privat PC □ Privat PC tilknyttet Internett ■ Videoopptak/fotografi □ Lydopptak □ Manuelt/papir □ Annen registreringsmetode	Sett flere kryss dersom opplysningene registreres på flere måter.
Annen registreringsmetode beskriv nærmere		
Behandles og/eller lagres lyd- og videoopptak og/eller fotografi på PC?	Ja ● Nei ○	
Hvordan er datamaterialet beskyttet mot at uvedkommende får innsyn i opplysningene?	PC er beskyttet med brukernavn/passord. PC har heller ikke skjerm, mus eller tastatur under innsamlingen av data.	Er f.eks. PC-tilgangen beskyttet med brukernavn og passord, og står PC-en i et låsbart rom?
Dersom det benyttes mobil lagringsenhet (bærbar PC, minnepenn, minnekort, cd, ekstern harddisk), oppgi hvilken type, og redegjør for hvorfor det benyttes mobil lagringsenhet		
Skal prosjektet ha medarbeidere som vil få tilgang til datamaterialet på lik linje med daglig ansvarlig/student?	Ja ○ Nei ●	
Hvis ja, hvem?		
Innhentes eller overføres personopplysninger ved hjelp av e-post/Internett?	Ja ○ Nei ●	
Hvis ja, oppgi hvilke opplysninger		
Vil personopplysninger bli utlevert til andre enn prosjektgruppen?	Ja ○ Nei ●	
Hvis ja, til hvem?		
Skal opplysningene samles inn/bearbeides av en databehandler?	Ja ○ Nei ●	Med databehandler menes en som samler inn og/eller behandler personopplysninger på vegne av den behandlingsansvarlige. Eksempler på ofte brukte databehandlere er Questback, Synovate MMI, Norfakta etc. Les mer om databehandleravtaler her
Hvis ja, hvilken?		

12. Vurdering/godkjenning fra andre instanser

Søkes det dispensasjon fra taushetsplikten for å få tilgang til data?	Ja ○ Nei ●	For å få tilgang til taushetsbelagte opplysninger fra f.eks. NAV, PPT, sykehus, må det søkes om dispensasjon fra taushetsplikten. Dispensasjon søkes vanligvis fra aktuelt departement. For dispensasjon fra taushetsplikten for helseopplysninger skal det for alle typer forskning søkes Regional komité for medisinsk og helsefaglig forskningsetikk
Kommentar		
Skal det innhentes godkjenning/tillatelse fra andre instanser?	Ja ● Nei ○	Det kan f. eks. være aktuelt å søke tillatelse fra registereier for tilgang til data, ledelsen for tilgang til forskning i firma, etc.
Hvis ja, hvilke?	Ledelse ved universitet, for tillatelse til å drive videoovervåkning.	

13. Prosjektperiode

Prosjektperiode	Prosjektstart:01/03/2011 Prosjektslutt:20/06/2011	Prosjektstart Tidspunkt for når førstegangskontakt opprettes og/eller datainnsamlingen starter. Prosjektslutt Tidspunkt for når datamaterialet skal anonymiseres, slettes, eller arkiveres i påvente av oppfølgingsstudier. Dette sammenfaller gjerne med publisering og ferdigstilling av oppgave, avhandling eller rapport.
Hva skal skje med datamaterialet ved prosjektslutt?	■ Datamaterialet skal anonymiseres □ Datamaterialet skal oppbevares med personidentifikasjon	Med anonymisering menes at det ikke lenger er mulig å føre opplysningene tilbake til enkeltpersoner i datamaterialet. Les mer om anonymisering
Hvordan skal datamaterialet anonymiseres?	All innsamlet data slettes ved prosjektslutt.	Hovedregel for lagring av data med personidentifikasjon er samtykke fra den registrerte. Årsaker til oppbevaring kan være konkrete oppfølgningsstudier, undervisningsformål eller annet. Datamaterialet kan lagres ved egen institusjon, offentlig arkiv eller annet. Les mer om arkivering
Hvorfor skal datamaterialet oppbevares med personidentifikasjon?		
Hvor skal datamaterialet oppbevares, og hvor lenge?		

14. Finansiering

Hvordan finansieres prosjektet?	Egne ressurser	

15. Tilleggsopplysninger

Tilleggsopplysninger		

16. Vedlegg

Antall vedlegg	1	

A.2 Concession

This section includes the response from NSD allowing me to collect the data as requested.

Norsk samfunnsvitenskapelig datatjeneste AS
NORWEGIAN SOCIAL SCIENCE DATA SERVICES

NSD

Harald Hårfagres gate 29
N-5007 Bergen
Norway
Tel: +47-55 58 21 17
Fax: +47-55 58 96 50
nsd@nsd.uib.no
www.nsd.uib.no
Org.nr. 985 321 884

Anders Kofod-Petersen
Institutt for datateknikk og informasjonsvitenskap
NTNU
Sem Sælandsvei 7-9
7491 TRONDHEIM

Vår dato: 16.03.2011 Vår ref: 26259 / 3 / TNS Deres dato: Deres ref:

KVITTERING PÅ MELDING OM BEHANDLING AV PERSONOPPLYSNINGER

Vi viser til melding om behandling av personopplysninger, mottatt 03.02.2011. Meldingen gjelder prosjektet:

26259	*Intention-based Sliding Doors*
Behandlingsansvarlig	*NTNU, ved institusjonens øverste leder*
Daglig ansvarlig	*Anders Kofod-Petersen*
Student	*John Sverre Solem*

Personvernombudet har vurdert prosjektet og finner at behandlingen av personopplysninger er meldepliktig i henhold til personopplysningsloven § 31. Behandlingen tilfredsstiller kravene i personopplysningsloven.

Personvernombudets vurdering forutsetter at prosjektet gjennomføres i tråd med opplysningene gitt i meldeskjemaet, korrespondanse med ombudet, eventuelle kommentarer samt personopplysningsloven/-helseregisterloven med forskrifter. Behandlingen av personopplysninger kan settes i gang.

Det gjøres oppmerksom på at det skal gis ny melding dersom behandlingen endres i forhold til de opplysninger som ligger til grunn for personvernombudets vurdering. Endringsmeldinger gis via et eget skjema, http://www.nsd.uib.no/personvern/forsk_stud/skjema.html. Det skal også gis melding etter tre år dersom prosjektet fortsatt pågår. Meldinger skal skje skriftlig til ombudet.

Personvernombudet har lagt ut opplysninger om prosjektet i en offentlig database, http://www.nsd.uib.no/personvern/prosjektoversikt.jsp.

Personvernombudet vil ved prosjektets avslutning, 20.06.2011, rette en henvendelse angående status for behandlingen av personopplysninger.

Vennlig hilsen

Bjørn Henrichsen Tone Njølstad Slotsvik

Kontaktperson: Tone Njølstad Slotsvik tlf: 55 58 24 10
Vedlegg: Prosjektvurdering
Kopi: John Sverre Solem, Alfred Larsens gate 13, 7031 TRONDHEIM

Personvernombudet for forskning

Prosjektvurdering - Kommentar

Prosjektnr: 26259

Formålet med prosjektet er å utvikle en "intelligent" automatisert dør.

Utvalget består av et uvisst antall personer som er hentet inn for å teste døren eller som er tilfeldig forbipasserende.

Datamaterialet samles inn gjennom kameraovervåkning av området ved døren.

Skriftlig informasjon om prosjektet vil være tilgjengelig for dem som passerer døren. Personvernombudet finner at informasjonsskrivet er tilfredsstillende utformet, forutsatt at anbefaling i epost fra ombudet 24.02.2011 følges. Informasjonsskrivet må henge synlig slik at alle som er i området rundt døren har anledning til å lese det. Passering gjennom døren regnes som samtykke til å bli filmet. Ombudet forutsetter at dette innebærer at de som ikke ønsker å bli filmet har andre muligheter til å komme seg dit de vil på, jf. epost fra ombudet 24.02.2011.

Senest ved prosjektslutt 20.06.2011 skal datamaterialet slettes.

A.3 Consent

When collecting personal data, it is important that this is done with the consent of the involved persons. This section includes the poster informing the test users about the collection and handling of data.

KAMERA-
OVERVÅKNING

Prosjekt: Intensjons-basert dør

Denne døra er en del av et masterprosjekt, som har som formål å konstruere en dør som responderer på ditt ønske om å gå gjennom den.

Du er velkommen til å gå gjennom døra, og/eller bevege deg i området rundt døra for å teste hvorvidt den faktisk åpner seg når du skal gå gjennom den.

For å muliggjøre dette er døra utstyrt med kamera og datamaskin som analyserer ditt bevegelsesmønster i området rundt den.

Samtykke

Ved å gå inn i området samtykker du i at du blir video-overvåket. Video brukes direkte av systemet for å åpne døren. Det kan være nødvendig å ta vare på video for evaluering av dørens prestasjon. Alle innsamlede data vil bli slettet ved prosjektslutt (juni 2011).

John Sverre Solem
johnsvs@stud.ntnu.no
Student

Anders Kofod-Petersen
anderpe@idi.ntnu.no
Veileder

Appendix B

Door schematics

In this chapter, some of the schematics of the door used during the building process are included. For convenience a total height for the construction was set in order to fit under a standard ceiling height of 2.40 meters, this gave a limited height of the door opening of about 2 meters. A width of 2.50 meters was also chosen to limit the total size.

The construction must be portable, and can therefore be disassembled into pieces of manageable size. The frame consists of two pockets, mounted together by a top case, holding the drive system and rails, and a floor board. Four feet support the frame in order to prevent it from tipping over. Figure B.1 shows the pockets together with the top case. Figure B.2 shows a cross section of the top case. The top case is designed in such a way that it can support the doors, providing rails for the wheels, as well as giving a separate room for the drive system on top. Figure B.3 illustrates the function of the drive system, including the belt, cog wheels and doors.

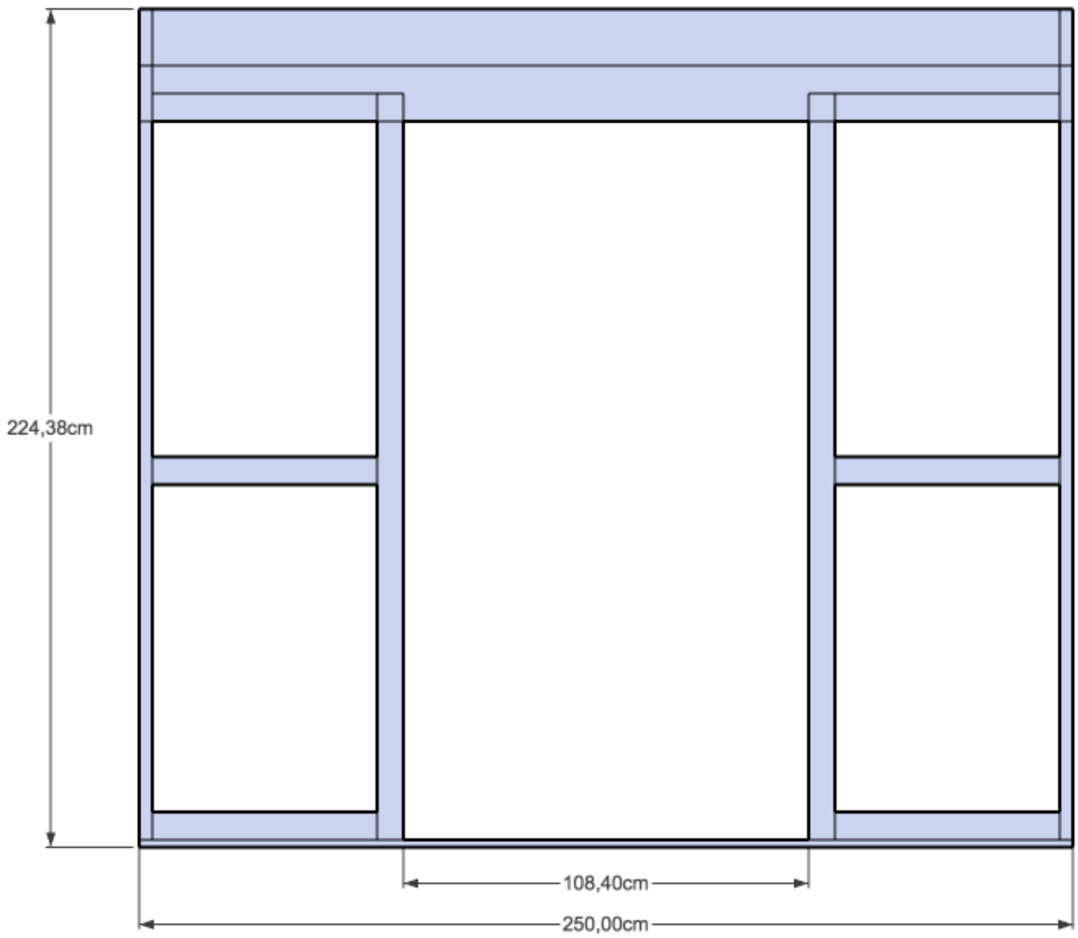


Figure B.1: Front view of the door frame

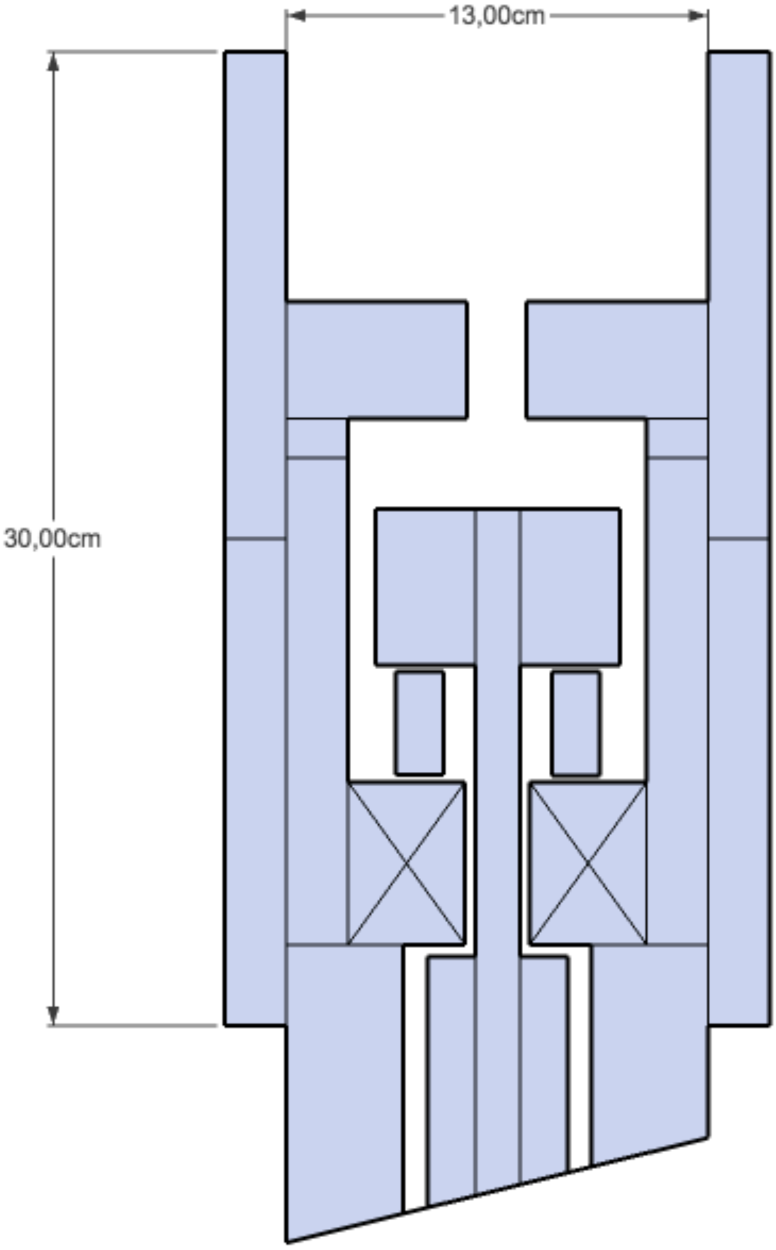


Figure B.2: Cross-section of door frame with door

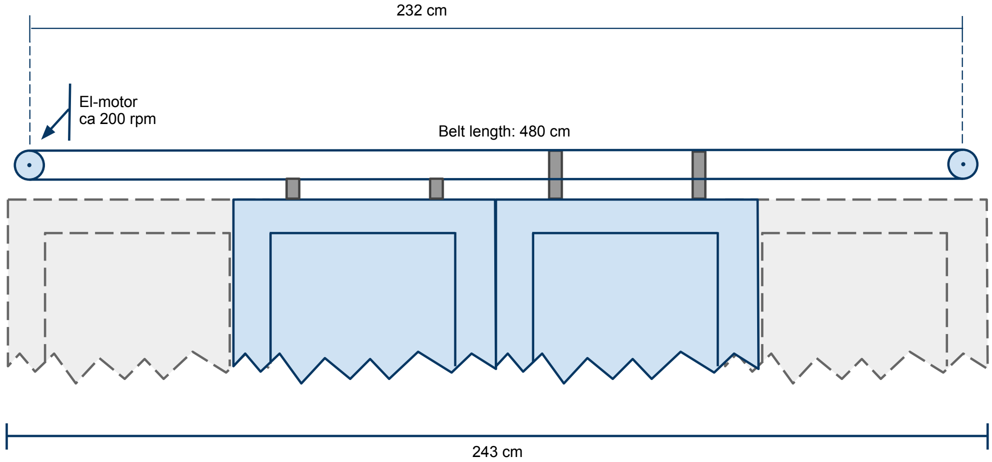


Figure B.3: Belt and wheels

Appendix C

Manuscripts

The manuscripts are written to provide a systematic approach to the testing of the sliding door. When testing the door, it is important to get a broad perspective covering some odd cases as well as the most frequently used cases. Random usage could also have provided results, but then you lose control of which situations that are played out, and the width of the cases is not guaranteed within a reasonable amount of time. The manuscripts in this section are written to cover what may be considered as normal use, bordering to abnormal. They are grouped in categories according to the challenge they provide. The manuscripts cover both positive cases, where the user wants to go through the door, and negative cases where the user does not want to go through the door. For all manuscripts the following parameters are defined:

Category The challenge the case poses to the system – Simple, Intermediate, Challenging.

Actors The number of actors involved in the scene.

Intention The actors intention of walking through the door – Positive or Negative.

Speed How the walk is done – Normal or Fast

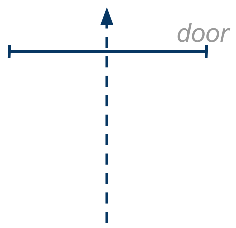
Acceleration How the speed changes throughout the case

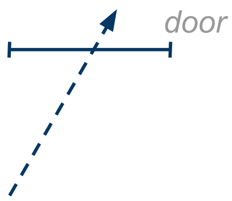
Path The direction of walking – Straight, Changing etc.

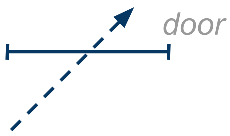
Description Detailed instructions on how the case must be played.

Simple

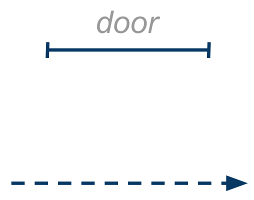
Positive

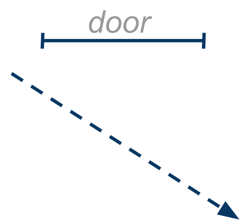
<table>
<tr><th colspan="4">Manuscript 1</th></tr>
<tr><td rowspan="4">
</td><th>Category</th><th>Actors</th><th>Intention</th></tr>
<tr><td>Simple</td><td>1</td><td>Positive</td></tr>
<tr><th>Speed</th><th>Acceleration</th><th>Path</th></tr>
<tr><td>Normal</td><td>None</td><td>Straight</td></tr>
<tr><td colspan="4">Description: A person walks with normal speed straight towards the door, intending to walk through it.</td></tr>
</table>

<table>
<tr><th colspan="4">Manuscript 2</th></tr>
<tr><td rowspan="4">
</td><th>Category</th><th>Actors</th><th>Intention</th></tr>
<tr><td>Simple</td><td>1</td><td>Positive</td></tr>
<tr><th>Speed</th><th>Acceleration</th><th>Path</th></tr>
<tr><td>Normal</td><td>None</td><td>Straight</td></tr>
<tr><td colspan="4">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° of the z-axis.</td></tr>
</table>

<table>
<tr><th colspan="4">Manuscript 3</th></tr>
<tr><td rowspan="4">
</td><th>Category</th><th>Actors</th><th>Intention</th></tr>
<tr><td>Simple</td><td>1</td><td>Positive</td></tr>
<tr><th>Speed</th><th>Acceleration</th><th>Path</th></tr>
<tr><td>Normal</td><td>None</td><td>Straight</td></tr>
<tr><td colspan="4">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° of the z-axis.</td></tr>
</table>

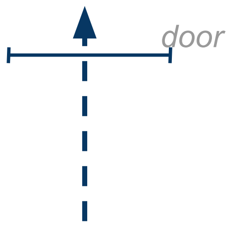
Negative

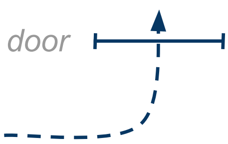
<table>
<tr><th colspan="4">Manuscript 4</th></tr>
<tr><td rowspan="4">
</td><th>Category</th><th>Actors</th><th>Intention</th></tr>
<tr><td>Simple</td><td>1</td><td>Negative</td></tr>
<tr><th>Speed</th><th>Acceleration</th><th>Path</th></tr>
<tr><td>Normal</td><td>None</td><td>Straight</td></tr>
<tr><td colspan="4">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.</td></tr>
</table>

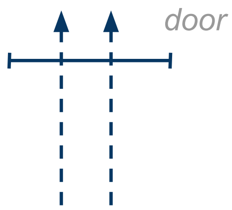
<table>
<tr><th colspan="4">Manuscript 5</th></tr>
<tr><td rowspan="4">
</td><th>Category</th><th>Actors</th><th>Intention</th></tr>
<tr><td>Simple</td><td>1</td><td>Negative</td></tr>
<tr><th>Speed</th><th>Acceleration</th><th>Path</th></tr>
<tr><td>Normal</td><td>None</td><td>Straight</td></tr>
<tr><td colspan="4">Description: A person is standing next to the door. He walks diagonally across the scene, away from the door.</td></tr>
</table>

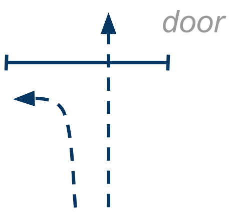
Intermediate

Positive

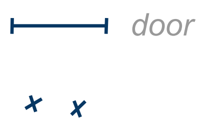
Manuscript 6			
	**Category**	**Actors**	**Intention**
	Intermediate	1	Positive
	Speed	**Acceleration**	**Path**
	Fast	None	Straight
Description: *A person walks quickly straight towards the door, intending to walk through it.*			

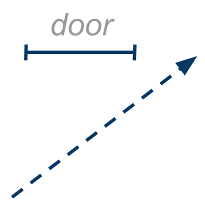
Manuscript 7			
	**Category**	**Actors**	**Intention**
	Intermediate	1	Positive
	Speed	**Acceleration**	**Path**
	Normal	Changing	Changing
Description: *A person walks parallel to the door and is suddenly told to walk through the door, simulating changing one's mind.*			

<table>
<tr><th colspan="4">Manuscript 8</th></tr>
<tr><td rowspan="4"></td><th>Category</th><th>Actors</th><th>Intention</th></tr>
<tr><td>Intermediate</td><td>2</td><td>Positive</td></tr>
<tr><th>Speed</th><th>Acceleration</th><th>Path</th></tr>
<tr><td>Normal</td><td>None</td><td>Straight</td></tr>
<tr><td colspan="4">Description: A pair walks together in a straight line, at normal speed, along the z-axis, both intending to walk through the door.</td></tr>
</table>

<table>
<tr><th colspan="4">Manuscript 9</th></tr>
<tr><td rowspan="4"></td><th>Category</th><th>Actors</th><th>Intention</th></tr>
<tr><td>Intermediate</td><td>2</td><td>Positive</td></tr>
<tr><th>Speed</th><th>Acceleration</th><th>Path</th></tr>
<tr><td>Normal</td><td>Mixed</td><td>Mixed</td></tr>
<tr><td colspan="4">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.</td></tr>
</table>

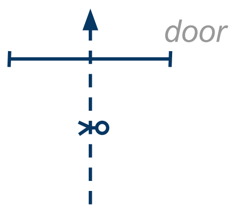
Negative

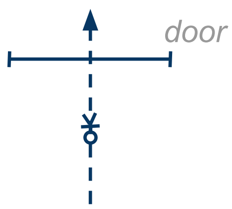
<table>
<tr><th colspan="4">Manuscript 10</th></tr>
<tr><td rowspan="4">
</td><th>Category</th><th>Actors</th><th>Intention</th></tr>
<tr><td>Intermediate</td><td>2</td><td>Negative</td></tr>
<tr><th>Speed</th><th>Acceleration</th><th>Path</th></tr>
<tr><td>None</td><td>None</td><td>Standing</td></tr>
<tr><td colspan="4">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.</td></tr>
</table>

<table>
<tr><th colspan="4">Manuscript 11</th></tr>
<tr><td rowspan="4">
</td><th>Category</th><th>Actors</th><th>Intention</th></tr>
<tr><td>Intermediate</td><td>1</td><td>Negative</td></tr>
<tr><th>Speed</th><th>Acceleration</th><th>Path</th></tr>
<tr><td>Normal</td><td>None</td><td>Straight</td></tr>
<tr><td colspan="4">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.</td></tr>
</table>

Complicated

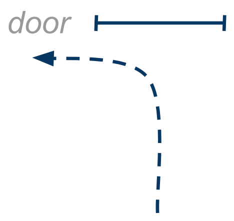
Positive

<table>
<tr><th colspan="4">Manuscript 12</th></tr>
<tr><td rowspan="4">
</td><th>Category</th><th>Actors</th><th>Intention</th></tr>
<tr><td>Complicated</td><td>1</td><td>Positive</td></tr>
<tr><th>Speed</th><th>Acceleration</th><th>Path</th></tr>
<tr><td>Normal</td><td>None</td><td>Straight</td></tr>
<tr><td colspan="4">Description: A person walks sideways in a straight line, along the z-axis, with the intention of walking through the door.</td></tr>
</table>

<table>
<tr><th colspan="4">Manuscript 13</th></tr>
<tr><td rowspan="4">
</td><th>Category</th><th>Actors</th><th>Intention</th></tr>
<tr><td>Complicated</td><td>1</td><td>Positive</td></tr>
<tr><th>Speed</th><th>Acceleration</th><th>Path</th></tr>
<tr><td>Normal</td><td>None</td><td>Straight</td></tr>
<tr><td colspan="4">Description: A person walks backwards in a straight line, along the z-axis, with the intention of walking through the door.</td></tr>
</table>

Manuscript 14			
	**Category**	**Actors**	**Intention**
	Complicated	1	Positive
	Speed	**Acceleration**	**Path**
	Low	None	Turning
***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.*			

Negative

<table>
<tr><th colspan="4">Manuscript 15</th></tr>
<tr><td rowspan="4">
</td><th>Category</th><th>Actors</th><th>Intention</th></tr>
<tr><td>Complicated</td><td>1</td><td>Negative</td></tr>
<tr><th>Speed</th><th>Acceleration</th><th>Path</th></tr>
<tr><td>Normal</td><td>Changing</td><td>Changing</td></tr>
<tr><td colspan="4">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.</td></tr>
</table>

Appendix D

SCAI Poster

The 11th Scandinavian Conference on Artificial Intelligence was held during my thesis work. I participated in this conference with a *poster* and a *demonstration* of the door. A downscaled version of the poster is included in this appendix. The original size is A0.

Figure D.1: Getting the door ready for SCAI

Intention-based Sliding Doors

John Sverre Solem, Håvar Aambø Fosstveit, Richard E. Blake and Anders Kofod-Pedersen

Norwegian University of Science and Technology

johnsvs@stud.ntnu.no

Introduction

When a person is interacting with an environment, a non-verbal body language is used. This language often expresses **intentions**. The intentions are not always shown explicitly. However, human beings can easily read this intentions. If the environment was able to read these intentions as well, miscommunication could be avoided. We attempt to explore this, using **sliding doors** as an example.

Sliding doors are today mostly based on naive motion sensing, and hence not very intelligent in deciding to open or not. Ignoring the user's intention, can result in a miscommunication between the door and the user, most often leading to erroneous openings the door.

We propose a solution to this problem:

- Capture human activity in front of a door using cameras.
- Recognize and track users by processing RGB and range data.
- Extract features according to a model of human behavior and intention.
- Make a decision using a rule-based reasoning mechanism.

1. Modeling human behavior

PERSONS wanting to go though a door, display a behavior signaling their intentions. This can easily be captured by human beings, but how can a computer do the same?

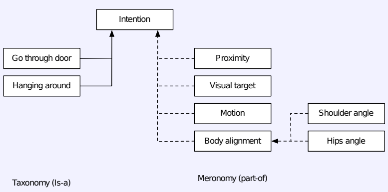


Figure 1: *Features for modeling intention (from Kofod-Petersen et al. [2009])*

This kind of features must be quantified, in order to be reasoned over:

Table 1: *Features, ranges and units*

Main feature	Feature	Range/Unit
Motion	Speed	$[0, 10]\ m/s$
	Acceleration	$[-5, 5]\ m/s^2$
	Heading	$[0, 360\rangle\ ^\circ$
Proximity	Distance	$[0, 10]\ m$
Visual target	Head angle	$[0, 360\rangle\ ^\circ$
Body alignment	Shoulder angle	$[0, 360\rangle\ ^\circ$
	Hip angle	$[0, 360\rangle\ ^\circ$

2. Kinect, OpenNI and NITE

KINECT is a sensor created for tracking humans, and can supply both RGB and Depth image streams.

Figure 2: *Microsoft Kinect Sensor for Xbox 360, providing RGB and Depth data*

OpenNI is a framework that helps in the development of applications utilizing natural interfaces. NITE is middleware supplying human tracking capabilities.

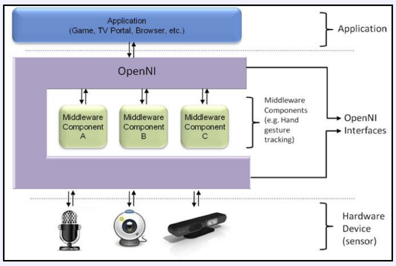


Figure 3: *Layered view of the OpenNI concept (from* `http://www.openni.org`*)*

3. Capturing an intention

CAPTURING the intention is a matter of tracking the right key points:

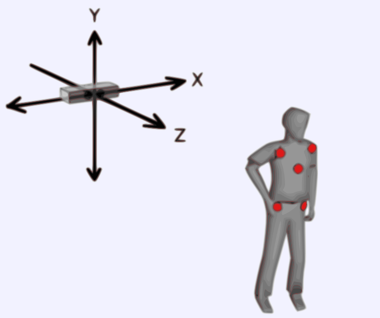


Figure 4: *Five keypoints marked on a user in a coordinate system*

The keypoints must be handled in order to extract the wanted features:

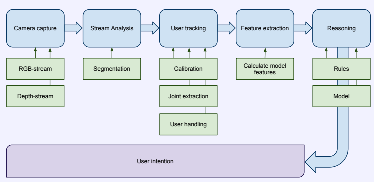


Figure 5: *Pipeline showing the steps in the process*

The reasoning process is the last step in the process, inferring the user's intention.

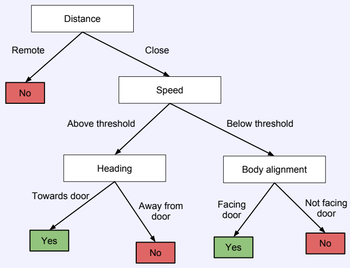


Figure 6: *Decision tree, illustrating reasoning process*

Our solution consists of different components, both on the hardware side and the software side:

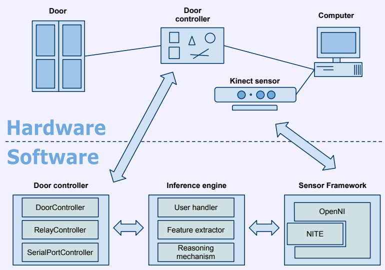


Figure 7: *Component diagram showing solution implementation*

4. Prospects

OUR application is a small step towards what could be much larger. Knowing the users intention, without explicitly asking for it, is very helpful. Transferred to other tasks, this can open a wide range of possibilities. Systems can improve their functionality and efficiency greatly by reading the user's intentions, and acting upon them. Operating on a symbolic level also allows the systems to give reasonable explanations for their choices.

References

Kofod-Petersen, A., Wegener, R., and Cassens, J. (2009). Closed doors – modelling intention in behavioural interfaces. In Kofod-Petersen, A., Langseth, H., and Gundersen, O. E., editors, *Proceedings of the first Norwegian Artificial Intelligence Symposium*, pages 93–102, Trondheim, Norway. Tapir Akademiske Forlag.

SCAI 2011, 11th Scandinavian Conference on Artificial Intelligence Norwegian University of Science and Technology, Trondheim, Norway May 24th - 26th, 2011