

How hand gestures are recognized using a dataglove

Mario Ganzeboom
Human Media Interaction MSc
University of Twente
The Netherlands
m.s.ganzeboom@student.utwente.nl

ABSTRACT

This paper presents research on the topic of glove based gesture recognition. Human Computer Interaction keeps moving toward interfaces which are more natural and intuitive to use, in comparison to traditional keyboard and mouse. In this paper how gestures can be recognized when using the dataglove as means of input is researched. The field of gesture recognition has many recognition techniques to offer. Based on personal experience with Hidden Markov Models (HMMs) in speech and language processing, it is eventually decided to use this technique to develop a recognizer. Gestures for basic interface tasks like clicking, dragging, zooming and rotating are defined. After having found a suitable HMM library, its algorithms are used to train and tune a HMM on a set of observation sequences. Eventually gestures are recognized with two HMMs parallelly connected. The possibilities of this gesture recognition technique are promising. However, future research in the form of an user evaluation is needed to further investigate this promising technique.

Keywords

gesture, glove, interface, haptic, hand, recognition, markov, machine, learning

1. INTRODUCTION

The popularity of gesture interaction with machines with the broader public has increased with films like *Minority Report* released in 2002 [5] and the gaming industry with Nintendo's release of the Wii in 2006. *Minority Report* illustrates the concept of controlling an interface by making gestures with gloves. The movie shows basic interface tasks like dragging, scrolling and zooming. Nintendo on the other hand implemented gesturing by means of accelerometers and infrared cameras embedded in a remote control called WiiRemote. Those remote controls are used in sports games like golf, boxing and racing to be able to use the golf swing, punches and steering respectively. These examples show that Human

Machine Interaction keeps moving toward interfaces which are more natural and intuitive to use. Interfaces that are closer to the human, requiring less adaptation from the human's side, then since the beginning of computers and computing devices, interaction has overall taken place by using a keyboard or mouse. Users had to adapt to these input devices to control the computers supporting them in their daily tasks. Although keyboard and mouse have in general sufficed for the necessary interaction, one wonders how other styles of interaction would perform in doing tasks. With this in mind the research presented in this paper is in the direction of glove-based hand gesturing.

2. PROPOSED RESEARCH

Interaction by means of gesturing could potentially be more intuitive over the use of keyboard and mouse. Because it lies within the nature of humans to make gestures while communicating their needs to do a task. However, to enable interaction by means of gesturing, gestures need to be recognized. The problem of gesture recognition can be solved by using technology. Part of the available technology is a dataglove which provides data on the angle of the bones in one's fingers and wrist. How can this be used in recognizing gestures? In other words the central question is:

- How can gestures be recognized using a dataglove as means of input?

To aid the research for an answer to this question, having a potential application for the glove gesture interface in mind helps. Intuitively, a potential type of application which could be ideal for this interface is a 3D viewer. Objects and whole models can be viewed in 3D. Inspired by the academic environment in which this research is conducted, the concept of a 3D map viewing application came to mind. It displays a 3D map of the university grounds including its campus. The idea is to supplement the current 2D map from the university website. For example when placing this application near the entrance, people who are new to the university can navigate the 3D map to see where they need to go or request additional information about a particular building. Typical actions for navigating such an interface include: pointing, clicking, typing, selecting, dragging, zooming and rotating. These are traditionally done by keyboard and / or mouse. So how can these actions be executed with gloves? In other words, what kind of gestures need to be defined to realize the actions mentioned above and how can the gestures made by

the user be recognized by the application? The remainder of this paper briefly touches upon the former while focusing on the latter.

3. CONSTRAINTS AND LIMITATIONS

The research in this assignment is applicable to constraints and limitations. This section describes those and their consequences.

Due to time constraints on this research subject, the research that was done is limited. For this reason interface actions which were estimated to take up a lot of time because of technological limitations of the glove were not researched. These actions are pointing and typing.

Pointing was not researched because the glove has no internal sensor to track its position, therefore there is no simple way to control a pointer by only using the glove. It is possible to augment the glove with a tracking sensor, but realizing that would be a separate research assignment.

The typing interface action was not researched because it involves having to solve the problem of enabling the user to input at least 36 characters (10 numerical + 26 alphanumeric) by the means of gestures. This would even be of larger proportions than this research assignment.

As mentioned in the previous section typical actions for navigating a 3D map application are clicking and selecting amongst others. When thinking about these two, there is actually no difference between them considering the context they are in. Therefore both actions can have the same gesture. Where one reads clicking in the remainder of this paper one could also read selecting.

This narrows the list of actions down to the following:

- Clicking
- Dragging
- Zooming
- Rotating

The next section continues with a brief overview of previous research on the various directions of gesture recognition.

4. RELATED WORK

In the past glove-based gesturing has been used in sign language training and translation [19, 13]. Another application is in the area of robotics in which a robot arm is controlled by a human wearing a glove [11]. The robot arm imitates the movements made by the human which are registered by the glove. When looking at gesture interaction in a broader sense, possible applications are smart homes or smart surroundings for that matter. For example turn the lights on or off by clapping your hands. Other gesture applications involve touch screens and even multitouch screens, which have the benefit of handling more than one finger. The Apple iPhone for example or several models of HTC's mobile phones have touch screens and make use of gesturing in basic interface tasks like clicking, scrolling, dragging and so on. Recognizing gestures through camera's is also researched. For example, this idea can be applicable on a table in a meeting room, where people who attend the meeting can communicate by drawing on this table with their hands. Another application in the direction of home entertainment is Sony's EyeToy <http://www.eyetoy.com>. EyeToy is a camera at the size of a webcam and is connected to the Playstation 2. Through the camera people can play games as if they were

the main character. The camera enables recognition of body motion of which appropriate game actions are derived.

As the previous paragraph has shown, gesturing has many applications. Applications in various directions of research. When looking closer into the field of gesture recognition, the following directions are found: recognizing facial expressions or head movement, recognizing motion with accelerometers, recognizing motion from video and recognizing hand gestures from glove data or video images.

Research concentrating on recognizing facial expressions or head movement. Both Pantic et al. [14] and Fasel et al. [7] give a good survey of the prominent facial analysis methods and systems.

Taking gesture recognition in another direction is recognizing motion with accelerometers. Perng et al. [15] developed a glove having 2-axis accelerometers on the finger tips and back of the hand enabling the recognition of pseudo static gestures. They also managed to enable the glove to be used as a mouse pointing device which shows an interesting possibility for this paper's research in the future.

With the introduction of the webcam, researchers started to use the video streams as input for hand gesture recognition, recognizing that this could make it available to everyone because of its low-cost setup. Yamato et al. [21] developed a human action recognition method in which a sequence of video images is used as input. With human action recognition is meant the recognition of humans and their movement in the images. They used sequential images of sports scenes and achieved recognition rates from over 90%.

Chen et al. [3] and Fang et al. [6] both developed a method for recognizing hand gestures from video. They achieved satisfactory performance of up to 90% recognition rate. Fang et al. applied their method to navigation of image browsing, which is of interest because it is similar to the application in this paper's research.

Heading more in the direction of this research, scientists research and have researched ways to recognize hand gestures. From the early 1980s researchers used a glove that was directly connected to a computer. These gloves were outfitted with several types of sensors to measure the bending of the fingers and eventually recognize a form of a hand. The use of datagloves has been researched for various applications. Zimmerman et al. [22] have researched applications in which a user manipulates computer-generated objects as if they were real, bringing real-world concepts into a virtual world. Concepts in the virtual world are made equal to the real world. This creates new possibilities in the area of manipulating digital objects. For example, instead of modelling a 3D object with keyboard and mouse, one could actually 'mold' it with ones hands. The techniques they described are in the direction of template matching, in which joint bending angles are recorded. Gestures are then recognized by either matching the bending angles precisely or determining threshold values. Deyou [20] is a more recent example of this application, in which a driving simulator for a military vehicle is in development. The simulator is used to train soldiers virtually in driving and operating the vehicle. Static hand gesture recognition by means of a dataglove is used in this application to perform driving tasks, like preparing the vehicle for movement, starting the engine, handling the instruments, and so on. They used a Back Propagating Neural Network for recognition. Fifteen gestures are defined in total, from which 98% are success-

fully recognized. However, 92% are successfully recognized when gestures are used outside of the training and validation sets. Lee et al. [11] describe a system in which gestures that are made using a glove can be learned interactively and online. This means that a user can teach the system new gestures by demonstrating the gestures himself. They have implemented their recognizer using Hidden Markov Models (HMMs), a technique very often used in the field of speech recognition. The final goal of their project is to move beyond simple keyboard/mouse/teach-dependant methods, by creating frameworks that enable productive real-time interaction between robots and humans. Takahashi et al. [19] and Jong-Sung et al. [10] have experimented with recognizing gestures from the Japanese and Korean sign language alphabet. Both used a dataglove to recognize the various forms of the hand. Takahashi et al. used Principal Component Analysis (PCA) as recognition technique, Jong-Sung et al. used a form of Neural Networks (NNs). Out of the 46 Japanese Kana manual alphabet gestures Takahashi et al. selected, about 30 gestures can effectively be recognized using their setup. The setup used to recognize sign language gestures is similar to that in this paper, which is a single dataglove from which joint angle and hand orientation information are used to recognize the gestures for the navigational actions. Does this mean however that with a similar setup the limit of gestures being able to recognize effectively with one dataglove is about 30? Jong-Sung et al. conducted a pilot study and took it a little further by using two datagloves. They selected 25 important basic gestures from the Korean Sign Language in which either one or both hands were required. After having conducted experiments with 25 different sign languages, they reached a success rate of up to 85% of the given words with their classification method. This may be an acceptable success rate for the context of sign languages, but in the context of navigation in 3D maps it could be very annoying if only one out of six or seven gestures is recognized. This could lead to rejection of the technology by its users. The gesture recognition in this paper probably performs better because of less complexity (only one dataglove) and fewer gestures (only four to five). Having described various applications for glove-based gesturing, the goal of this research is to evaluate a hand gesture recognition technology for manipulating digital objects. Previous research was done on manipulating virtual representations of real-world objects, This paper researches gesture recognition technology for manipulation in the direction of trivial interface tasks. Tasks like clicking, dragging, rotating and zooming in the context of a 3D map. Besides the various hardware (instrumented gloves, camera) that is used for input, there is variety in pattern recognition techniques as well. Several of those were already mentioned in the forelast paragraph. Among them are Principal Component Analysis (PCA) [1], Hidden Markov Models (HMMs) [1, 18, 16, 17, 9], Kalman filtering [1], Multilayer perceptron (MLP) [1] and various types of Artificial Neural Networks (ANN) [1, 18]. For a complete overview of techniques, the reader refers to Mitra and Acharya, [12]. As the above listing shows, many techniques have been developed and used in various fields of pattern recognition. The most prominently applied hand gesture recognition techniques using input from instrumented gloves are HMM and ANN. The advantage of these techniques above the others is that they can model spatio-temporal information. In other words infor-

mation which is time dependent. This property is necessary because the continuously changing positions of the hand are analyzed in real-time. Researchers have also referred to this as dynamic gesture recognition as apposed to static gesture recognition [12]. HMMs have a useful property called the Markov property. A time dependent process demonstrates the Markov property if the conditional probability density of the current event, given all present and past events, depends only on the j^{th} most recent event. If the current event depends solely on the most recent past event, then the process is termed a first order Markov process. This is useful when considering the position and movement of the hands through time. It takes past position and movement of the hands into account, without making it computationally intractable. Additionally, HMMs use several well established algorithms for learning and state recognition like K-Means, Baum-Welch and Viterbi. Although forms of ANNs seem to be less prominently used in hand gesture recognition, they are suitable. The design of an ANN is based on a neuron, which is a brain cell having the principal function of collecting, processing and disseminating electrical signals [18]. It is presumed that the brain's information-processing capability comes from a network of these neurons, hence the name artificial neural network. ANNs are composed of nodes connected by directed links. These nodes are activated or 'fired' when a linear combination of its inputs exceed some threshold. Because of their design, the networks have the ability to perform distributed computation. Other advantageous properties are that they can tolerate noisy inputs, which is very useful when considering the use of glove sensor input, and they have the ability to learn like HMMs. Considering the advantageous and disadvantageous of HMMs and ANNs, HMMs have the advantage that the initial properties can be learned directly from input sequences via the K-Means algorithm. Such an algorithm was not found for ANNs. Taking this into account including the past experience with HMMs, the decision was made to use HMMs as hand gesture recognition technique in this paper.

The remainder of this paper discusses the application of HMMs in this context. The next section continues with the definitions of the hand gestures.

5. GESTURE DEFINITIONS

Since there is no real standard for defining gestures it was argued that current literature does not provide a format for defining gestures. In the literature used for this paper standards are not mentioned [22, 8, 20, 11]. Perhaps there are no universal gestures symbolizing certain actions in user tasks, in other categories there are. Sign language for example already has an established 'gesture vocabulary' although they differ per language [10, 19, 13]. In what other way could gestures be found and defined? The context in which these gestures are used naturally plays a roll. As described in section two, the context is a 3D map application in which the research is limited to finding gestures for clicking, dragging, zooming and rotating. The following paragraphs explain the gestures defined for these actions by a description and a summary of its flow.

Clicking

Description

In my opinion, the most intuitive gesture for clicking is when

the hand is in pointing stance (like in pointing at something with your index finger) then pushing wrist downwards, as if touching/tapping a surface. Considering that a sensor to control the mouse pointer is placed on the back of your hand this movement interferes with pointing. This makes it difficult to extrapolate the position of the click. Therefore similar to clicking the left mouse button, the gesture for recognizing clicking will only be pushing your index finger down and back up again. When pushed down, the index finger should be bend (not stretched). The other fingers should be bend like forming a fist. See figure 1 for an example of the click gesture.



Figure 1: The click gesture.

Flow

Begin - Only index finger down - Only index finger up immediately - Begin

Rotating in 3D

Description

To define a gesture for rotating in 3D, again how that action is done with the mouse is taken as an example. Rotating in 3D with a mouse is in most programs like Maya and 3D Studio Max done by grabbing the camera's current view point and rotate around it by moving the mouse up, down, left and right. For the hand gesture the actions will be similar: grabbing current view point, moving up, down, left or right. The most natural hand gesture for grabbing something is actually grasping it with your hand and fingers. Therefore the choice was made to form a fist with your hand to grab the current view point. For actually rotating the view, the choice was made to move the wrist up, down, left and right. The following paragraphs explain the gestures in detail.

X-axis: Start by making a fist with the hand. Does not matter where you are pointing at. It should grab the camera's current location. Keeping your hand in a fist, move your wrist up and down to rotate about the X-axis right and left respectively. The movement range of the wrist up and down acts as a speed scale. Moving more down should make rotating left go faster and the opposite holds for making rotating right go faster.

Y-axis: Same as X-axis only with moving the wrist left and right. This respectively rotates left and right about the Y-axis.

Z-axis: In respect to the 3D map application, rotating about

the Z-axis is not useful, then one would get a tilted view of the map. This does not add anything useful to viewing the map. One could argue that a 3D map application is therefore unnecessary, but that is not so. The third dimension also serves the purpose to give the user spatial information increasing the user's orientation of the area.

See figure 2 for a visual explanation.



Figure 2: The rotating gesture. Hand forms a fist and wrist joint rotates left, right and up and down.

Flow

Begin - Form fist - Move wrist left, right, up and / or down - Form open hand

Zooming in 3D

Description

The most intuitive gesture for zooming in 3D is the one that resembles the actual effect of zooming in or out, that is 'drawing' the viewer deeper into the view or the opposite, 'pulling' the viewer out of the view. An analogy which is similar is 'pushing' the view from the viewer or the opposite, 'pulling' it closer. The similar hand gesture would then be grabbing the current view point and moving the whole arm to your body to zoom in and away from your body to zoom out. However you need to know the current position of the hand to measure the displacement. This unfortunately cannot be registered with the glove used in this research without adding extra sensors. A different gesture was defined therefore.

To keep the analogy with pushing and pulling, the following was thought of. To zoom in and out of objects in 3D, the gesture starts with an open hand (not a fist) and pushing all fingers together. The hand and wrist are levelled with the lower arm. By bending your wrist downwards you can control the speed of zooming out and by bending it upwards from the levelled position the speed of zooming in is controlled. This is done whilst keeping the fingers stretched, so only the wrist joint is used.

Flow

Begin - Push fingers together and level wrist with lower arm - Move wrist up or down - Level wrist and / or release fingers

Dragging in 3D

Description

The gesture for dragging in 3D is similar to that of rotating



Figure 3: The zoom gesture. Stretching all fingers and pushing them together. Rotating wrist up and down zooms in and out respectively.

in 3D. The difference is that with dragging the lower arm is moved left, right, up and down to perform the actual dragging.

X-axis: Start by making a fist with the hand. This should grab the current location which is pointed at. Move your lower arm to the left or right, to drag positively or negatively along the x-axis respectively, keeping your hand in a fist. The upper arm is about 45 degrees with your shoulder. Maximum drag positions are when the lower arm is stretched or fully bend. After dragging is done open the hand to release the grabbing point.

Y-axis: Same as with x-axis only moving complete arm up and down along the y-axis.

Z-axis: To drag along the Z-axis one first needs to rotate along the Y-axis to change the view from for example YX y|_x to ZY z_|y. After that it is the same as dragging about the X and Y-axes as described above.

Figure 4 displays the dragging gesture visually (same as rotating).

Flow

Begin - Form fist - Move arm left, right, up and / or down - Form open hand



Figure 4: The drag gesture. Same as rotate gesture. Form a fist, but move whole arm up, down, left and right.

6. HMM THEORY

To recognize the gestures the choice was made to develop a recognizer based on HMMs, as described in the section ‘Related work’. In this section the theory behind HMMs is described, including how it is applied to the context of this paper.

First of all, to start developing a HMM one needs to know what HMMs are and how they are modelled. A brief description of the HMMs definition and properties is given next. A HMM is a statistical model. It recognizes patterns which are learned through observing sequences of a particular kind of data. It models a system which is considered to be in one of a set of a particular number of states. The system modelled is assumed to be a Markov process. A process is a Markov process when it qualifies the criteria that the conditional probability distribution of future states of the process, given the present state and a constant number of past states, depend only upon the present state and given states in the past, but not any other states in the past. Additionally, in a HMM the states of the system are hidden, not observable, hence the name Hidden Markov Model. With not observable is meant that we do not know at any point in time in which state the system is.

A HMM model consists of various parameters. Let a HMM model be denoted as λ , the following are the parameters which need to be known [1]:

1. N : Number of states in the model

$$S = \{S_1, S_2, S_3 \dots, S_N\}$$

2. M : Number of distinct observation symbols in the alphabet

$$V = \{v_1, v_2, \dots, v_M\}$$

3. State transition probabilities:

$$A = [a_{ij}] \text{ where } a_{ij} \text{ is defined as } P(q_{t+1} = S_j | q_t = S_i). q_{t+1} \text{ is the state being recognized and } q_t \text{ is the current state.}$$

4. Observation probabilities:

$$B = [b_j(m)] \text{ where } b_j(m) \text{ is defined as } P(O_t = v_m | q_t = S_j). q_t \text{ is the current state.}$$

5. Initial state probabilities:

$$\Pi = [\pi_i] \text{ where } \pi_i \text{ is defined as } P(q_1 = S_i)$$

Summarized, $\lambda = (A, B, \Pi)$, because N and M can be derived from A and B . For more on the theory behind HMMs the reader is referred to [1, 18, 17, 16].

Given the fact that the above parameters need to be known to define a HMM model, can we derive any of these given the application context described in this paper? The answer is yes, two parameters can be derived. Considering the gestures described in the section ‘Gesture definitions’, $N = 7$, because every gesture represents a state which needs to be recognized. The hardware which is used is the CyberTouchTM from CyberGlove Systems [4]. It is a glove outfitted with up

to 22 high-accuracy joint-angle measurement sensors. They measure the flexion of the fingers, wrist flexion and abduction and thumb crossover. From the version used in this research 18 sensors can be used, but one of them is broken. Fortunately that sensor is not crucial in our gesture recognition. Each of these sensors provide integers in the range of 0 - 255 as output. These are used as input for the recognizer. The stream of sensor data from the glove is continuous. The recognizer is given this input in a vector of 18 dimensions. Each distinct vector is an observation symbol, v , in the alphabet of observation symbols, V . That is the second parameter of the HMM model which is known to us. The other parameters mentioned above cannot be derived given the application context. They need to be estimated or 'learnt', to stay in terms of machine learning. Learning these parameters of the HMM model is done with a training set of observation sequences. Is such a set available? Yes, it is, because such a set is easily generated from the input of the glove. Given such a set of sequences of observations, three problems are of interest [1, 18]:

- P 1. Given a HMM model λ , the evaluation of the probability of any given observation sequence, $O = \{O_1 O_2 \dots O_T\}$, denoted as $P(O|\lambda)$.
- P 2. Given a HMM model λ and an observation sequence O , find out the state sequence $Q = \{q_1 q_2 \dots q_T\}$ which has the highest probability of generating O . Also denoted as, $P(Q|O, \lambda)$.
- P 3. Given a training set of observation sequences, $\chi = \{O^k\}_k$, learn the model that maximizes the probability of generating χ . Also denoted as: $P(\chi|\lambda)$.

The third problem described above equals the context of this paper, because a HMM model is needed and only a training set of observation sequences is available. Therefore problem three matches the current situation in that a training set of observation sequences is given and the HMM that maximizes the probability of generating that training set is to be found. However, in the end gestures need to be recognized. These are recognized from the output data generated by the glove. So, that means that observation sequences (the glove data) goes into the HMM model and needs to be translated to a state sequence having the highest probability of generating that observation sequence. This matches the second problem mentioned above.

Concluding this section, to develop a HMM model, first a solution needs to be found to learn the model that maximizes the probability of generating the given training set of observation sequences (P 3) and second, find out the state sequence which has the highest probability of generating a given observation sequence from the learned model (P 2).

7. IMPLEMENTING, TRAINING AND TUNING

As the title of this section already suggests, it is about implementing, training and tuning a HMM. Recall from the previous section that first a solution needs to be found to the problem of training a HMM from only a set of observation sequences.

At the Human Media Interaction (HMI) group ([http://](http://hmi.ewi.utwente.nl)

[hmi.ewi.utwente.nl](http://www.utwente.nl)) of the University of Twente (<http://www.utwente.nl>), a *CyberTouchTM* system is available. A proper training set of observation sequences was not however. Such a set existing of at least 50 observation sequences per gesture is required to train a HMM in recognizing each gesture pattern. So, that was the first task to do. The glove is connected via a COM port to a PC and in the past a driver for reading the sensors and controlling the actuators was already written by the HMI group. The driver is written in Java. After setting it up and writing a simple recording tool, the recording of a training set began. However the question remained how to record a set having good training properties. Not knowing the answer to that question directly, ten observation sequences were recorded for purposes of setting up and testing the training and recognition process. This of course is nothing near a proper training set. Having these sequences, the next step was to actually train a HMM from training data. Since the introduction of HMMs at around 1972 [9] in the field of speech recognition, several algorithms have been developed for this: K-Means [1], Expectation Maximization [1, 18] and a special case the Baum Welch algorithm [1, 2].

To save time understanding these algorithms in full detail and implementing them, existing HMM libraries / packages were searched. Eventually the JaHMM package (<http://code.google.com/p/jahmm/>) was chosen, because of the language it is written in and the fact that it was more developed than the other packages.

After having studied JaHMM's API documentation and several examples, the process of training a HMM with this package became clear. With the few observation sequences recorded, an initial HMM with 2 states (fist and neutral) was trained using the K-Means algorithm. To improve the recognition the initial HMM was trained with the Baum-Welch algorithm on the same training set, running it with two iterations. However, divisions by zero occurred during this process. It appeared that overlearning or overfitting was the cause. Because the training set was only for try out purposes and setting up the recognition process, it learned these so well that it could hardly generalize other less 'fitting' sequences. It could only recognize the start state of the hand when it is in resting position. As soon as a fist was formed the recognizer kept throwing erroneous data, as a side effect of the overfitting. Therefore the proper training set was generated containing 50 observation sequences for the neutral and fist form of the hand. This solved the overlearning problem and overall recognition improved to 93% (14 out of 15 were correctly recognized). Additionally the output of two sensors near the wrist and on the top of the hand could be ignored, because those are not useful in the recognition process. The same procedure was followed for the other gestures: zooming and clicking. The new observation sequences for these gestures added to the training set, made it grow and the HMM to learn needed to recognize two extra states. Letting the HMM recognize the zoom gesture was not that difficult and only needed a little tuning. The click gesture was however more difficult, because it is similar to the fist form. At first the output of all sensors (15) were used to learn the HMM the click gesture. Combined with the observation sequences of the other gestures this interfered with the recognition of the zoom and fist gesture. On the other hand training a HMM only with the click data led to 0% recognition, because only lowering and raising the

index finger was too little for the recognizer to see it as a separate gesture. Maybe reducing the dimensions used in the input would lead to better results. For the click gesture the sensors on the thumb and index finger are the most important. This gives a four dimensional input. Training a HMM with only those four sensors and also using the output of those four during recognition drastically improved the recognition of the click gesture to 100% (15 out of 15 were correctly recognized).

Because of this fact the idea came up to train a second HMM for recognizing the click gesture alongside the HMM for the zoom, drag and rotate gesture. But how to connect them so that the glove input runs through both and the right gesture is recognized? That is pretty straight forward in this case. Both HMMs receive the input, calculate for themselves what gesture they think they see with the corresponding probability. The next step is to evaluate these probabilities and take the one which is highest as the gesture recognized by the whole HMM. However in some important cases like having the hand close to the neutral form and during the forming of a fist, the form of the hand was also recognized as a click gesture. In other words in those cases the actual gestures to be recognized were too much like the click gesture. In trying to prevent this, the middle finger was added to the training of the 'click HMM', increasing its input dimensionality to 6. The middle finger was chosen because in the click gesture it is always bend completely and stays in that position throughout the gesture. This made the difference in the previously mentioned important cases, improving the recognition of the overall HMM to recognizing 14 out of 15 gestures when returning the hand in resting position between gestures. 15 out of 15 gestures were recognized when transitioning smoothly between gestures. See table 1 and table 2 for the confusion matrices with the results of the final tests. Given these results, the definition of the click gesture needed to be modified. The sentence stating that the gesture does not depend on the state of the other fingers needed to be rephrased to that it depends on the middle finger being bend.

With this last improvement in recognition this section is

Confusion matrix	Rest position	Fist	Zoom	Click
Rest position	5	0	0	0
Fist	0	3	0	0
Zoom	0	0	4	0
Click	0	1	0	2

Table 1: Confusion matrix for 15 gestures when hand returns in resting position between gestures. On the left the predicted values and from the top the actual values.

Confusion matrix	Rest position	Fist	Zoom	Click
Rest position	5	0	0	0
Fist	0	3	0	0
Zoom	0	0	4	0
Click	0	0	0	3

Table 2: Confusion matrix for 15 gestures when transitioning smoothly between gestures. On the left the predicted values and from the top the actual values.

nearing its end. All the defined gestures are recognized when normally executed. To visually make the recognition process more attractive, a simple GUI next to the command line output was programmed in Java. It displays the gesture recognized in that instant and also some system output statements. The application has been 'baptized' as 'The Hand Gesture Recognizer' or HGR for short. Figure 5 shows a screenshot of the application.



Figure 5: Screenshot of the Hand Gesture Recognizer application showing it recognizes the clicking gesture and also showing system output statements.

8. CONCLUSION

To summarize, this research was inspired by films like Minority Report and novel technologies of that of the WiiRemote. The goal was to research how hand gestures can be recognized when using a dataglove for input. Literature was searched to explore potential possibilities for recognition techniques making use of glove data as input. Several techniques were found. Eventually it was decided to use HMMs for recognition. Through a practical process of training and tuning all defined gestures were eventually recognized. The research question as stated in section two was how gestures could be recognized using a dataglove as means of input. The answer is that this is possible by training a HMM with the KMeans algorithm from a training set in which around 50 observation sequences are available for every gesture. Afterwards the HMM is tuned by training it with the Baum-Welch algorithm on the same training set. The resulting recognition has been found to be pretty accurate and fast at 93% (with hand in resting position between gestures) and 100% (smooth transition from one gesture to the next), proving to be a potentially good technological option for using gesturing in basic interface tasks. Now that the technological side of the process has shown great potential, it is also interesting to evaluate the recognition with users in a real-world setting.

9. FUTURE WORK

The research described in this paper ends here, but there is enough research for the future. Throughout the research a training set was used, which has

been created by a single person's movement of the hand. The glove output is therefore limited to a single hand and since hands differ in size and freedom of motion, the question arises how the recognizer performs if used by others. In other words, there is a chance that the current training set only performs accurate and fast for that single user, limiting its application.

Eventually it is interesting to find out how users are able to perform with glove based gesturing. Do they find it more natural and more efficient? A complete user evaluation should give insight into these questions. To do that, the application context described in the 'Proposed research' section needs to be implemented. After that the glove based gesturing can be evaluated in a real-world context.

As described in the section 'Related work', many techniques for gesture recognition exist. How does one come to know the advantages and disadvantages? By implementing a gesture recognition framework in the future the various techniques and algorithms can be compared to each other. This creates a useful overview of which techniques and algorithms are best used in a particular context.

Concluding this section, the previous paragraphs have described various directions of research which can be followed in the future. The field of gesture recognition remains an interesting and challenging field for research.

10. REFERENCES

- [1] ALPAYDIN, E. *Introduction to Machine Learning*. The MIT Press, Massachusetts, 2004.
- [2] BAUM, L. E., PETRIE, T., SOULES, G., AND WEISS, N. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics* 41, 1 (1970), 164–171.
- [3] CHEN, Q., GEORGANAS, N., AND PETRIU, E. Real-time vision-based hand gesture recognition using haar-like features. In *Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE* (May 2007), pp. 1–6.
- [4] CYBERGLOVE SYSTEMS. CyberGlove Systems website. Retrieved April 8, 2009 from <http://www.cyberglovesystems.com/>, 2009.
- [5] DICK, P. K., FRANK, S., AND J., C. Minority Report. Retrieved April 29, 2009 from <http://www.imdb.com/title/tt0181689/>, 2002.
- [6] FANG, Y., WANG, K., CHENG, J., AND LU, H. A real-time hand gesture recognition method. In *Multimedia and Expo, 2007 IEEE International Conference on* (July 2007), pp. 995–998.
- [7] FASEL, B., AND LUETTIN, J. Automatic facial expression analysis: a survey. *Pattern Recognition* 36, 1 (2003), 259 – 275.
- [8] ISHIKAWA, M., AND MATSUMURA, H. Recognition of a hand-gesture based on self-organization using a dataglove. In *Neural Information Processing, 1999. Proceedings. ICONIP '99. 6th International Conference on* (1999), vol. 2, pp. 739–745 vol.2.
- [9] JURAFSKY, D., AND MARTIN, J. H. *Speech and Language Processing*. Prentice-Hall, New Jersey, NJ, USA, 2000.
- [10] KIM, J.-S., JANG, W., AND BIEN, Z. A dynamic gesture recognition system for the korean sign language (ksl). *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 26, 2 (Apr 1996), 354–359.
- [11] LEE, C., AND XU, Y. Online, interactive learning of gestures for human/robot interfaces. In *In IEEE International Conference on Robotics and Automation* (1996), pp. 2982–2987.
- [12] MITRA, S., ACHARYA, T., MEMBER, S., AND MEMBER, S. Gesture recognition: A survey. *IEEE Transactions on Systems, Man and Cybernetics - Part C* 37 (2007), 311–324.
- [13] MURAKAMI, K., AND TAGUCHI, H. Gesture recognition using recurrent neural networks. In *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 1991), ACM, pp. 237–242.
- [14] PANTIC, M., AND ROTHKRANTZ, L. Automatic analysis of facial expressions: the state of the art. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22, 12 (Dec 2000), 1424–1445.
- [15] PERNG, J., FISHER, B., HOLLAR, S., AND PISTER, K. Acceleration sensing glove (asg). In *Wearable Computers, 1999. Digest of Papers. The Third International Symposium on* (1999), pp. 178–180.
- [16] RABINER, L. R. A tutorial on hidden markov models and selected applications in speech recognition. *Readings in speech recognition* (1990), 267–296.
- [17] RABINER, L. R., AND JUANG, B. H. An introduction to hidden Markov models. *IEEE ASSP Magazine* (January 1986), 4–15.
- [18] RUSSELL, S., AND NORVIG, P. *Artificial Intelligence: A Modern Approach*. Pearson Education Inc., Upper Saddle River, NJ, USA, 2003.
- [19] TAKAHASHI, T., AND KISHINO, F. Hand gesture coding based on experiments using a hand gesture interface device. *SIGCHI Bull.* 23, 2 (1991), 67–74.
- [20] XU, D. A neural network approach for hand gesture recognition in virtual reality driving training system of spg. *Pattern Recognition, International Conference on* 3 (2006), 519–522.
- [21] YAMATO, J., OHYA, J., AND ISHII, K. Recognizing human action in time-sequential images using hidden markov model. *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on* (Jun 1992), 379–385.
- [22] ZIMMERMAN, T. G., LANIER, J., BLANCHARD, C., BRYSON, S., AND HARVILL, Y. A hand gesture interface device. *SIGCHI Bull.* 17, SI (1987), 189–192.