Examining the feasibility of face gesture detection using a wheelchair mounted camera

Gregory Fine

John K. Tsotsos

Technical Report CSE-2009-04

May 27 2009

Department of Computer Science and Engineering

4700 Keele Street Toronto, Ontario M3J 1P3 Canada

Abstract

While the interest in human-friendly autonomous wheelchairs for disabled people grows, the issue of obtaining feedback from the user in convenient and unobtrusive ways becomes critical. The major research focus in this area is concentrated on issues of autonomous capabilities, such as obstacle avoidance and autonomous navigation. Relatively little attention is being paid to the issue of interaction of the autonomous wheelchair with a user.

Naturally, mobility impaired persons are limited in their activities and may have difficulties performing their everyday activities, so providing efficient and convenient interaction with a wheelchair user becomes an important component of autonomous wheelchairs. Some of the sample tasks that such a system can perform are detecting when the user points to something, looks at a display, is happy or distressed, etc. Having this type of information about the actions of the user, provides valuable feedback to the autonomous wheelchair and greatly facilitates its decision making process. Existing autonomous wheelchair systems have user interaction systems

that concentrate on controlling the wheelchair, either by hand or head gestures of the user. These systems have several drawbacks, including inability to provide any feedback from the user to the autonomous wheelchair, limitations in functionality and unsuitability for some users. The purpose of this research is to examine the feasibility of a system that obtains feedback from the user by monitoring his/her gestures using a camera mounted on a wheelchair. The prototype of such a system, which recognizes static facial gestures, has been implemented and tested, achieving 90% recognition rate with 6% false positive and 4% false negative rates.

Table of Contents

1 Introduction

1.1 Motivation

In 2002, 2.7 million people that were aged fifteen and older used a wheelchair in the USA [1]. This number is greater than the number of people who are unable to see or hear [1]. The majority of these wheelchair-bound people have serious difficulties in performing routine tasks and are dependent on their caregivers. The problem of providing disabled people with greater independence has attracted the attention of researchers in the area of assistive technology. Wheelchair-bound individuals are the most vulnerable group of disabled people because they are most limited in their mobility. Therefore, the problem of creating an intelligent wheelchair, which allows performing of some routine everyday tasks, has especially attracted attention of researchers. Controlling such a wheelchair and ensuring its safe operation may be challenging for disabled people. Hence, obtaining feedback from the user and taking independent decisions based on this feedback is one of the important components of an intelligent system. Ideally, an autonomous wheelchair should be able to perform

some of the routine tasks autonomously instead of relying on the direct input of a user. Such a wheelchair requires some form of feedback to obtain information about the intentions of the user. For example, an automatic wheelchair may determine if the user is pointing at anything, looking at the display, showing happiness or looking elsewhere. It is desirable to obtain the feedback in an unconstrained and non intrusive way and the use of a video camera is one of the most popular methods to achieve this goal. This work explores the feasibility of a system capable of obtaining visual feedback from the user for usage by an automatic wheelchair. In particular, this work considers visual feedback, namely facial gestures. Current research appears to lack work on obtaining visual feedback from a wheelchair user.

The requirement to obtain feedback in a non-intrusive way that does not allow placement of a video camera directly in front of the user. This fact makes obtaining visual feedback from the person sitting in the wheelchair challenging, due to an inability to obtain pure frontal images of the person. Facial features in non frontal images are usually distorted and occluded; therefore, their detection and analysis are difficult.

1.2 An Approach to Wheelchair User Monitoring

Growing demand to provide an increasing number of disabled people with a wheelchair that can give them a greater degree of independence has led to great interest in

research for the area of assistive technology. Significant progress has been achieved in the area of intelligent wheelchairs. Modern intelligent wheelchairs are able to autonomously navigate indoors and outdoors, and avoid collisions during movement without intervention of the user. However, in order to serve the user in the best possible way, even the most sophisticated wheelchair should be able to accept some sort of input from the user. Such input provides the wheelchair with directions for the next task and feedback from the user about the task being executed. The form of the input has the greatest impact on the convenience of using the wheelchair. Ideally, the user should not be involved in the low level direct control of the wheelchair. For example, if the user wishes to move from the bedroom to the bathroom, the wheelchair should receive instruction to move to the bathroom and navigate there autonomously without any assistance from the user. During the execution of the task, the wheelchair will monitor the user in order to detect if the user is satisfied with the decisions taken by the wheelchair, if he/she requires some type of assistance or he/she wishes to give new instructions. The task of moinitoring the user may be difficult for an autonomous wheelchair due to the fact that the wheelchair is generally unaware whether the user produced a facial expression in response to an action of the wheelchair or as a result of some unrelated event. To the best of the author's knowledge, there is no such system available for intelligent wheelchairs. This work concentrates on one component of such a system; namely,

on the monitoring system.

This section proceeds with a general overview of the proposed system, then continues with the general design, description of training of the system and integration into existing intelligent wheelchair systems, and finally discusses possible future directions and implementation.

1.2.1 System Overview

While intelligent wheelchairs are becoming more and more sophisticated, the task of controlling them becomes increasingly important in order to utilize their full potential. The direct control of the wheelchair that is customary for non-intelligent wheelchairs cannot utilize fully the capabilities of an autonomous wheelchair. Moreover, the task of directly controlling the wheelchair may be too complex for some patients. To overcome this drawback this work proposes to add a monitoring system to a controlling system of an autonomous wheelchair. The purpose of such a system is to provide the wheelchair with timely and accurate feedback of the user on the actions performed by the wheelchair or about the intentions of the user. The wheelchair will use this information for planning of its future actions or correcting the actions that are currently performed. The response of the wheelchair to feedback of the user depends on the context in which this feedback was obtained. In other words, the wheelchair may react differently or even ignore feedback of the

user in different situations. Due to the fact that it is difficult to infer intentions of the user from his/her facial expressions, the monitoring system will complement regular controlling system of a wheelchair instead of replacing it entirely. Such an approach facilitates the task of controlling an autonomous wheelchair and makes a wheelchair more friendly to the user. The most appropriate way to obtain feedback of the user is to monitor the user constantly using some sort of input device and classify the observations into categories that can be understood by the automatic wheelchair. To be truly user friendly, the monitoring system should neither distract the user from his/her activities nor limit the user in any way. Wearable devices, such as gloves, cameras or electrodes, usually distract the user and therefore, are unacceptable for the purposes of monitoring. Microphones and similar voice input devices are not suitable for passive monitoring, because their usage requires explicit involvement of the user. In other words, the user has to talk, so that the wheelchair may respond appropriately. Vision based approaches are the most suitable for the purposes of monitoring the user. Video cameras do not distract the user, and if they are installed properly, they do not limit the field of view.

The vision based approach is versatile and capable of capturing a wide range of forms of user feedback. For example, they may capture facial, head and various hand gestures as well as face orientation and gaze direction of the user. As a result, the monitoring system may determine, for example, where the user is looking, is the

user is pointing at anything, is the user happy or distressed. Moreover, the vision based system is the only system that is capable of passive and active monitoring of the user. In other words, a vision based system is the only system that will obtain the feedback of the user by detecting intentional actions or by inferring the meaning of unintentional actions. The wheelchair has a variety of ways to use this information. For example, if the user looks at a certain direction, which may differ significantly from the direction of movement, the wheelchair may slow down or even stop, to let the user look at the area of interest. If the user is pointing at something, the wheelchair may identify the object of interest and move in that direction or bring the object over if the wheelchair is equipped with a robot manipulator. If there is a notification that should be brought to attention of the user, the wheelchair may use only visual notification if the user is looking at the screen or a combination of visual and auditory notifications if the user is looking away from the screen. The fact that the user is happy may serve as confirmation of the wheelchair actions, while distress may indicate incorrect action or a need for help. As a general problem, inferring intent from action is very difficult.

1.2.2 General Design

The monitoring system performs constant monitoring of the user, but it is not controlled by the user and therefore, does not require any user interface. From

the viewpoint of the automatic wheelchair, the monitoring system is a software component that runs in the background and notifies the wheelchair system about detected user feedback events. To make the monitoring system more flexible, it should have the capability to be configured to recognize events. For example, one user may express distress using some sort of face gesture while another may do the same by using a head or hand gesture. The monitoring system should be able to detect the distress of both kinds correctly depending on a user observed. Moreover, due to the high variability of the gestures performed by different people, and because of natural variability of disorders, the monitoring system requires training for each specific user. The training should be performed by trained personnel at the home of the person for which the wheelchair is designed. Such training may be required for a navigation system of the intelligent wheelchairs, so the requirement to train the monitoring system is not exaggerated. The training includes collection of the training images of the user, manual processing of the collected images by personnel and training the monitoring system. During training, the monitoring system learns head, face and hand gestures as they are produced by the specific user and their meanings for the wheelchair. In addition, various images that do not have any special meaning for the system are collected and used to train the system to reject spurious images. Such an approach produces a monitoring system with maximal accuracy and convenience for the specific user. It may take a long time to train

the monitoring system to recognize emotions of the user, such as distress, because a sufficient number of images of genuine facial expressions of the user should be collected. As a result, the full training of the monitoring system may consist of two stages: in the first stage, the system is trained to recognize hand gestures and the face of the user, and in the next stage, the system is trained to recognize the emotions of the user.

To provide the wheelchair system with timely feedback, the system should have good performance that allows real-time processing of input images. Such performance is sufficient to recognize both static and dynamic gestures performed by the user.

To avoid obstructing the field of view of the user, the camera should be mounted outside the user's field of view. However, the camera should be also capable of taking images of the face and hands of the user. Moreover, it is desirable to keep the external dimensions of the wheelchair as small as possible, because a compact wheelchair has a clear advantage when navigating indoors or in crowded areas. To satisfy these requirements one of the places to mount the camera is on an extension of the side handrail of the wheelchair. This does not enlarge the overall external dimensions of the wheelchair, limit the field of view of the user and allows tracking of the face and hands of the user. However, this requires that the monitoring system deals with non-frontal images of the user, taken from underneath of the face of the

user. Such images are prone to distortions and therefore, the processing of such images is challenging. To the best of the author's knowledge, there is no research that deals with facial images taken from underneath of the user face at such large angles as required in this work. In addition, the location of the head and hands is not fixed, so the monitoring system should deal with distortions due to changes of the distance to the camera and viewing angle.

The block diagram of the proposed monitoring system is presented in Fig. 1.1. The block diagram illustrates the general structure of the monitoring system and its integration into the controlling system of an intelligent wheelchair.

1.2.3 Implementation and future work

The implementation of the monitoring system, which can satisfy all requirements, is a very complex task. This work takes the first step towards the creation of such a system by implementing a system that is capable of recognizing ten static facial gestures. The proposed monitoring system uses an existing automatic wheelchair system and satisfies the basic requirements as specified in this chapter. The system uses as input, images that are obtained by using a video camera. Moreover, the camera is mounted in the way that does not obstruct the field of view of the user. In the current implementation, the proposed system does not have real time performance and is not integrated into the wheelchair system.

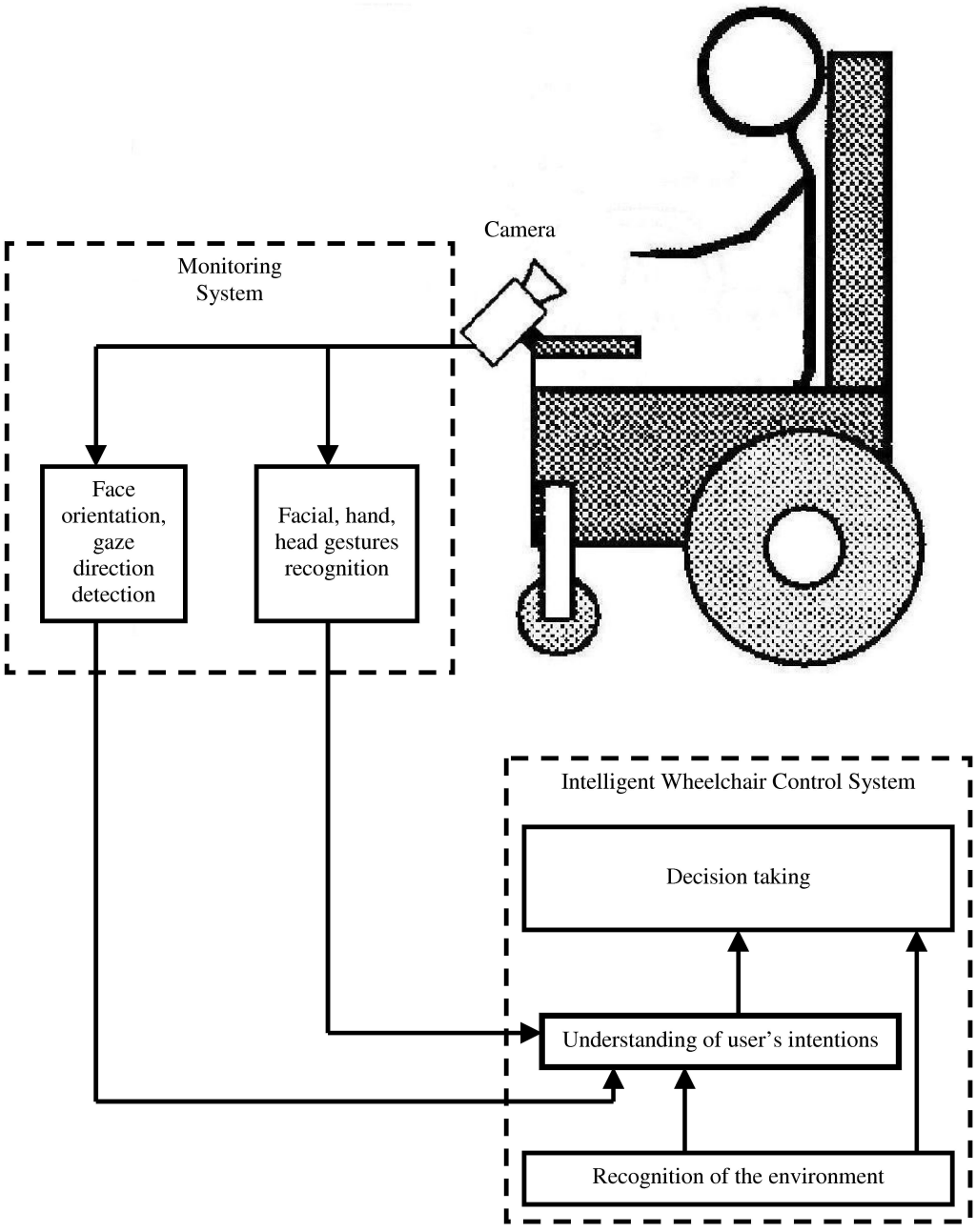


Figure 1.1: The block diagram of monitoring system

The results allow the usage of this research as a base for future development. One of the directions is implementing the capability of recognizing emotions. The demonstrated recognition performance of the proposed algorithm may be sufficient enough to recognize emotions from static images with proper training. The algorithm may be also extended to recognize dynamic facial gestures of the user. Another possible extension of the algorithm is detection of the direction of the user's gaze. The proposed algorithm produces contours of eyes as an interim result, which greatly simplifies the task of detecting the direction of the user's gaze, for example, using approaches proposed by Wang and Sung [99] or Wang et al. [100]. In addition, the proposed approach may be adapted to detect static hand gestures of the user.

1.3 Contributions

The research described in this report, works towards the development of an automatic wheelchair user monitoring system. This work presents a system that is capable of monitoring static facial gestures of a user of an automatic wheelchair in a non-intrusive way. The system obtains the images using a standard camera, which is installed in the area above the knee of the user as illustrated in Figure 3.1. Such a design does not obstruct the field of view of the user and obtains input in a non-intrusive and unconstrained way.

Previous research in the area of interfaces of automatic wheelchairs with humans concentrates on the issue of controlling the wheelchair by a user. The majority of proposed approaches are suitable for controlling the wheelchair only. One of the major contributions of this work is that it examines the feasibility of creating a monitoring system for users of autonomous wheelchairs and proposes a general purpose static facial gesture recognition algorithm that can be adopted for a variety of applications that require feedback from the user. In addition, unlike other approaches, the proposed approach relies solely on facial gestures, which is a significant advantage for users with severe mobility limitations. Moreover, the majority of similar approaches requires the camera to be placed directly in front of the user, obstructing his/her field of view. The proposed approach is capable of handling non frontal facial images and therefore, does not obstruct the field of view.

The proposed approach has been implemented in software and evaluated on a set of 9140 images from ten volunteers, producing ten facial gestures. Overall, the implementation achieves a recognition rate of 90%.

1.4 Report outline

This report consists of five chapters. The first chapter provides motivation for the research and describes the entire monitoring system in general. The second chapter discusses previous related work. The third chapter provides technical and algorith-

mic details of the proposed approach. The fourth chapter details the experimental evaluation of a software implementation of the proposed approach. Finally, chapter five provides a summary and conclusion of this work as well as suggestions for future work.

2 Background and Related Research

Automatic wheelchairs attract much attention from researchers (see e.g., [38, 88, 103] for general reviews). However, most research in the area of automatic wheelchairs focus on automatic route planning, navigation and obstacle avoidance. Relatively, little attention has been paid to the issue of the interface with the user. To the best of the author's knowledge, all existing research in the area of user interface is concentrated on the issue of controlling the automatic wheelchair by the user [88]. The methods that control the automatic wheelchair include mechanical devices, such as joysticks, touch pads, etc. (e.g. [17]); voice recognition systems (e.g.[52]); electrooculographic (e.g.[6]), electromyographic (e.g.[40]) and electroencephalographic (e.g.[93]) devices; and machine vision systems (e.g.[70]). The systems involving machine vision have clear advantages over other approaches because mechanical devices and voice recognition systems are unsuitable for user monitoring, and electro-oculographic devices are too intrusive. Therefore, only machine vision approaches are considered in this work. This chapter proceeds with a brief

review of the research on interaction with automatic wheelchairs and then continues with research on gesture recognition.

2.1 Review of research on interaction with automatic wheelchairs

Due to the fact that this work considers only machine vision approaches, this section reviews a research on controlling autonomous wheelchairs using various computer vision approaches.

Bley et al. [17] proposed a system that allows the control of an automatic wheelchair, using a joystick, touchscreen and facial gestures. The facial gestures are used to control the motion of the wheelchair. The authors proposed the use of Active Appearance Models (AAMs) [90] to detect and interpret facial gestures, using the concept of Action Units (AUs) introduced by Ekman and Friesen [29]. To improve the performance of the algorithm, an AAM is trained, using an artificial 3D model of a human head, on which a frontal image of the human face is projected. The model of the head can be manipulated in order to model variations of a human face due to head rotations or illumination changes. Such an approach allows one to build an AAM which is insensitive to different lighting conditions and head rotations. The authors do not specify the number of facial gestures recognizable by the proposed system or the performance of the proposed approach.

Bien et al. [14] proposed a wheelchair system that has several control mechanisms. One of the mechanisms, called Eye-mouse, is of particular interest in the context of this work. This system tracks the direction of the user gaze and uses it to manipulate objects on the computer screen. The flexibility of such an approach is limited only by the application. To eliminate an effect of the head rotation, the user is required to wear a special device that tracks the pupil of the eye. Wearing such a device seriously obstructs the field of view of the user. No quantitative results were provided by the authors.

Moon et al. [74] proposed a system that combines various techniques to communicate with the user. The techniques include an electromyographic device, a voice recognition system and face directional gesture recognition system. Such an approach provides great flexibility and applicability to a wide range of disorders. The face directional gesture recognition system uses a combination of face direction with head gestures to encode a face directional gesture. The system classifies all possible face directions into three categories: forward, left and right. Head gestures, defined in the context of this work, are shaking and nodding. Nodding is used to set the direction of movement of the wheelchair. For example, if the user turns head left and simultaneously nodes his/her head, the wheelchair will move to the left. Shaking is used to stop the wheelchair. The system uses frontal images of a user sitting in the wheelchair as input. To obtain the direction of the face, the

system detects a face by using skin color segmentation and then infers the direction of the face from the difference between the center of the facial region and center of gravity of facial features. In total, the system is able to recognize four commands with a recognition rate of 93%.

Kang et al. [49] proposed an algorithm that utilizes static palm gestures to control the wheelchair. To improve the performance of the algorithm, gestures were performed against a semi opaque surface and the camera was mounted behind the surface. The images that are obtained in this way are actually shadows of the palm that can be easily segmented and processed. Such an approach makes the algorithm insensitive to skin color. In addition, the proposed approach does not require extensive hand movements and therefore it is usable by people with limited hand mobility. In order to obtain images of palm gestures, the source of light and semi opaque screen should be mounted on the wheelchair. To classify the gestures, the algorithm uses eight geometric properties of a palm shape. The authors reported that their approach is capable of recognizing fourteen gestures reliably. However, no quantitative results were provided.

Nakanishi et al. [75], and Adachi et al. [2], similarly to [74], proposed the use of the face direction of a wheelchair user, to control the wheelchair. The system uses face direction to set the direction of the movement of the wheelchair. However, a straightforward implementation of such an approach produces poor results

because unintentional head movements may lead to false recognition. To deal with this problem, the authors ignored quick movements and took into account the environment around the wheelchair [75]. Such an approach allows improvement of the performance of the algorithm by ignoring likely unintentional head movements. The algorithms operated on images obtained by a camera tilted by 15 degrees, which is much less than the angles in this work. To ignore quick head movements, both algorithms performed smoothing on a sequence of angles obtained from a sequence of input images. While this technique effectively filters out fast and small head movements, it does not allow fast and temporally accurate control of the wheelchair. Unfortunately, only subjective data about the performance of these approaches have been provided.

Yoda et al. [105], and Yoda et al. [104] proposed to control a wheelchair using head gestures which were detected by a stereo camera. The camera obtained non-frontal facial images, although in a less steep angle than in this work. The authors defined a head gesture as turning the head right or left and monitored the face orientation angle in order to detect these gestures. The approach is similar to the ones proposed in [75] and [2], in the sense that face orientation angles obtained from a sequence of input images are used to classify head movements into gestures. The authors proposed to estimate face orientation angle from disparity information, obtained from an analysis of stereo images. Unlike the approaches proposed in [75]

and [2], this approach uses thresholds instead of smoothing to classify a sequence of face orientation angles into head gestures. The proposed approach achieves a 94% success recognition rate. Recognized facial gestures are used to select wheelchair commands from a predefined set, which may be of any size.

Satoh and Sakaue [84] suggested the use of an omnidirectional stereo camera to detect gestures of a wheelchair bound person. The omnidirectional stereo camera, mounted above the wheelchair, had the shape of a regular dodecahedron with trinocular camera units on each face, with 36 cameras in total. Obtained images were used to generate a panoramic image of the environment. To generate such an image, the algorithm also required data of attitude sensors in order to deal with inclination of the camera during movement and inaccurate installation of the camera on the wheelchair. This approach has a clear advantage over other approaches due to the absence of blind spots in the field of view of the camera, except occlusions caused by the wheelchair itself or the user. The authors reported that the proposed system is capable of recognizing two types of gestures: emergency stop and pointing to the direction of movement. No quantitative data about the performance of the algorithm was provided.

Kuno et al. [57] proposed a system to control an automatic wheelchair, using hand gestures. The most distinctive features of this approach is the ability to distinguish between intentional and unintentional hand gestures and "guessing" of

the meaning of unrecognized intentional hand gestures. The system assumed that a person who makes an intentional gesture will continue to do so until the system recognizes it. Once the system established the meaning of the gesture, the person continued to produce the same gesture. Hence, to distinguish between intentional and unintentional gestures, repetitive patterns in hand movement are detected. Once a repetitive hand movement is detected, it is considered an intentional gesture. In the next stage, the system tried to find the meaning of the detected gesture by trying all possible actions until the user confirmed the correct action by repeating the gesture. The authors reported that the proposed wheelchair supports four commands, but they do not provide any data about the performance of the system.

Matsumoto et al. [70] suggested the use of a combination of head gestures and gaze direction to control an automatic wheelchair. The system obtained images of the head of a wheelchair user by a stereo camera. The camera of the wheelchair was tilted upward 15 degrees, so that the images obtained by the camera were almost frontal. The usage of a stereo camera permits a fast and accurate estimate of the head posture as well as gaze direction. The authors used the head direction to set the direction of wheelchair movement. To control the speed of the wheelchair, the authors used a combination of face orientation and gaze direction. If face orientation coincided with a gaze direction, the wheelchair moved faster. To start or stop the wheelchair, the authors used head shaking and nodding. These gestures

were defined as consecutive movements of the head of some amplitude in opposite directions. The authors do not provide data on the performance of the proposed approach.

Hu et al. [42] presented a system that controlled the wheelchair using static head gestures. The input images were obtained by a camera, installed in front of the wheelchair bound person. The system supported five commands, which were given by turning the head left, right, up, down or keeping the head straight. The head posture was detected using a combination of face detection [98] and template matching. Unfortunately, the quantitative data on the performance of this approach has not been provided by the authors.

Bergasa et al. [12], and Bergasa et al. [11] proposed to control a wheelchair by using a combination of head and facial gestures. Input images were obtained using a video camera placed in front of a wheelchair user. In the context of this approach, head gestures were defined as turning the head left, right, up or down; facial gestures were defined as opening or closing of the eyes or mouth. The face of a user was detected using color based segmentation and analyzed using some heuristics, to obtain the contours of the eyes and mouth. To detect the event of turning the head, the face was tracked using a Kalman filter [46]. Each gesture defined a command for the wheelchair, so the wheelchair supported a set of eight commands. The authors do not provide quantitative performance of the proposed

approach.

While the approaches presented in this section mainly deal with controlling the wheelchair, some of the approaches may be useful for the monitoring system. The approach proposed in [17] is extremely versatile and can be adopted to recognize facial gestures of a user. The approaches presented in [2, 75] and especially in [70] may be used to detect the area of interest of the user. The approach presented in [57] may be useful to distinguish between intentional and unintentional gestures. However, more research is required to determine whether this approach is applicable to head or facial gestures.

2.2 Review of research on gesture recognition

Gesture recognition is an active research area in computer vision (see, e.g., [73, 101] for general reviews). Gestures can be defined as a body motion intended to communicate with the environment. Of particular interest are gestures performed by hands, fingers, heads, faces or the whole body. Given the breadth of this area, the review of all relevant work is beyond the scope of this work. In particular, this work does not consider body language or gestures formed by body motions of the whole body. This subsection proceeds with brief reviews of the research in facial, head and hand gesture recognition.

2.2.1 Review of research on facial gesture recognition

Detection and recognition of facial gestures attract a great deal of research attention (see e.g., [30, 81] for general reviews). This section proceeds with a brief review of the research in the area of recognition of facial gestures and expressions.

Algorri and Escobar [4] proposed the use of PCA analysis [41] for recognizing facial features. The main purpose of the proposed algorithm is to compress face images for videoconferencing. Hence, no classification of the detected facial features into categories is performed. However, this approach may be adapted for classification of facial gestures. The authors trained separate detectors for the left and right eyes as well for the mouth. Aside from training, the algorithm required an initial setup in order to track the eyes and mouth of a person. The authors admitted that their method is not able to handle all possible variations due to movements of the user. Unfortunately, a quantitative evaluation of the performance of the proposed algorithm is not provided.

Fazekas and Santa [31] proposed the use of an SVM classifier [24] to recognize facial gestures from static frontal monochrome images of a face. The authors proposed two approaches to solve the problem of facial gesture recognition. In the first approach, the separate classifier was trained using the whole face for each gesture. The second approach used two layers of classifiers. The first layer contained two

classifiers for each gesture that were trained to recognize the eyes and mouth for each facial gesture. The second layer contained a separate classifier for each gesture. Each classifier at the second layer received as an input, the output of all classifiers of the first layer. Both approaches were trained to recognize five facial gestures. The authors do not provide the overall performance of their approaches. For the first approach, the recognition rate ranges from 67.78% to 85.56% for images of people unknown to the classifier and from 77.65% to 88.24% for people whose images were used to train the classifier. For the second approach, the recognition rate ranges from 77.78% to 95.56% for images of people unknown to the classifier and from 84.71% to 94.12% for people whose images were used to train the classifier

Liao and Cohen [62] proposed an algorithm to recognize facial gestures in the presence of head motion. The authors estimated 3D head poses by modeling the head as a 3D cylinder whose position and rotation were estimated from an image sequence. According to this approach, facial gestures may be modeled as a combination of local deformations of some regions on the face. The authors defined nine such regions that represent facial gestures, so that facial gestures were represented as a combination of local deformations in each region. The motion inside each region was estimated using an affine motion model. The parameters of the affine motion of all regions are considered as a random multidimensional variable. The likelihood of this variable is estimated using a graph model of a face and classified

into facial gestures using an SVM classifier [24]. According to the authors, the algorithm is capable of recognizing six facial gestures, which represent emotions, in the presence of head motion. The performance of the algorithm ranges from 70.46% to 83.24% for different gestures.

The algorithm presented by La Cascia et al. [61] is capable of detecting facial gestures in low resolution video sequences. To estimate the posture of a head, the head was modeled as a texture mapped cylinder. The approach recognized two facial gestures: mouth opening and eyebrow raising. To detect these facial gestures, the authors proposed two approaches to extend the tracker. The first approach used head models with raised eyebrows or an opened mouth for tracking so that facial gestures were detected simultaneously with the head tracking. However, this approach performs poorly when head motion occurs simultaneously with facial gestures. The second approach performed the head posture estimation and then facial gesture detection. In the experiments conducted by the authors, the recognition rate for mouth opening ranged from 27% to 72% for the first approach and from 50% to 88% for the second approach. The recognition rate for the eyebrows raising ranged from 25% to 70% for the first approach and from 42% to 86% for the second approach.

The facial expression recognition algorithm, proposed by Bartlett et al. [7], is capable of recognizing seven facial expressions which correspond to emotions.

The algorithm used a face detector, similar to the face detector proposed by Viola and Jones [98], to detect frontal faces in input images. To detect characteristic facial features, the results were processed using Gabor filters [37]. In the next stage, detected features were used as an input for the facial expressions classifier. The facial expression classifier consisted of seven SVM classifiers [24]. Each of the SVM classifiers was trained to recognize a specific facial expression. To improve the classification performance, the authors suggested the use of the Adaboost algorithm, first proposed by Freund and Schapire [35], to select the most informative Gabor features for classification by the SVM classifiers. In the experiments conducted by the authors, the algorithm achieved a 93.3% recognition rate.

Unlike the approaches proposed in [7, 98], the algorithm of facial expression recognition by Chen et al. [19] is robust to pose and size variations as well as partial occlusions. The authors proposed the use of a multi-class hybrid-boost learning classifier to detect faces and classify facial expressions. This classifier is similar to other boost algorithms, e.g. Adaboost [35], in a sense that its output is combined from the outputs of several weak classifiers. Weak classifiers are usually simple, but do not have a good recognition rate. The boost classifier selects a number of weak classifiers from a large pool during the training and combines them into a single strong classifier, which has better performance than any weak classifier. The authors proposed the use of Gabor [37] and Haar-like features [97] for weak

classifiers. They reported that the proposed approach is capable of detecting six facial expressions which correspond to emotions, with a 93.1% average recognition rate.

Zhan et al. [108] presented an algorithm to recognize facial expressions using 2D Gabor wavelet transformation, derived from [37], and elastic template matching that was first described by Balkenius [5]. The Gabor wavelet transformation is used to extract distinct facial expression features, which are used by the elastic template matching algorithm to detect facial expressions. To train the algorithm, the authors calculated the Gabor wavelet transformation on latticed training images and selected the points, located around the eyes, mouth and nose, with the largest amplitude values to train the elastic template classifier. At the recognition stage, trained elastic templates were matched to values of the Gabor wavelet transformation of an input image, which were calculated in a way similar to the training stage. The authors reported that the proposed algorithm recognizes six facial gestures which correspond to emotions, with an average recognition rate of 90.4%.

Kim et al. [55] proposed an extension of the face detector, which was suggested by Viola and Jones [98], for detection of facial expressions. In the first stage, the algorithm detected a face in the input image using the face detection algorithm proposed in [98]. However, any face detection algorithm may be used. To adapt

the face detection algorithm [98] to the task of recognizing facial expressions, the authors proposed an extension of a set of rectangular features used in [98] to a set of all possible rectangular features that is sized 3×3. The algorithm selected the five most efficient features for each trained facial expression. The selected features along with features used in [98] were used to train the classifier. The authors reported that the proposed algorithm is capable of recognizing seven facial expressions which represent emotions, with an average recognition rate of 92.2%.

The approach proposed by Shan and Gritti [86] uses local binary patterns, which were first introduced by Ojala et al. [78], and an SVM classifier [24] to recognize facial expressions. Local binary patterns were used to describe the local structure of an image due to their computational simplicity and tolerance to monotonic illumination changes. Generally, the local binary pattern is a number, which encodes a texture around some point in an image. The structure of some image region may be represented as a histogram, containing values of local binary patterns at each point that belongs to this region. Similar to [7], the authors proposed the use of an Adaboost to select the most discriminative regions of an image by examining the histograms of local binary patterns. An SVM classifier was used to classify facial expressions using histograms of local binary patterns selected by the Adaboost algorithm. In the experiments conducted by the authors, the proposed algorithm recognized seven facial gestures which corresponded to emotions, with a

93.1% recognition rate.

Sohail and Bhattacharya [89] proposed the classification of facial expressions by classifying distances between the salient features of a face using the k Nearest Neighbor classifier [32]. The algorithm relies on the concept of action units, introduced by Ekman and Friesen [29], to define facial features for classification. The algorithm detected eleven feature points, which were located in the areas that provide information about action units involved in an expression, mainly around the eyes and mouth. The eyelids, eyebrows, nostrils and mouth were detected using various heuristics and were used to locate the feature points. The distances between the detected feature points were used to classify the facial expressions. The authors reported that the algorithm is capable of recognizing six facial gestures which represent emotions, with an average recognition rate of 90.76%.

Due to the fact that all of the described approaches deal with images taken by a camera that is located in front of a face, none of the approaches can be readily used in autonomous wheelchairs. The approaches based on the algorithm of Viola and Jones [98], e.g. [19, 55], have good recognition performance of over 90%, but generally require extensive training and the large number of training samples. Therefore, such approaches may be inapplicable for usage in a wheelchair monitoring system. The approach that combines an Adaboost with SVM classifiers, e.g. [7, 86], usually have better performance than other approaches and do not require extensive

training. The approach proposed by Zhan et al. [108] has a good performance and does not require large amounts of training samples. Both approaches are capable of recognizing facial expressions. However, more research is required to use these approaches for the monitoring of a wheelchair user.

2.2.2 Review of research on head gesture recognition

The approach proposed by Kapoor and Picard [51] recognizes head shakes and nods by tracking the pupils. To detect the pupils, the authors proposed the use of a camera with infrared LEDs. Switched on LEDs create a red-eye effect, hence, the pupils may be easily and accurately detected by analyzing the difference between a pair of images taken with LEDs switched on and off. The locations of pupils were used for classification of head movements. The authors used two separate Hidden Markov Model (HMM) [9] based classifiers to classify head movements for nods and shakes. The HMM classifier is capable of classifying a sequence of observations into categories. This fact makes the HMM classifier especially suitable for gesture classification. In the experiments, the algorithm achieved an average recognition rate of 78.46%.

Kawato and Ohya [54] proposed an approach to detect head nodding and shaking by tracking a point between the eyes, which is a midpoint on the line connecting the centers of the two eyes. The area between the eyes has dark parts on the left

and right (eyes and eyebrows) and bright parts on the top and bottom (forehead and nose). To detect this area, the authors suggested the extraction of the face area using skin color thresholding and applying a circle frequency filter to the detected region. The authors defined the circle frequency filter of a pixel p as a discreet Fourier transform of pixels laying along a circle centered at the pixel p. Such an approach allows the detection of circular structures in the image, such as the area between the eyes. After initial detection, the area was tracked using template matching. The position of the point between the eyes was used to classify head movements into nodding or shaking. The classification was performed using empirically selected rules. The authors reported that the proposed algorithm is capable of detecting shaking and nodding in real time with an average recognition rate of 86.22%.

The algorithm proposed by Ng and De Silva [77] is capable of recognizing three head gestures in low resolution videos of poor quality with a complex background. The algorithm worked with frontal color images of a face and recognized nodding, shaking and tilting gestures. Face and hair regions were detected using simple color thresholding. In the next stage, feature vectors for the classification were formed from the first four invariant moments, which were proposed by Hu [43], for each detected region. The authors proposed the use of a separate HMM classifier [9] for each head gesture to classify feature vectors to head gestures. They reported that

the proposed algorithm achieves an average recognition rate of 87%.

The approach proposed by Tang and Nakatsu [94] is based on a feature point tracker, proposed by [67, 87, 96], and a neural network classifier. In the first stage, the algorithm extracted the head region using color thresholding. In the next stage, the head was tracked using a feature point tracker. Finally, a feature vector was formed from tracked point coordinates and fed to the neural network classifier, proposed by Lin and Kung [64]. To improve the performance of the classification, a separate neural network was trained for each gesture. The output of such a classifier is an output of a neural network that produced the best result. In the experiments conducted by the authors, the algorithm successfully recognized ten head gestures with a recognition rate that ranges from 84.3% to 95.7% for different gestures.

Bayesian networks, first proposed by Pearl [83], were used by Lu et al. [66] to recognize head nods and shake gestures. In the first stage, the head was detected in the input image by using the face detection algorithm, proposed in [65]. The detected face was aligned using Active Shape Models, first proposed by Cootes et al. [22]. The color model of the detected face was learned and used to track the head and infer its posture. Inferred poses were used to infer a head gesture using an HMM [9] classifier. These estimations of head posture and gesture were used by the Bayesian network to estimate the probability of a head gesture. The authors used a separate Bayesian network for each gesture. They reported that the

proposed algorithm is capable of recognizing head shakes and nods with an average recognition rate of 92.1%.

Similar to the approach proposed in [51], Kang and Rhee [50] suggested an algorithm to recognize head gestures by tracking eyes. The authors did not use any special hardware, e.g. as in [51] and used the face detection algorithm, proposed by Nam and Rhee [76]. In the next stage, the eyes were detected in the detected face region using heuristic rules. The eyes were detected in each input frame. In cases where eye detection failed, the authors proposed the interpolation of the location of the eyes using information from previous input frames. The detected eye coordinates were used to classify the gestures by using the classifier based on HMMs [9]. The algorithm distinguished between head shakes, referred as negative gestures, nods, referred as positive gestures, and all other head movements, which are referred to as neutral gestures. To improve the performance of the classification, separate classifiers were used to classify head nods and shakes. In the experiments conducted by the authors, the algorithm achieved an average recognition rate of 93.3%.

The majority of the approaches presented in this section are capable of recognizing a very limited number of head gestures: head shakes and nods. It is not clear if these approaches may be extended to recognize other gestures. The approach proposed in [94] is capable of recognizing a large number of gestures with a

good recognition rate. These facts make this algorithm especially attractive for use in the monitoring system of an autonomous wheelchair. Other approaches, such as [50, 66], have good recognition and computational rates and may be also used in monitoring systems. However, more research is required to ensure that more gestures may be recognized using these approaches.

2.2.3 Review of the research on hand gesture recognition

Recognition of hand gestures is an active research area in computer vision (see, e.g., [45, 82] for general reviews). This section proceeds with a brief review of research in the area of recognition of hand gestures.

Marcel et al. [69] proposed an algorithm based on Input-Output HMMs, introduced by Bengio and Frasconi [10], to recognize hand gestures. While HMMs [9] calculate the probability that a sequence of observations can be generated by a model, Input-Output HMMs calculate a probability that a sequence of output events can be generated by a sequence of input observations. In the context of the proposed algorithm, a gesture is defined as a movement of a person's hand relative to his/her face. The hand and face were detected using skin color thresholding. The algorithm classified all gestures into two categories: deictic gestures, representing pointing movements, and symbolic gestures, representing intentions to execute a command, e.g. grasp, rotate. In the experiments conducted by the authors, the

algorithm recognized 97.6% of deictic gestures and 98.9% of symbolic gestures.

The algorithm proposed by Okkonen et al. [79] is able to recognize hand gestures in a cluttered environment. The authors defined a hand gesture as a static hand posture formed by the fingers of a hand. To initialize the algorithm, the authors proposed the learning of the background and usage of this information to perform initial segmentation of the hand. This initial segmented region was used to learn the color of the hand. After the initialization, the algorithm used a combination of learned background and hand color to perform segmentation of the hand in input images. In the next stage, the segmented hand contour was represented by Fourier descriptors [107] and classified by an SVM classifier [24]. The authors reported that the proposed algorithm is capable of recognizing five gestures with a recognition rate of 89.2%.

The approach proposed by Binh et al. [15] combined the Kalman filter [46] and Pseudo 2D HMMs, first proposed by Kuo and Agazzi [60], to recognize 36 American Sign Language gestures in real-time. The hand region was extracted using skin color segmentation and tracked using the Kalman filter [46]. The Kalman filter was used to predict the location of the hand in the subsequent input frames, so this information is used to accelerate the detection of the hand region and reject spurious skin color regions. The trajectory of the centroid of the hand region was classified using the Pseudo 2D HMM classifier. The Pseudo 2D HMM classifier

allows incorporation of the hand shape and trajectory information into one HMM [9] to improve recognition. The authors reported that the proposed algorithm recognizes 36 hand gestures with a recognition rate up to 98%.

Yoon et al. [106] proposed an algorithm that recognizes 48 hand gestures using hand location, velocity and angle. Their work considered planar hand gestures, performed in front of a camera, representing alphanumeric characters and graphic elements. The authors detected the hand in every input image using skin color thresholding. The locations of the detected hand from several input frames were collected and the trajectory as well as the velocity of the hand movement were obtained. Unlike similar approaches, e.g. [15, 69], this algorithm does not use the hand trajectory for classification. Instead, the coordinates and velocity at every point of the hand trajectory were clustered into 48 gesture tokens using the k-means algorithm which was first proposed by Steinhaus [91]. Obtained clustered gesture tokens were classified using HMM classifier [9]. In the experiments conducted by the authors, the algorithm recognized 48 hand gestures with a 93.25% recognition rate.

The main purpose of the approach, proposed by Licsar and Sziranyi [63], is to control a virtual reality system, so it uses a projector to display the user interface on a surface while a user performs static hand gestures against this surface, hence, the background is also projected on the hand. Such an approach makes straightforward

hand detection based on skin color thresholding impossible due to the influence of the projected background. To perform hand detections, the authors used the fact that the projected background is known and proposed to detect the hand using background subtraction. The obtained contours of a hand were processed using Fourier descriptors [107] and classified using the k Nearest Neighbors classifier [32]. In the experiments conducted by the authors, the algorithm recognized nine hand gestures with a recognition rate that ranges from 86.2% to 99.8%. To improve the recognition efficiency, the authors retrained the system dynamically. According to this approach, if the gesture is recognized correctly, the algorithm updates the previously learned gesture parameters to ensure adaptation of the system to small gradual changes in gestures performance; if the gesture is not recognized, the user may retrain this gesture. This approach improves the performance and adaptability of the algorithm. The performance of the algorithm after dynamic retraining ranges from 96.1% to 99.2%.

The approach proposed by Chen et al. [18] combines spatial and temporal features to classify dynamic hand gestures in real time. To achieve the computational efficiency and robustness a combination of edge, motion, skin color information was used to detect the hand. As a result of the detection, shape and motion parameters of the hand were estimated. The shape parameters of the detected hand were represented by Fourier descriptors [107]. The calculated Fourier descriptors as well

as motion parameters formed the feature vector for classification. Similar to [106], feature vectors were clustered into 64 gesture tokens, which were used to classify the gestures. The authors used a separate HMM classifier [9] for each gesture and selected as an output, the output of a classifier that achieved the highest classification score. In the experiments, the algorithm recognized 20 hand gestures with a recognition rate of 93.5%.

Huang and Jeng [44] proposed an algorithm that uses hand shape to classify dynamic hand gestures. The authors used hand motion to detect a hand in the input image and ignored the trajectory of hand movements during classification. It was assumed that the hand is the only moving object in a scene and the hand was detected in input frame using edge and motion information. To create a compact representation of the shape of the hand, the shapes were aligned by using Procrustes analysis [39], which is similar to [22], and then a PCA analysis [41] was performed. This approach allows compact representation of possible hand shape variations and achieves invariance to scale, rotation and translation. A single HMM classifier [9] was used to classify 18 hand gestures. The input for the classifier was the distance between the hand shape detected in the image and the previously trained hand model. In the experiments conducted by the authors, the performance of the algorithm ranged from 79% to 96%.

The approach proposed by Malima et al. [68] is extremely simple, computa-

tionally efficient, and capable of recognizing static hand gestures. Gestures were defined by the number of unbent fingers. Such an approach limits the number of gestures that can be recognized, but makes the algorithm extremely simple and computationally efficient. In the first stage, the hand was detected using skin color thresholding. In the next stage, the center of the hand region was estimated. Then, the most extreme point, belonging to the hand region, was determined. It is assumed that this point belongs to the tip of one of the unbent fingers. In the next stage, a circle was placed on the center of the hand region. The radius of this circle was suggested to be 0.7 of the distance from the center of the head to the most extreme hand point. This circle crosses all of the fingers participating in the gesture. The values of the pixels laying on this circle were extracted and considered as the 1D signal. The number of fingers was determined as a number of low-to-high transitions of the signal minus one (for the wrist). Naturally, the number of gestures recognizable by this algorithm was limited to five. The authors reported that the algorithm achieved a 91% recognition rate during the experiments.

Freeman and Roth [34] suggested the use of orientation histograms to classify static hand gestures in real time. The authors did not specify how to locate the hand region in the input image and assumed that the hand is the only object in an input image. The local dominant orientations of the input image was calculated in accordance to Freeman and Adelson [33]. A histogram of these orientations would

also be built. Such an approach is invariant to illumination changes and translations. Each gesture was represented by a histogram of local dominant orientations and classified using a classifier similar to the k-Nearest Neighbors classifier [32]. Orientation histogram of each gesture was compared to those of trained gestures in terms of Euclidean distance. The authors noted that this approach may also be extended to the recognition of dynamic gestures. Unfortunately, no quantitative data were provided on the performance of the algorithm.

The approach suggested by Chen et al. [20] uses Haar-like features with an AdaBoost classification algorithm [97] and grammar based syntactic analysis to recognize dynamic hand gestures. The authors defined a hand gesture as a sequence of static hand postures connected by global and local motions over a period of time, so that gesture classification consists of the problem of recognizing static hand postures and classifying the sequence of these postures to hand gestures. In this work the authors dealt with a first part of the problem, recognition of static hand postures. To improve the performance, separate classifiers were trained for each recognized gesture. As a result, some gestures may be classified by several classifiers. To overcome such ambiguity, the authors proposed the selection of a correct gesture using a grammar based analysis. The proposed algorithm is reported with capability of recognizing four static hand gestures with a recognition rate that is above 90%.

Coogan et al. [21] presented an approach that uses hand shape and motion to recognize dynamic hand gestures. Color, motion and position information were used to detect a hand. Unlike similar approaches, e.g. [44], the proposed approach is robust to occlusions of the face by a hand. The occlusion detection was performed by predicting the positions of the head and hands using a Kalman filter [46]. In addition, the prediction of the hand position by the Kalman filter was used to improve the detection speed of the algorithm by limiting the region, where the search was performed. The shape of a hand for a particular gesture was encoded as a subspace created using PCA analysis [41]. Unlike the approach proposed by Huang and Jeng [44], this work does not use Procrustes analysis [39]. A PCA analysis was performed on a set of rotated, translated and scaled images of a hand gesture. This approach ensures invariance to translation, rotation and scaling of a hand in input images. The classification of a hand shape was performed by projecting input hand shape into each subspace and finding the subspace which is the closest to the input hand shape. The authors reported that the proposed approach is capable of recognizing 28 static shapes with 94.5% accuracy. To classify dynamic gestures, the authors used a separate HMM classifier [9] for each gesture. The shape and location of a hand were used as input for the classifier. To improve the performance of the classifier, the image was divided into nine regions and the location of the hand was encoded as a region, where the hand was located in the input image. The

authors reported that the proposed approach is capable of recognizing 17 dynamic gestures with a recognition rate of 98.6%.

Derpanis et al. [26] suggested the use of linguistic theory to recognize American Sign Language gestures in real-time. Such an approach, originally proposed in [92], allows representing of complex hand gestures in terms of simple components, such as hand shape, location and movement. To describe the movement of the hand the authors proposed the use of an affine model. The parameters of the model were estimated from motion analysis of the input images. The motion analysis can be performed automatically or with manual initialization of a hand region in the first input frame. In the next stage, kinematic features were computed from the motion model. The authors assumed that each gesture has distinctive kinematic features, referred as a gesture signature. To classify the gestures, the signature of the input gesture is compared to a set of signatures of prototypical gestures. The prototypical gesture which has the closest signature in terms of Euclidean distance was selected as an output of the algorithm. The authors reported that the algorithm classified 592 hand gestures and achieved 86% recognition rate for fully automatic processing and 97.13% for manual initialization.

While all of the presented approaches deal with frontal images of a person, it is not clear whether these approaches may be applied to monitor a user of an autonomous wheelchair, because it is impossible to obtain frontal images of

a wheelchair user. However, some of the presented approaches are particularly interesting. The idea of dynamic training, proposed in [63], may significantly simplify the training of a monitoring system and make the whole autonomous wheelchair more user friendly. The approaches that use a combination of hand shape and trajectory to recognize hand gestures, e.g. [15, 18, 21, 106] and especially [26], are generally capable of recognizing a large number of gestures with very good accuracy. These facts make such approaches especially attractive for the monitoring system of an autonomous wheelchair. The idea of using grammar based analysis for classification, proposed in [20], may greatly improve the performance of gesture recognition. However, more research is required to make these approaches applicable for the monitoring of a wheelchair user.

3 A Methodology for Gesture Recognition

3.1 System Overview

This chapter describes the theory and technical details behind the current work. This section briefly describes the requirements for the monitoring system and then proceeds with a detailed description of the algorithm.

The facial gesture recognition system is part of an existing automatic wheelchair and this fact will have some implications on the system. It takes an image of the face as input, using a standard video camera, and produces the classification of the facial gesture as an output. The software for the monitoring system may run on a computer that controls the wheelchair. However, the input for the monitoring system can not be obtained using the existing design of the wheelchair and requires installation of additional hardware. Due to the fact that the system is intended for automatic wheelchair users, the hardware should neither limit the user nor obstruct his or her field of view. Currently, the automatic wheelchair has a touch screen as an operating console for the user, which is mounted on a wheelchair handrail.

This location may also be suitable for mounting a video camera for the monitoring system because in doing so, it will neither limit the user nor obstruct the field of view. This approach has one serious drawback: the camera mounted in such a manner produces non frontal images of the face of the user who is sitting in the wheelchair. Non frontal images are distorted and some parts of the face may even be invisible. These facts make detection of facial gestures extremely difficult. Dealing with non frontal facial images taken from underneath of a person is very uncommon and rarely addressed. The automatic wheelchair with an installed camera for the monitoring system and a sample of the picture that is taken by the camera, are shown in Figure 3.1.

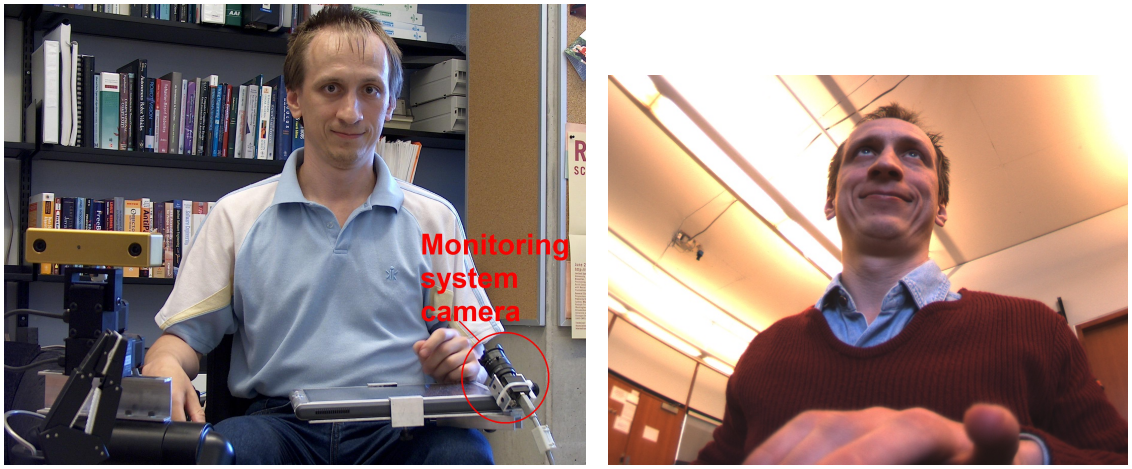


Figure 3.1: (a) The automatic wheelchair[left]. (b) Sample of picture taken by face camera [right].

3.1.1 Facial Gestures

Generally, facial gestures are caused by the action of one or several facial muscles. This fact along with the great natural variability of the human face make the general task of classifying facial gestures difficult. Ekman and Friesen [29] proposed Facial Action Coding System (FACS), a comprehensive system that classifies facial gestures. The approach is based on classifying clearly visible changes on a face and ignoring invisible or subtly visible changes. It classifies a facial gesture using a concept of Action Unit (AU) which represents a visible change in the appearance on some area of the face. Ekman [28] classified over 7000 possible facial gestures. It is beyond the scope of this work to deal with this full spectrum of facial gestures.

In this work, a facial gesture is defined as a consistent and unique facial expression that has some meaning in the context of application. The human face is represented as a set of contours of various distinguishable facial features that can be detected in the image of the face. Naturally, as the face changes its expression, contours of some facial features may change their shapes, some facial features may disappear, and some new facial features may appear on the face. Hence, in the context of the monitoring system, the facial gesture is defined as a set of contours of facial features, which uniquely identify a consistent and unique facial expression that has some meaning for the application. It is desirable to use a constant set of

facial features to identify the facial gesture. Obviously, there are a lot of possibilities in selecting facial features, whose contours define the facial gesture. However, selected facial gestures should be easily and consistently detectable. Taking into consideration the fact that the most prominent and noticeable facial features are the eyes and mouth, the facial gestures produced by the eyes and mouth are most suitable for usage in the system. Therefore, only contours of the eyes and mouth are considered in this research. Facial gestures formed by only the usage of the eyes and mouth, are a small subset of all facial gestures that can be produced by a human. Hence, many gestures cannot be classified using this approach. However, it is assumed that the facial gestures that have some meaning for the monitoring system differ in the contours of the eyes and mouth. Hence, this subset is enough for the purpose of this research, namely a feasibility study. The samples of facial gestures used in this work are shown in Figure 4.1.

3.1.2 System Design

Conceptually, the algorithm behind the facial gesture detection has three stages: (1) detection of the eyes and mouth in the image and obtaining their contours; (2) conversion of contours of facial features to a compact representation that describes the shapes of contours; and (3) classification of contour shapes into categories representing facial gestures. This section proceeds to briefly describe these stages;

the rest of the chapter discusses these stages in more details.

In the first stage, the algorithm of the monitoring system detects the eyes and mouth in the input image and obtains their contours. In this work, the modified AAM algorithm, first proposed by Taylor et al. [95] and later modified by Stegmann [90], is used. The AAM algorithm is a statistical, deformable model-based algorithm, typically used to fit a previously trained model into an input image. One of the advantages of the AAM and similar algorithms is their ability to handle variability in the shape and the appearance of the modeled object due to prior knowledge. In this work, the AAM algorithm successfully obtains contours of the eyes and mouth in non-frontal images of individuals of different gender, race, facial expression, and head pose. Some of these individuals wore eyeglasses. Section 3.2 begins with a detailed description of the original AAM algorithm and its modification used in this work and then, proceeds to the details of how the AAM algorithm is used here.

In the second stage, contours of facial features obtained in the first stage are converted to a representation suitable for the classification to categories by a classification algorithm. Due to movements of the head, contours, obtained in the first stage, are at different locations in the image, have different sizes and are usually rotated at different angles. Moreover, due to non-perfect detection, a smooth original contour becomes rough after detection. This factors make classification of contours

using homography difficult. In order to perform robust classification of contours, a post processing stage is needed. The result of post processing should produce a contour representation, which is invariant to rotation, scaling and translation. To overcome non perfect detection, such a representation should be insensitive to small, local changes of a contour. In addition, to improve the robustness of the classification, the representation should capture the major shape information only and ignore fine contour details that are irrelevant for the classification. In this work, Fourier descriptors, first proposed by Zahn and Roskies [107], are used. Several comparisons [53, 58, 72, 110] show that Fourier descriptors outperform many other methods of shape representation in terms of accuracy, computational efficiency and compactness of representation. Fourier descriptors are based on an algorithm that performs shape analysis in the frequency domain. The major drawback of Fourier descriptors is their inability to capture all contour details with a representation of a finite size. To overcome non-perfect detection by the AAM algorithm, the detected contour is first smoothed and then Fourier descriptors are calculated. Therefore, a representation of the finest details of the contour that would not be well-captured by the method are removed. Moreover, the level of detail that can be represented using this method is easily controlled. Section 3.3 contains a discussion of the Fourier descriptors and details about their usage in this work.

In the third stage, contours are classified into categories. A classification al-

gorithm is an algorithm that selects a hypothesis from a set of alternatives. The algorithm may be based on different strategies. One is to base the decision on a set of previous observations. Such a set is generally referred in the literature as a training set. In this research, the best results are obtained using the k-Nearest Neighbors algorithm [32]. The k-Nearest Neighbors is one of the oldest, simplest, and most intuitive classification algorithms. Yet, as shown in Chapter 4, it demonstrated the best results classifying contours, that were processed using Fourier descriptors in the previous stage, into facial gestures. Zhang and Lu [109] suggested the use of the nearest neighbor algorithm, which is a special case of k-Nearest Neighbors algorithm, for searching and retrieving shapes, represented by Fourier descriptors. Section 3.4 contains a detailed description of the algorithm and its usage in this work.

3.2 Active Appearance Models (AAMs)

This section presents the main ideas behind AAMs, first proposed by Taylor et al. [95]. AAM is a combined model-based approach to image understanding. In particular, it learns the variability in shape and texture of an object that is expected to be in the image, and then, uses the learned information to find a match in the new image. The learned object model is allowed to vary; the degree to which the model is allowed to change is controlled by a set of parameters. Hence, the

task of finding the model match in the image, becomes the task of finding a set of model parameters that maximize the match between the image and modified model. The resulting model parameters are used for contour analysis in the next stages. The learned model contains enough information to generate images of the learned object. This property is actively used in the process of matching.

The shape in an AAM is defined as a triangulated mesh and can be expressed as:

$$s = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)^T \tag{3.1}$$

where, u_i and v_i are x and y coordinates of vertex i of the mesh. Basically, the vertices $(u_1, v_1, u_2, v_2, u_3, v_3)$ represent vertices of the first triangle in the mesh, traversed clockwise; the vertices $(u_4, v_4, u_5, v_5, u_6, v_6)$ represent vertices of the second triangle in the mesh; etc.

To simplify the process of the optimization, only the linear variation of a shape is allowed. In other words, any shape s can be expressed as a base shape s_0 plus a linear combination of m basis shapes s_i:

$$s = s_0 + \sum_{i=1}^{m} p_i s_i \tag{3.2}$$

where the coefficients p_i are shape parameters and vectors s_i are orthonormal, the shapes s_0 and s_i are of the form depicted in Equation 3.1.

The texture of the AAM is the pattern of intensities or colors across an image

patch. More specifically, the set of pixels u $= (u, v)^T$ that lie inside the base shape s_0 is denoted as a_0, so the texture or appearance of the AAM is then an image A(u) defined over the pixels u $\in a_0$. Similar to shapes, the AAM allows a linear variation of appearance i.e. the appearance A(u) can be expressed as a base appearance A_0(u) plus a linear combination of l basis appearance images A_i(u):

$$A(u) = A_0(u) + \sum_{i=1}^{l} \lambda_i A_i(u) \ \forall u \in a_0 \tag{3.3}$$

where the coefficients λ_i are the appearance parameters and images A_i are orthonormal.

Equations 3.2 and 3.3 are used to generate a model instance. Given the shape parameters p$=(p_1, p_2, \ldots, p_n)^T$, the shape s is generated, using Equation 3.2. Similarly, given appearance parameters $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)^T$, the appearance $A(u)$ of AAM defined over the a_0 is generated and then warped to the shape s. In particular, the pair of meshes s_0 and s defines a piecewise affine warp $W(u, p)$ from s_0 to s.

The goal of AAM fitting is, given the input image I, to minimize:

$$\sum_{u \in s_0} F[A(u) - I(W(u, p))] \tag{3.4}$$

simultaneously with respect to shape and appearance parameters λ_i and p_i. $A(u)$ is of the form depicted in Equation 3.3. $F(x)$ is an error norm function that will be described later. In general, the optimization is non-linear in the shape parameters

p, and linear in the appearance parameters λ. The problem of optimization can be solved, using any available method of the numerical optimization. Cootes et al. [23] proposed an iterative optimization algorithm and suggested multi-resolution models to improve the robustness and speed of model matching. According to this idea, in order to build the multi-resolution AAM of an object with k levels, the set of k images is built by successively scaling down the original image. For each image in this set, a separate AAM is created as described below. This set of AAMs is multi-resolution AAM with k levels. The matching of the multi-resolution AAM with k levels to an image is performed as follows: first, the image is scaled down k times, and the smallest model in the multi-resolution AAM, is matched to this scaled down image. The result of the matching is scaled up and matched to the next model in the AAM. This procedure is performed k times until the largest model in the multi-resolution AAM is matched to the image of the original size. This approach is faster and more robust than the approach that matches the AAM to the input image directly.

The main purpose of building an AAM is to learn the possible variations of object shape and appearance. However, it is impractical to take into account all of the possible variations of shape and appearance of object. Therefore, all observed variations of shape and appearance in training images are processed statistically in order to learn the statistics of variations that explain some percentage of all observed

variation. The best way to achieve this, is to collect a set of images of the object and manually mark the boundary of the object in each image. Marked contours are first aligned using the Procrustes analysis [39], and then, processed using PCA analysis [41] to obtain the base shape s_0 and the set of m shapes that can explain a certain percentage of shape variation. Similarly, to obtain the base appearance A_0 and the appearance variation A_i, training images are first normalized by warping the training shape to the base shape s_0, and then, PCA analysis is performed in order to obtain l images that can explain a certain percentage of variation in the appearance. For more detailed description of AAMs, the reader is referred to [23, 27, 95].

The introduction of AAMs has attracted much attention among researchers and numerous improvements and modifications have been proposed. In this work, the modified version of AAM, proposed by Stegmann [90], is used. The modifications of original AAMs that were used in the current work are summarized in the following subsections.

3.2.1 Increased Texture Specificity

As described above, the accuracy of AAM matching is greatly affected by the texture of the object. If the texture of the object is uniform, AAM tends to produce contours that lie inside the real object. This happens because the original AAM

algorithm is trained on the appearance inside of training shapes; it has no way to discover boundaries of an object with a uniform texture. To overcome this drawback, Stegmann [90] suggested the inclusion of a small region outside the object. Assuming that there is a difference between the texture of the object and background, it is possible for the algorithm to accurately detect boundaries of the real object in the image. Due to the fact that the object may be placed on different backgrounds, a large outside region included in the model may badly affect the performance of the algorithm. In this work, a strip that is 1 pixel wide around the original boundary of the object, as suggested in [90], is used.

3.2.2 Robust similarity measure

According to Equation 3.4 the performance of the AAM optimization is greatly affected by the measure, or more formally, the error norm, by which texture similarity is evaluated, and denoted as $F(x)$ in the equation. The quadratic error norm, also known as least squares norm or L_2 norm, is one of the most popular among the many possible choices of error norm. It is defined as:

$$F(e) = e^2 \tag{3.5}$$

where e is the difference between the image and reconstructed model. Due to the fast growth of function x^2, the quadratic error norm is very sensitive to outliers, and

thus, can affect the performance of the algorithm. Stegmann [90] suggested the usage of the Lorentzian estimator, which was first proposed by Black and Rangarajan [16], and defined as:

$$F(e, \sigma_s) = \log(1 + \frac{e^2}{2\sigma_s^2}) \tag{3.6}$$

where e is the difference between the textures of the image and the reconstructed AAM model; σ_s is a parameter that defines the values considered as outliers. The Lorentzian estimator grows much slower than a quadratic function, and thus, it is less sensitive to outliers and hence it is used in this research. According to Stegmann [90], the value of σ_s^2 is taken equal to the standard deviation of appearance variation.

3.2.3 Initialization

The performance of the AAM algorithm depends highly on the initial placement, scaling and rotation of the model in the image. If the model is placed too far from the true position of the object, it may not find the object or mistakenly matches the background as an object. Thus, finding good initial placement of the model in the image, is a critical part of the algorithm. Generally, initial placement or initialization depends on the application, and may require different techniques for different applications to achieve good results. Stegmann [90] proposed a technique to find the initial placement of a model that does not depend on the application. The idea is to test any possible placement of the model, and build a set of most probable

candidates for the true initial placement. Then, the algorithm tries to match the model to the image at every initial placement from the candidate set using a small number of optimization iterations. The placement that produces the best match is selected as a true initial placement. After the initialization, the model at the true initial placement is optimized using a large number of optimization iterations. This technique produces good results at the expense of a high computational cost. In this research, a grid with a constant step is placed over the input image. The value of the step is obtained empirically as described in Section 3.5. At each grid location, the model is matched with the image at different scales. To improve the speed of the initialization, only a small number of initialization iterations is performed at this stage. Pairs of location and scale, where the best match is achieved, are selected as a candidate set. In the next stage, a normal model match is performed at each location and scale from the candidate set, and the best match is selected as the final output of the algorithm. This technique is independent of application and produces good results in this research. However, the high computational cost makes it inapplicable in applications requiring real time response. In this research, the fitting of a single model may take more than a second in the worst cases, which is unacceptable for the purposes of real-time monitoring the user.

3.2.4 Fine-tuning the model fit

The usage of prior knowledge when matching the model to the image, does not always lead to an optimal result because the variations of the shape and the texture in the image may not be strictly the same as observed during the training [90]. However, it is reasonable to assume that the result produced during the matching of the model to the image, is close to the optimum [90]. Therefore, to improve the matching of the model, Stegmann [90] suggested the application of a general purpose optimization to the result, produced by the regular AAM matching algorithm. However, it is unreasonable to assume that there are no local minimums around the optimum and the optimization algorithm may become stuck at the local minimum instead of optimum. To avoid a local minima near the optimum, Stegmann [90] suggested the usage of a simulated annealing optimization technique, which was first proposed by Kirkpatrick et al. [56], a random-sampling optimization method that is more likely to avoid local minimum and hence it is used in this research.

3.2.5 Usage in current research

As mentioned above, in this work, the contours of the eyes and mouth define a facial gesture for the proposed system. Hence, in this research, the model consists of 3 shapes, where each shape consists of 64 landmarks. The choice of the number of

landmarks for each shape is determined by the fact that the shapes obtained by the AAM matching are used for representation a shape by using Fourier descriptors. According to the experiments conducted by Zhang and Lu [110], the use of 64 points per shape produced the best results for shape classification. To improve the accuracy of the match, a multi-resolution AAM model with five levels is built. To make the model more compact, the percentage of shape variation that can be explained, using Equation 3.2, is chosen to be 95%, as suggested in [90]. Similarly, the percentage of appearance variation that can be explained, using Equation 3.3, is also selected to be 95% [90]. In addition, as observed during the experiments, the heads of people sitting in the wheelchair were located in approximately the same area in the image. Hence, it is possible to find the best initial placement for the model by testing only a relatively small area of the image.

The described AAM algorithm is not capable of rejecting images that do not contain the trained model. In other words, the algorithm always fits a trained model to any image. To reject spurious matches, the fitted model is classified by its similarity measure. If the match is classified as a valid facial gesture image, it is passed to the next stage of the algorithm for further processing; otherwise the match is rejected. The classification algorithm is described in Section 3.5.

To train the AAMs, the boundaries of eyes and mouth were manually delineated in the images of the training set, and then, each boundary was normalized to

have 64 landmarks, placed equidistantly. Such processing creates the shape for AAM training and matching consisting of 192 landmarks. For details regarding the training model in this research, the reader is referred to Chapter 4.

3.3 Fourier Descriptors

Contours, obtained in the previous stage, are not suitable for classification because it is difficult to define a robust and reliable similarity measure between two contours, especially when neither centers nor sizes nor orientations of these contours coincide. Similarly to [21], the contours may be classified by their model parameters as produced by AAM algorithm. However, to produce good recognition results, such an approach requires very extensive training. Therefore, there is a need to obtain some sort of shape descriptor for these contours. Shape descriptors represent the shape in a way that allows robust classification, which means that the shape representation is invariant under translation, scaling, rotation, and noise due to imperfect model matching. There are many shape descriptors available. In this work, Fourier descriptors, first proposed by Zahn and Roskies [107], are used. Fourier descriptors provide compact shape representation, and outperform many other descriptors in terms of accuracy and efficiency [53, 58, 72, 110]. Moreover, Fourier descriptors are not computationally expensive and can be computed in real time. The performance of the Fourier descriptors algorithm is due to the fact that it processes contours

in the frequency domain, and it is much easier to obtain invariance to rotation, scaling, and translation in the frequency domain than in the spatial domain. This fact, along with simplicity of the algorithm and its low computational cost, are the main reasons for selecting this algorithm for usage in this research.

The Fourier descriptor of a contour is a description of the contour in the frequency domain that is obtained by applying the discrete Fourier transform on a shape signature and normalizing the resulting coefficients. The shape signature is a one dimensional function, representing two dimensional coordinates of contour points. The choice of the shape signature has a great impact on the performance of Fourier descriptors. Zhang and Lu [109] recommended the use of a centroid distance shape signature that can be expressed as follows. Suppose the shape consists of L points represented as a set $(x(t), y(t)), t = 0, 1, \ldots, L-1$ where $x(t)$ and $y(t)$ are x and y coordinates of the t^{th} shape point. Then, the centroid distance shape signature is expressed as the Euclidean distance of the contour points from the contour centroid (x_c, y_c) or formally:

$$r(t) = \sqrt{(x(t) - x_c)^2 + (y(t) - y_c)^2} \tag{3.7}$$

. This shape signature is translation invariant due to the subtraction of shape centroid and therefore, Fourier descriptors that are produced, using this shape signature, are also translation invariant.

Despite the fact that AAMs for the first stage were trained using contours with

landmarks and placed equidistantly, the landmarks of contours produced by the first stage are not placed equidistantly due to deformation of the model shape during the match of the model to the image. In order to obtain a better description of the contour, the contour should be normalized. The main purpose of normalizing is to ensure that all parts of the contour are taken into consideration, and improve the efficiency and insensitivity to noise of Fourier descriptors by smoothing the shape. Zhang and Lu [109] compared several methods of contour normalization and suggested that the method of equal arc length sampling produces the best result among other methods. According to this method, landmarks should be placed equidistantly on the contour or in other words, the contour is divided into arcs of equal length, and the end points of such arcs form a normalized contour. Then, the shape signature function is applied to the normalized contour, and the discrete Fourier transform is calculated on the result according to the equation:

$$F_n = \frac{1}{N} \sum_{t=0}^{N-1} r(t) e^{\frac{-j2\pi nt}{N}} \tag{3.8}$$

where $r(k)$ is a shape signature, defined in Equation 3.7.

Note that the rotation of the boundary will cause the shape signature, used in this research, to shift. According to the time shift property of the Fourier transform, it causes a phase shift of Fourier coefficients. Thus, taking only a magnitude of the Fourier coefficients and ignoring the phase provides invariance to rotation. In addition, the output of the shape signature are real numbers, and according to the

property of discrete Fourier transform, Fourier coefficients of a real-valued function are conjugate symmetric. However, only the magnitude of Fourier coefficients are taken into consideration, which means that only half of the Fourier coefficients have distinct values. The coefficient $|F_0|$ represents the scale of the contour only, so it is possible to normalize the remaining coefficients by dividing by $|F_0|$ in order to achieve invariance to scaling. The resulting Fourier descriptors can be calculated as $FD = (\frac{|F_1|}{|F_0|}, \frac{|F_2|}{|F_0|}, \ldots, \frac{|F_{N/2}|}{|F_0|})$, where $F_0, F_1, \ldots F_{N/2}$ are calculated according to Equation 3.8. The fact that only the first few Fourier coefficients are taken into consideration allows Fourier descriptors to catch the most important shape information and ignore fine shape details and boundary noise. As a result, a compact shape representation is produced, which is invariant under translation, rotation, scaling, and insensitive to noise. Such a representation is appropriate for classification by various classification algorithms.

3.3.1 Usage in this research

The output of the first stage is the boundaries of eyes and mouth that each contains 64 landmarks. For each boundary, Fourier descriptors are calculated as described above, producing as output, 3 Fourier descriptor vectors of length 31. For the sake of simplicity of classification, these vectors are concatenated to a single vector of length 93. During the testing of the algorithm, it was discovered that combining

the vector of Fourier descriptors with elongations of contours, slightly improves the performance of the classification. One of the reasons for the improvement of the classification can be the fact that the contour elongation has good discriminating ability for the typical shapes of eyes and mouth. The elongation of a contour can be calculated using the equation:

$$E_{contour} = \frac{S_{contour}}{L^2_{contour}} \tag{3.9}$$

where $L_{contour}$ is a length of the contour, and $S_{contour}$ is an area of the contour. The length of the contour can be calculated as a sum of distances between consequent landmarks. According to [13], and assuming that the contour does not intersect itself, the signed area of the contour with landmarks $(x_1, y_1), \ldots, (x_n, y_n)$ can be calculated according to the following equation:

$$S = \frac{1}{2}\left(\begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} + \begin{vmatrix} x_2 & x_3 \\ y_2 & y_3 \end{vmatrix} + \ldots + \begin{vmatrix} x_n & x_1 \\ y_n & y_1 \end{vmatrix}\right) \tag{3.10}$$

where $|X|$ denotes a determinant. Equation 3.10 can be written as:

$$S = \frac{1}{2}(x_1y_2 - x_2y_1 + x_2y_3 - x_3y_2 + \ldots + x_ny_1 - x_1y_n) \tag{3.11}$$

In Equation 3.9, the absolute value of the area, calculated by Equations 3.10 or 3.11, should be used.

The resulting vector, representing the contours of eyes and mouth, is passed to the next stage for classification.

3.4 k-Nearest Neighbors classification

The third stage performs classification of facial features, obtained in the previous stage, into categories or in other words, it determines which facial gesture is represented by the detected boundaries of the eyes and mouth. This stage is essential because boundaries represent numerical data, whereas the system is required to produce facial gestures corresponding to boundaries or in other words, the system is required to produce categorical output. The task of classifying items into categories attracts much research, and numerous classification algorithms have been proposed. For this research, a group of algorithms that learn categories from training data and predict the category for an input image, is suitable. In the literature, these algorithms are called supervised learning algorithms. Generally, no algorithm performs equally in all applications, and it is impossible to analytically predict which algorithm will have the best performance in the application. In the case of Fourier descriptors, Zhang and Lu [109] recommended classification according to the nearest neighbor, or in other words, Fourier descriptor of the input image is classified according to the nearest, in terms of Euclidean distance, Fourier descriptor of the training set. In this research, the generalization of this method, known as the k-Nearest Neighbors which was first proposed by Fix and Hodges [32], is used. In this work this method produced better results than SVM [24] and its modifica-

tion, NuSVM [85]. The general idea of the method is to classify the input sample by a majority of its k nearest, in terms of some distance metrics, neighbors from the training set. k is a positive integer. Specifically, distances from an input sample to all stored training samples are calculated and k closest samples are selected. The input sample is classified by majority vote of k selected training samples. A major drawback of such an approach is that classes with more training samples tend to dominate the classification of an input sample. Figure 3.2 illustrates classification of an input sample by k-Nearest Neighbors algorithm. In this figure green circle represents an input sample, blue rectangles and red triangles represent training samples belonging to two different classes. In the specific example, if the value of k is selected to be equal to three an input sample is classified as a red triangle because there are two red triangles among three nearest neighbors of the input sample; if the value of k is selected to be equal to five an input sample is classified as a blue rectangle because there are three blue rectangles among five nearest neighbors of the input sample. The distance between two samples can be defined in many ways. In this research, Euclidean distance is used as a distance measure. The Euclidean distance between samples $P = (p_1, p_2, \ldots, p_n)$ and $Q = (q_1, q_2, \ldots, q_n)$ is defined as:

$$D = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{3.12}$$

The process of training of k-Nearest Neighbors is simply caching of training samples in internal data structures. Such an approach is also called in the literature, as lazy

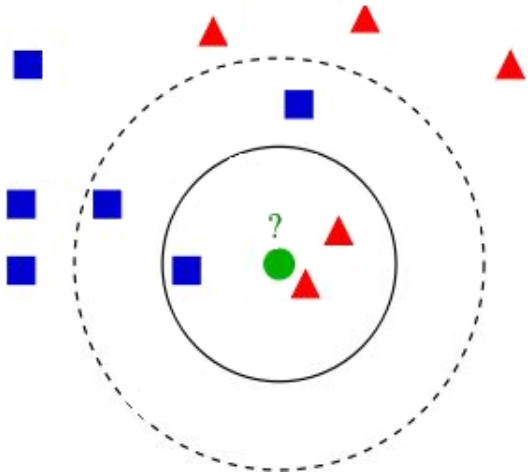


Figure 3.2: Example of k-Nearest Neighbors classification

learning [3]. To optimize the search of nearest neighbors some sophisticated data structures, e.g. Kd-trees [25], might be used. The process of classification is simply finding the k nearest, cached training samples, and deciding the category of the input sample. The value of k has a significant impact on the performance of the classification. Low values of k may produce a better result, but are very vulnerable to noise. Large values of k are less susceptible to noise, but in some cases, the performance may degrade. The process of selection of the optimal value of k is described in Section 3.5. The result of the classification, produced by this stage, is a final result of the static facial gesture recognition system.

3.4.1 Usage in this research

To obtain a fast and effective implementation of the algorithm, the OpenCV Library [80] is used in this work. This library is very popular for the fast and effective implementation of many algorithms, used in computer vision. To the best of the author's knowledge, the version of the OpenCV library used in this workdoes not use any sophisticated data structures, e.g. Kd-trees [25], and uses a straight-forward algorithm to find nearest neighbors of an input sample. The input for this stage is a vector of length 93 representing a facial gesture. This vector is produced in the previous stage by concatenation of Fourier descriptors of the contours of the eyes and mouth and elongations of these contours. In this work the best results were produced by selecting the value of k equal to 1 and by training the classifier using randomly selected 30% of all contours. The optimal value of k as well as a number of contours to train the classifier are determined empirically. For the details regarding the training of the classifier, selection of values of k and performance of the classification, the reader is referred to Section 3.5.

The described classifier is not capable of rejecting samples that are not similar to the trained samples. In other words, the algorithm always classifies an input sample into a class. To reject the spurious classifications, the distance of the input sample from the nearest trained sample is checked against the threshold. The sample is

rejected if the distance is greater than the value of the threshold. The method to determine the value of the threshold is described in Section 3.5.

3.5 Selection of the optimal configuration of the algorithm

The purpose of selecting the optimal configuration is to find the values of various algorithm parameters that ensure the best recognition rate with the lowest false positive recognition rate.

Due to the fact that there are several parameters that affect the recognition rate and false positive recognition rate (e.g. initialization step of AAM algorithm, choice of classifier, number of samples used to train the classifier, number of neighbors for k-Nearest Neighbors classifier), the testing of all possible combinations of parameters is impractical. To simplify the process of finding the optimal configuration for the algorithm, the optimal initialization step of the AAM algorithm with an optimal number of training images and neighbors for k-Nearest Neighbors classifier are obtained. The obtained configuration is used to compare the performance of several classifiers and check the influence of adding shape elongation of eyes and mouth on the performance of the whole algorithm. In addition, this configuration is used to tune the spurious images classifier to improve the false positive recognition rate of the algorithm. This approach works under the assumption that the configuration that provides the best results without the classifier of the spurious

images will still produce the best results when the classifier is engaged.

Both the AAM and k-Nearest Neighbors algorithms do not have the ability to reject spurious samples automatically. However, the algorithm proposed in this work should be able to reject the facial gestures that are not considered as having special meaning and therefore not trained. To reject such samples, the confidence measures (similarity measure for the AAM algorithm; the shortest distance to training sample for k-Nearest Neighbors algorithm) should be evaluated to determine if the sample is likely to contain the valid gesture. The performance of such classification has a great impact on the performance of the whole algorithm. It is clear that any classifier will inevitably reject some valid images and classify some of the spurious images as valid. The classifier used in this work consists of two parts: the first part classifies the matches obtained by the AAM algorithm; the second part classifies the results obtained by the k-Nearest Neighbors classifiers. These parts are independent of each other and trained separately.

In this work, the problem of classifying spurious images is solved by analyzing the distribution of the values of confidence measures of valid images and classifying the images using simple thresholding. First, the part of the classifier that deals with results of the AAM algorithm is tuned. The results produced by the first part of the classifier are used to tune the second part of the classifier. While such an approach does not always provide the best results, it is extremely simple and

computationally efficient. Some ideas to improve the classifier are described in Chapter 5. For details on the tuning of the spurious image classifier, the reader is referred to Chapter 4.

Chapter 4 describes the process of selecting the optimal values of the parameters, which influence the performance of the algorithm. Due to the great number of such parameters and range of their values, testing of all possible combinations of values of the parameters goes beyond the scope of this research. In this research, the initialization step for the AAM algorithm, number of images for the training of the shape classifier, type of the shape classifier, and usage of shape elongation have been tested. It was found that the initialization step of 20×20, usage of shape elongations along with Fourier descriptors, k Nearest Neighbors classifier as a shape classifier with k equal to 1, and 2748 shapes to train the shape classifier, provide the best classification results. For the details on obtaining the values of these parameters, the reader is referred to Chapter 4.

4 Experimental results

4.1 Experimental design

In order to test the proposed approach, the software implementation of the system was tested on a set of images that depicted human volunteers producing facial gestures. The goal of the experiment was to test the ability of the system to recognize facial gestures, irrespective of the volunteer, and measure the overall performance of the system.

Due to the great variety of facial gestures that can be produced by humans by using their eyes and mouth, the testing of all possible facial gestures is not feasible. Instead, the system was tested on a set of ten facial gestures that were produced by volunteers. The participation of volunteers in this research is essential due to specificity of the system. The system is designed for wheelchair users, and to test such a system, images of people sitting in a wheelchair are required. Moreover, the current mechanical design of the wheelchair does not allow frontal images of a person sitting in the wheelchair, so the images should be acquired from the same angle as

in a real wheelchair. Unfortunately, there is no publicly available image database that contains such images. All volunteers involved in this research have normal face muscle control. This fact limits the validity of the results of the experiment to people with normal control of facial muscles. Signed consent was obtained for each volunteer to participate in the experiment as required by York University rules. The sample of consent is presented in Appendix A.

The experiment was conducted in a laboratory with a combination of overhead fluorescent lighting with natural lighting from windows of the laboratory. The lighting was not controlled during the experiment and remained more or less constant. To make the experiment closer to the real application, volunteers sat in the automatic wheelchair, and their images were taken by the camera mounted on the touch screen as described in Section 3.1. The mechanical design of the wheelchair allows the touch screen to move freely and therefore, it is impossible to fix the location of the camera relative to the face of a person sitting in the wheelchair. In addition, volunteers were allowed to move during the experiment in order to provide a greater variety of facial gesture views. Each of the ten volunteers produced ten facial gestures. Five volunteers wore glasses during the experiment; two were females and eight were males; two were of Asian origin and others of Caucasian origin. Such an approach allows the testing of the robustness of the proposed approach to the variability of facial gestures among different volunteers of different gender and ori-

gin. To make the testing process easier for volunteers, they were presented with samples of facial gestures and asked to reproduce the gesture as close as possible to the sample. The task of selecting proper facial gestures for the facial gesture recognition algorithm for monitoring system is very complex, because many samples of facial expressions of disabled people expressing genuine emotions need to be collected. Such work is beyond the scope of this research. The purpose of the experiments described in this chapter is to prove that the algorithm has the capability to classify facial expressions by testing it on a set of various facial gestures. Samples of facial gestures are shown in Figure 4.1. In addition, five volunteers produced various gestures to measure the false positive rate of the algorithm. The volunteers were urged to produce as many gestures as possible. However, to avoid testing the algorithm only on artificial and highly improbable gestures, some of the volunteers were encouraged to talk. The algorithm is very likely to deal with facial expressions produced during talking, so it is critical to ensure that the algorithm is robust enough to reject such facial expressions. Such an approach ensured that the algorithm was tested on a great variety of facial gestures.

Each gesture was captured as a color image at a resolution of 1024×768 pixels. For each volunteer and each facial gesture, 100 images were taken, which creates a resulting set of 10000 images. However, not every image in the resulting set is acceptable for further processing. Blinking, for example, confuses the system be-

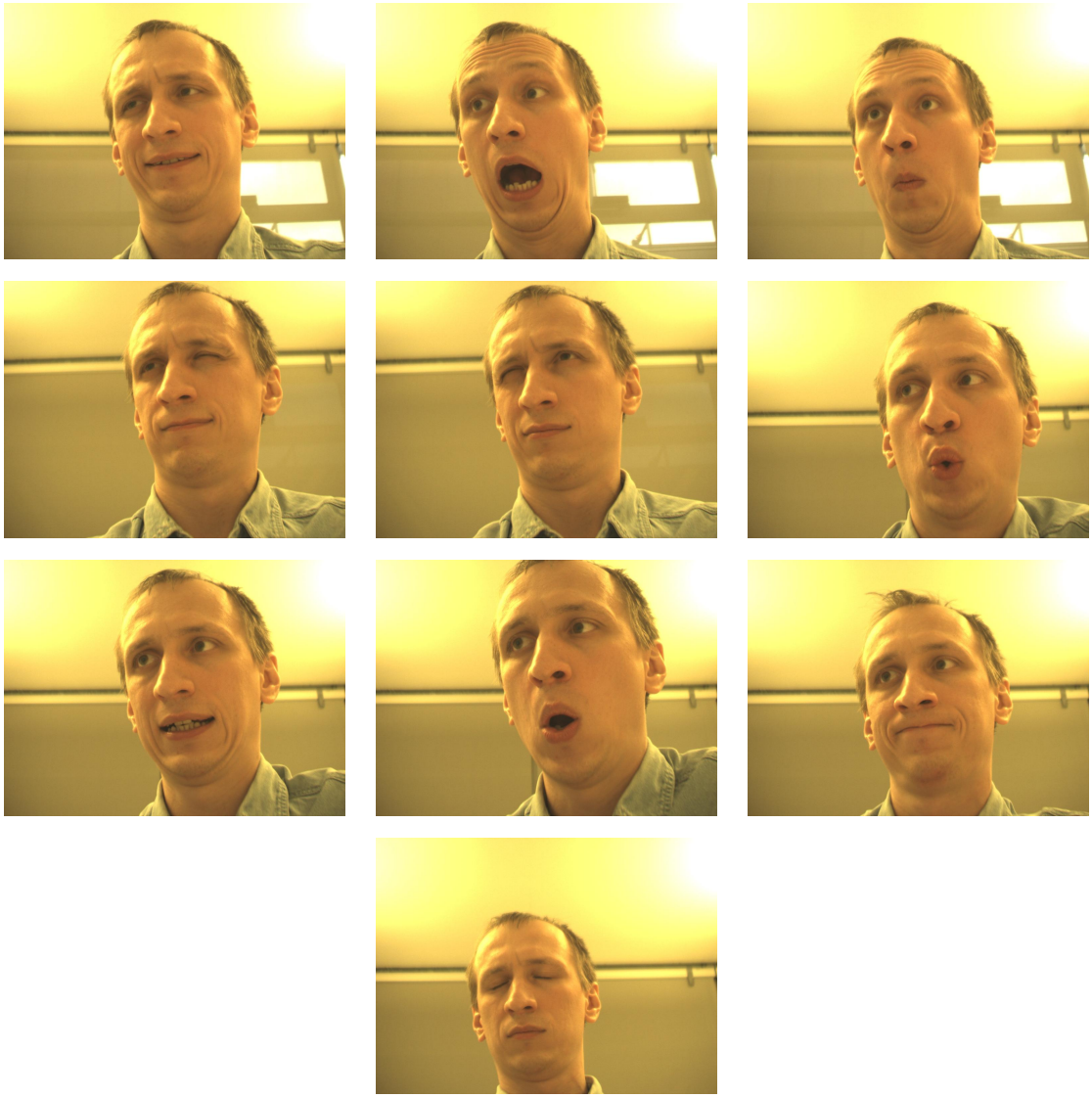


Figure 4.1: Facial gestures recognized by the system

cause closed eyes are part of a separate gesture. In addition, due to the limited field of view of the camera, accidental movements may cause the eyes or mouth to be occluded. Such images can not be processed by the system because the system requires both eyes and the entire mouth be clearly visible in order to recognize the facial gesture. These limitations are not an inherent drawback of the system. Blinking, for instance, can be overcome by careful selection of facial gestures; occlusions can be treated using more sophisticated contour detection techniques that will be briefly described in Chapter 5. Out of a resulting set of 10000 images, 9140 images were manually selected for training and testing of the algorithm. Similarly, to test the algorithm for false positive rate, each of 5 volunteers produced 100 facial gestures. Out of a resulting set of 500 images, 440 images were selected manually for testing of the algorithm. Examples of images, used for the training and testing of the system, are shown in Figure 4.2. Samples of facial gestures used for testing false positives are shown in Figure 4.3.

4.2 Training of the system

The task of training the system consists of two parts. First, the system is trained to detect contours of the eyes and mouth of a person sitting in the wheelchair. Then, the system is trained to classify the contours of the eyes and mouth to facial gestures. Generally, training of both parts can be performed independently, using

Figure 4.2: Sample images processed by the system

Figure 4.3: Sample images used to measure the system for false positive rate

manually marked images. However, in order to speed up the training and achieve better results, the training of the second part is performed, using results obtained by the first part. In other words, the first stage is trained using manually marked images; the second stage is trained using contours which are produced as a result of the processing of input set of images by the first part. This approach produces better final results because the training of the second stage is performed, using real examples of contours. The training, using real examples that may be encountered as input, generally produces better results than using manually or synthetically produced examples, because it is impossible to accurately predict the variability of input samples and reproduce it in training samples. In addition, such an approach facilitates and accelerates the process of training for the system, especially when the system is retrained for a new person. In this work, the best results are obtained using 100 images to train the first part of the system and 2748 contours to train the second part of the system.

4.2.1 Training of AAMs

The performance of AAMs has a crucial influence on the performance of the whole system. Therefore, the training of AAMs becomes crucial for the performance of the system. As described in Chapter 3, AAMs learn variability of training images to build a model of eyes and mouth, and then, try to fit the model to an input image.

To provide greater reliability of the results of these experiments, several volunteers participated in the research. However, a model built from training samples of all participants leads to poor detection and overall results. This phenomenon is due to the great variability among images of all volunteers that can not be described accurately by a single model. To improve the performance of the algorithm, several models are trained. Models are trained independently, and each model is trained on its own set of training samples. The fitting to the input image is also performed independently for each model, and the result of the algorithm is a model that produces the best fit to the input image. Generally, the algorithm that uses more trained models, tends to produce better results due to more accurate modeling of possible image variability. However, due to the high computational cost of fitting an AAM to the input image, such an approach is impractical in terms of processing time. Selecting the optimal number of models is not an easy task. There are techniques that allow selecting the number of models automatically. In this work, a simple approach has been taken: each model represents all facial gestures, produced by a single volunteer. While this approach is probably not optimal in terms of accuracy of modeling, the variability and number of models, it has clear advantage in terms of simplicity and ease of use. This technique does not require a great number of images in a training set: one image for each facial gesture and volunteer is enough to produce acceptable results. To build the training set from each set of

100 images representing a volunteer producing a facial gesture, one image is selected randomly. As a result, the training set for AAM consists of only 100 images. To train an AAM model, the eyes and mouth are manually marked on these images. The marking is performed, using custom software, which allows the user to draw and store the contours of eyes and mouth over the training image. These contours are then normalized to have 64 landmarks that are placed equidistantly on the drawn contour. Samples of contours that are used in the research are shown in Fig. 4.4. The images and contours of every volunteer are grouped together, and a separate AAM model is trained for each volunteer. Such an approach has a clear advantage when the wheelchair has only a single user. In fact, this represents the target application.

Each AAM is built as a five level multi-resolution model as described in Chapter 3. The percentage of shape and texture variation that can be explained, using the model is selected to be 95%. In addition to building the AAM, the location of the volunteer's face in each image is noted. These locations are used to optimize the fitting of an AAM to an input image by limiting the search for the best fit by a small region, where the face is likely to be located.

As mentioned in Chapter 3, the performance of the AAM fitting depends on the initial placement of the model. In this research, it is proposed that a grid be placed over the input image and to fit the model at each grid location. The location where

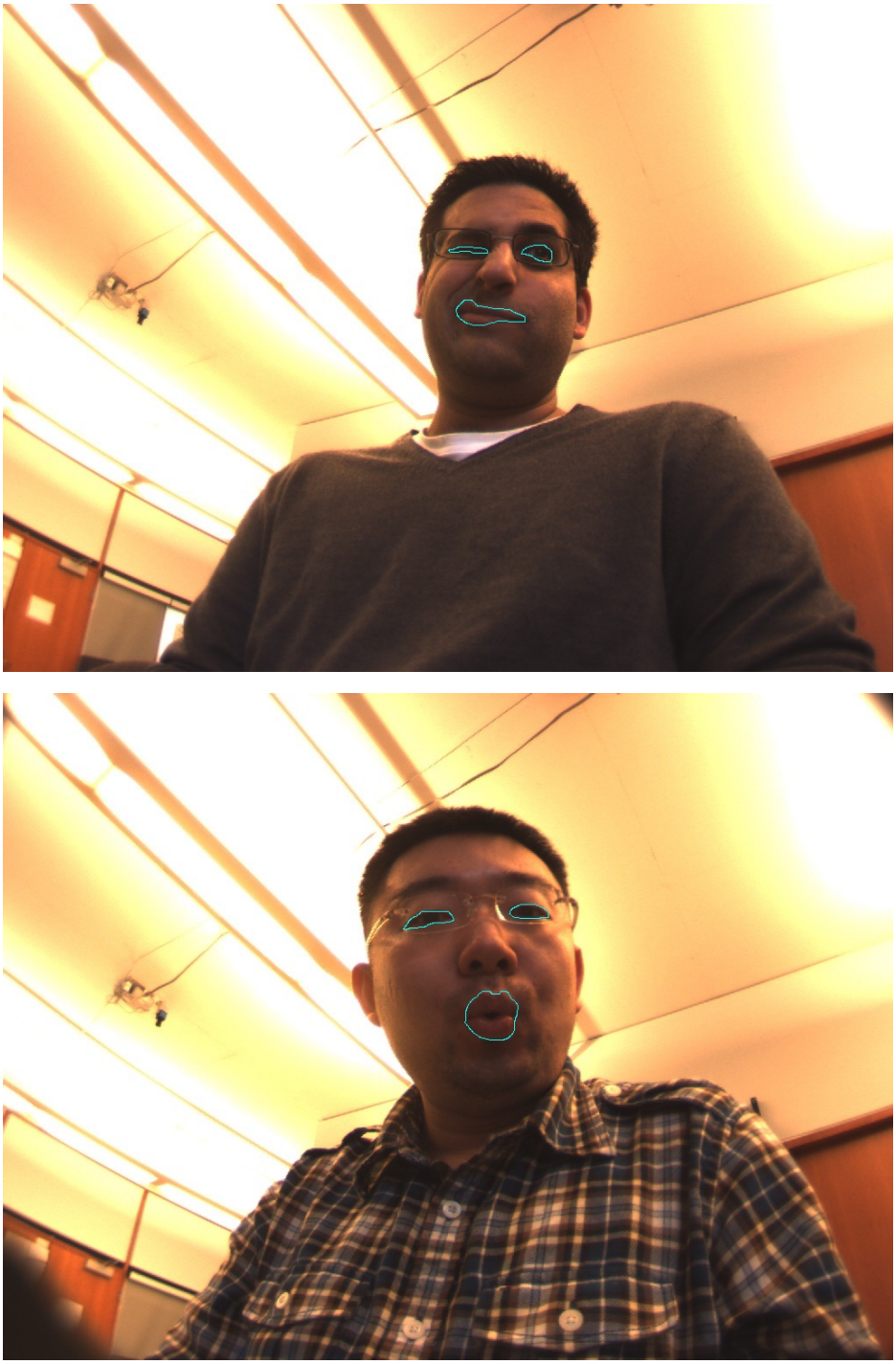


Figure 4.4: Sample images used to train AAM

Figure 4.4: Sample images used to train AAM

Figure 4.4: Sample images used to train AAM

the best fit is obtained, is considered the true location of the model in the image. Therefore, the size of the grid has a great impact on the performance of fitting of the model. The usage of the the small grid obtains excellent fitting results, but has prohibitively high computational cost, whereas the usage of the last grid has a low computational cost, but leads to poor fitting results. In this research, the optimal size of the grid was empirically determined to be 20×20. In other words, the initialization grid, placed on the input image, has 20 locations in width and 20 locations in height. Therefore, the AAM algorithm tests 400 locations during the initialization phase of the fitting. The size of the grid was chosen after series of experiments to select the optimal value. The results are presented in Section 4.3.

As mentioned in the Section 3.5 the AAM algorithm can not reject spurious images. To reject the spurious images, the statistics about similarity measures of valid images and spurious images is collected. The spurious images are detected using simple thresholding. The collected statistics and the value of the threshold are presented in the Section 4.3.

4.2.2 Training of the shape classifier

The shape classifier is the final stage of the whole algorithm, so its performance influences the performance of the entire system. The task of the shape classifier is to classify the shapes of eyes and mouth, represented as a vector, to categories

representing facial gestures. To accomplish this task, this research uses a technique of supervised learning. According to this technique, in the training stage, the classifier is presented with labeled samples of the input shapes. The classifier learns training samples and tries to predict the category of input samples using the learned information. In this research, the k-Nearest Neighbors classifier is used for shape classification. This classifier classifies input samples according to the closest k samples from the training set. Naturally, a large training set tends to produce better classification results at the cost of large memory consumption and slower classification. Hence, it may be impractical to collect a large number of training samples for the classifier. However, a small training set may produce poor classification results. The number of neighbors k, according to which the shape is classified, also has an impact on the performance of the classification. Large values of k are less susceptible to noise, but may miss some input samples. Small values of k usually produce better classification, but are more vulnerable to noise.

To train the classifier, the input images are first processed by the AAM algorithm to obtain the contours of the eyes and mouth. Then, Fourier descriptors of each contour are obtained and combined to a single vector, representing a facial gesture. As a result, a set of 9140 vectors, representing the facial gestures of volunteers, is built. Out of these vectors, some are randomly selected to train the classifier. The remaining vectors are used to test the performance of the classifier. The results of

the testing are presented in Section 4.3.

As mentioned in Section 3.5, the k-Nearest Neighbors classifier can not reject shapes obtained from spurious images. To reject the spurious shapes, the statistics on the closest distance of the input sample to the training set of valid images and spurious images are collected. The spurious shapes are detected using simple thresholding. The collected statistics and the value of the threshold are presented in Section 4.3.

4.3 Results

The testing was performed on a computer that has 512 megabytes of RAM and 1.5 GHz Pentium 4 processor under Windows XP. To detect the contours of eyes and mouth, a slightly modified C++ implementation of AAMs, proposed by Stegmann [90], is used. To classify the shapes, the k-Nearest Neighbors classifier implementation of OpenCV Library [80] was used.

The input images were first processed by the AAM algorithm to obtain the contours of the eyes and mouth. Then, Fourier descriptors of each contour were obtained and combined to a single vector, representing a facial gesture. In the last stage, the vectors were classified by the shape classifier. The performance of the algorithm was measured according to the final results produced by the shape classifier.

To obtain the best performance of the algorithm, the optimal configuration of parameters should be obtained. Performing a full sensitivity analysis is beyond the scope of this work, but the experiments described in this chapter may provide some ideas about the influence of the parameters on the performance of the algorithm. The parameters of the algorithm are summarized in Table 4.1. Due to the large number of parameters that influence the performance of the algorithm, it is unpractical to test all the possible combinations of parameters. Hence, the optimal number of training images, number of neighbors for the shape classifier and size of the initialization grid of the AAM were determined first. The tuning of spurious images classifier as well as checking the influence of other parameters were performed using obtained optimal configuration.

The tested values of the size of the initialization grid for the AAM algorithm are 5×5, 10×10, 15×15, 20×20, and 25×25. The tested values of the number of neighbors of the k-Nearest Neighbors shape classifier are from 1 to 32. The tested values of the number of the shapes used for the training of the shape classifier are 919, 1833, 2748, 3661, 4572, 5488, 6401, 7316.

Figures 4.5 and 4.6 show the success rate of the classification as a function of the value of k for a different number of contours, used to train the classifier and the size of the initialization grid of the AAM algorithm. Figure 4.5 groups the data by the size of the initialization step of the AAM algorithm, while Figure 4.6

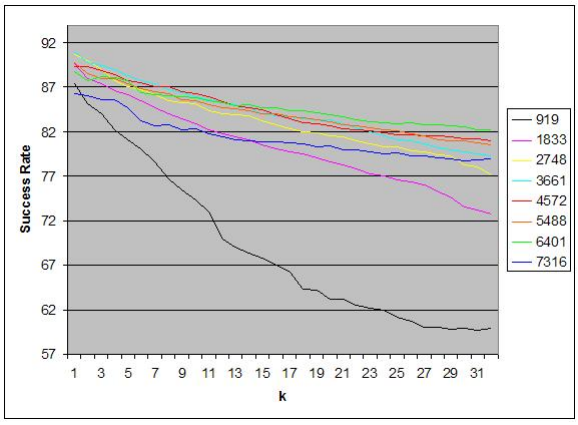
Parameter	Tested values	Best value in set
The size of the initialization grid of the AAM algorithm	5×5, 10×10, 15×15, 20×20, 25×25	20×20
The type of the shape classifier	k Nearest Neighbors, SVM, NuSVM	k Nearest Neighbors
The number of shapes used for the training of the shape classifier	919, 1833, 2748, 3661, 4572, 5488, 6401, 7316	2748
The number of neighbors for the k Nearest Neighbors shape classifier	1 . . . 32	1
Usage of elongation along with Fourier descriptors	Yes, No	Yes
Threshold for the AAM spurious images classifier		0.0089
Threshold for the shape classifier spurious images classifier		0.000958

Table 4.1: Summary of parameters of the algorithm

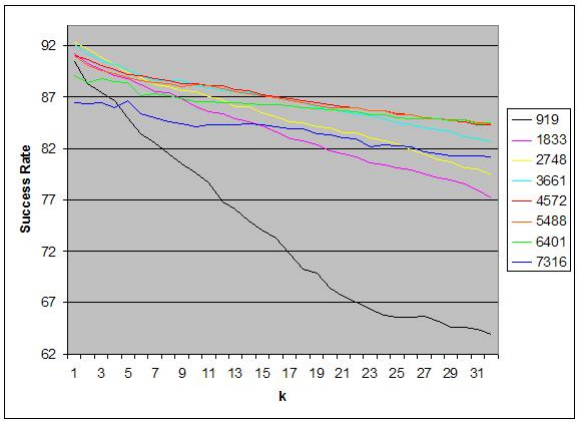
groups data by the number of shapes used to train the shape classifier. The graphs show that the configuration, where 919 shapes are used to train the shape classifier, performs poorest for any initialization grid size of the AAM. Other configurations have similar performances for any given size of the initialization grid. In addition, the recognition rates decrease for greater values of k. Generally, increase in the size of the initialization grid for the AAM leads to better recognition rate. However, the differences in the performance between 20×20 and 25×25 grids are small.

The graphs show that the 5×5 initialization grid performs poorly with any number of shapes used to train the shape classifier. Other configurations have very similar performances for any given number of shapes used to train the shape classifier. In addition, the recognition rates decrease for greater values of k. Generally, an increase in the number of the shapes used to train the shape classifier leads to better recognition rates. However, when the number of shapes used to train the shape classifier is greater than 4572, the differences in the performance are small.

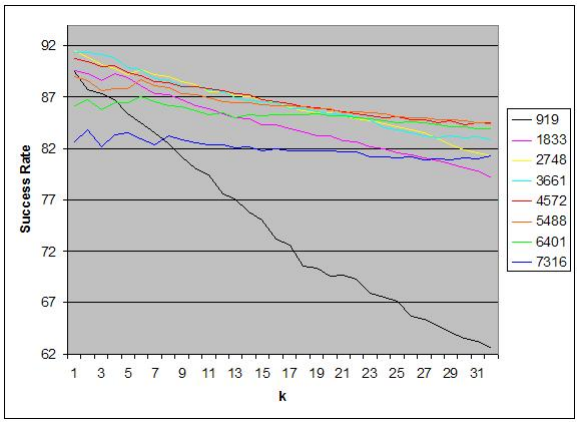
The experiments showed that the two configurations produce the highest recognition rates: a 20×20 grid to initialize the AAM, one neighbor for the shape classifier, 2748 shapes to train the shape classifier and a 25×25 grid to initialize the AAM, one neighbor for the shape classifier, and 3661 shapes to train the shape classifier. The latter configuration has a very slim advantage in terms of performance at the expense of a larger training set and higher computational cost (up



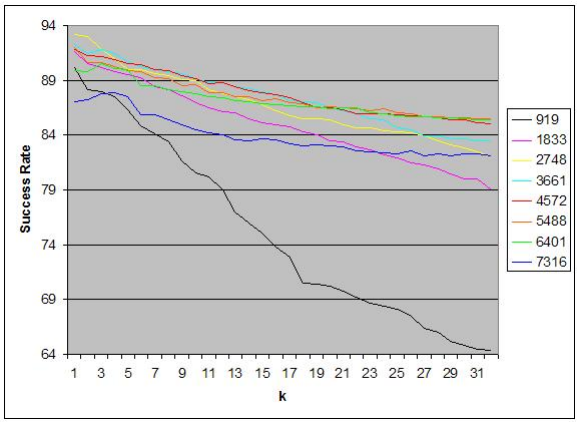
(a) 5×5 grid to initialize the AAM



(b) 10×10 grid to initialize the AAM

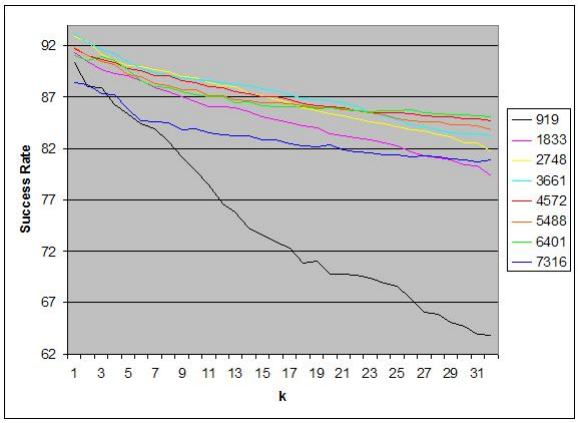


(c) 15×15 grid to initialize the AAM



(d) 20×20 grid to initialize the AAM

Figure 4.5: Recognition rate at various values of k and the number of training shapes grouped by the size of the initialization step of the AAM algorithm

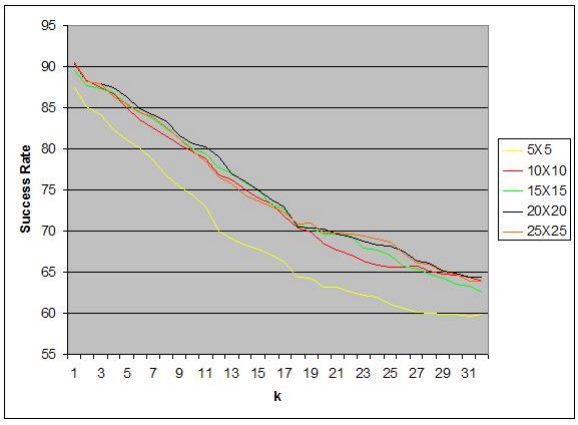


(e) 25×25 grid to initialize the AAM

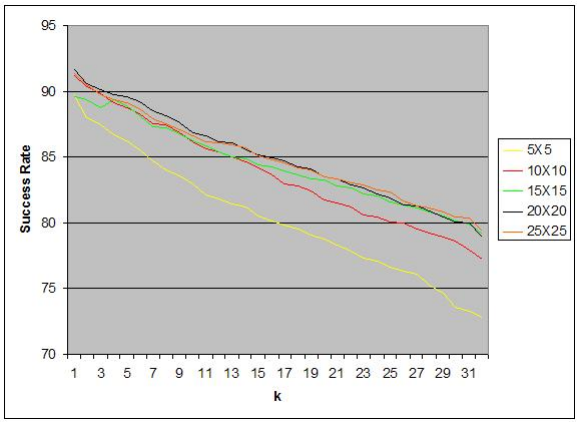
Figure 4.5: Recognition rate at various values of k and the number of training shapes grouped by the size of the initialization step of the AAM algorithm

to 45 seconds per image), so the first configuration is selected as a base for further experiments. The value of k is taken equal to one; the number of contours, used to train the classifier, is taken equal to 2748, which is 30% of all the contours and the grid size of the initialization of the AAM algorithm is taken equal to 20×20. The processing of the image in this configuration takes about 15 seconds on average. The most time consuming part of the algorithm is the AAM matching, and the time consumed by other components is negligible. Samples of contours, produced by the AAM algorithm, are shown in Fig. 4.7. The performance achieved by the algorithm in this configuration is 93.15%.

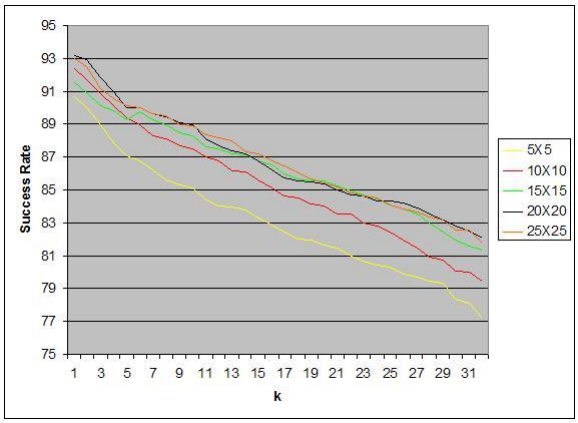
In addition, other classifiers, such as SVM [24] and its modification NuSVM



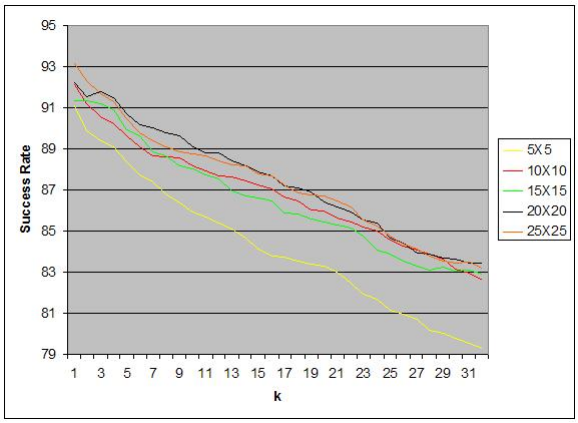
(a) 919 shapes to train the shape classifier



(b) 1833 shapes to train the shape classifier

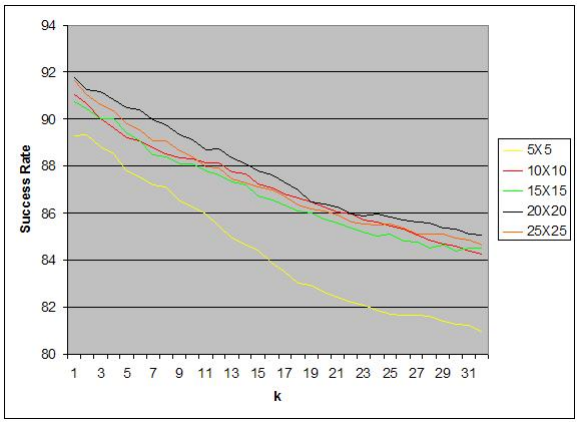


(c) 2748 shapes to train the shape classifier

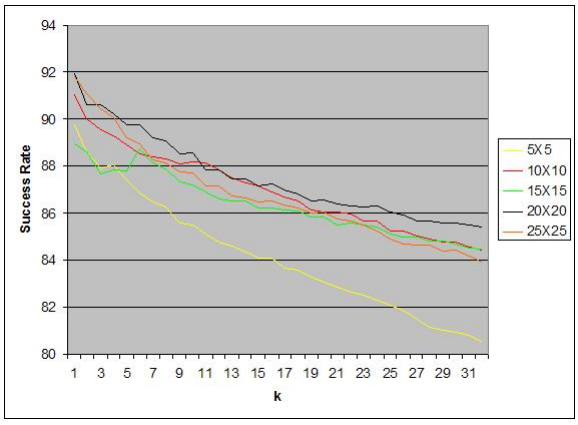


(d) 3661 shapes to train the shape classifier

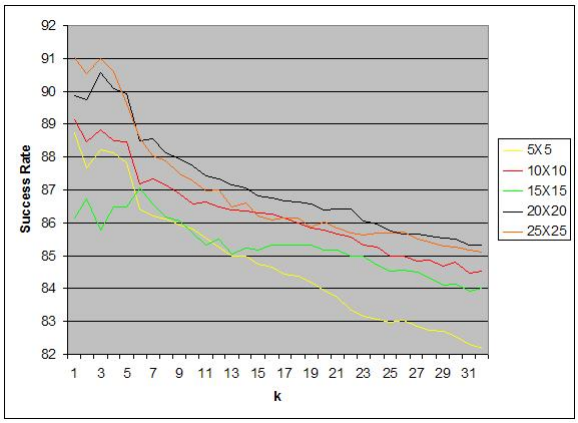
Figure 4.6: Recognition rate at various values of k and the size of the initialization step of the AAM algorithm grouped by the number of training shapes



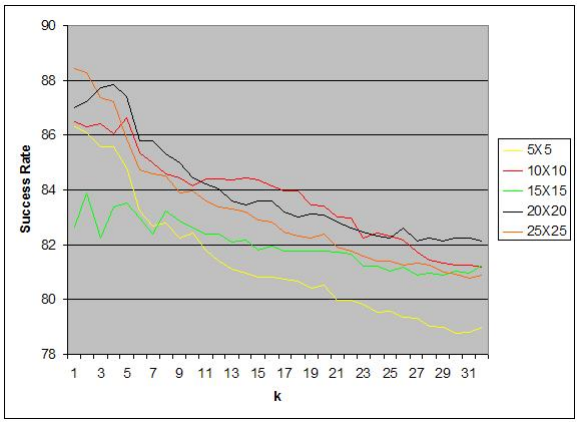
(e) 4572 shapes to train the shape classifier



(f) 5488 shapes to train the shape classifier



(g) 6401 shapes to train the shape classifier



(h) 7316 shapes to train the shape classifier

Figure 4.6: Recognition rate at various values of k and the size of the initialization step of the AAM algorithm grouped by the number of training shapes

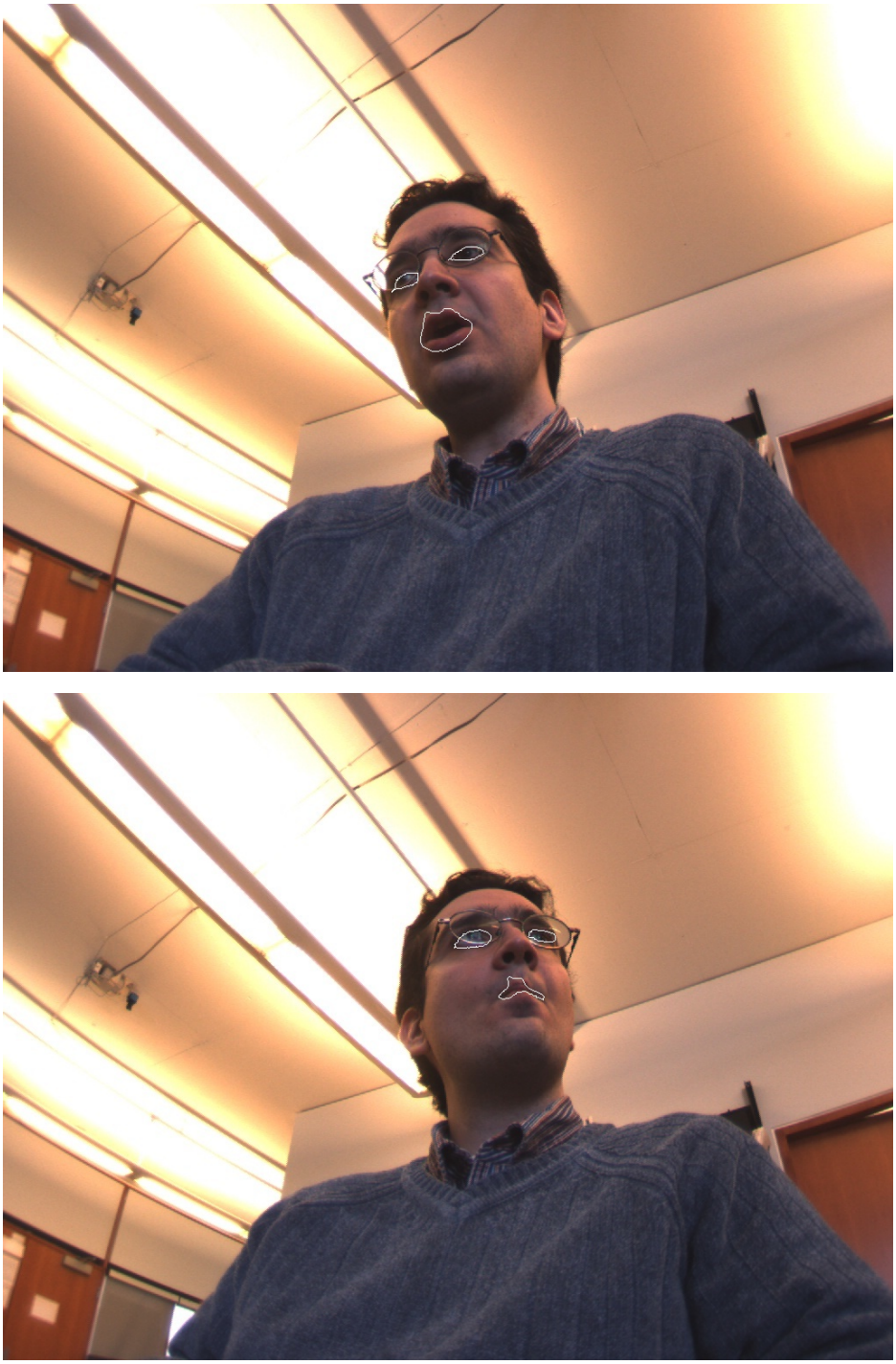


Figure 4.7: Sample images produced by AAM algorithm

Figure 4.7: Sample images produced by AAM algorithm

Figure 4.7: Sample images produced by AAM algorithm

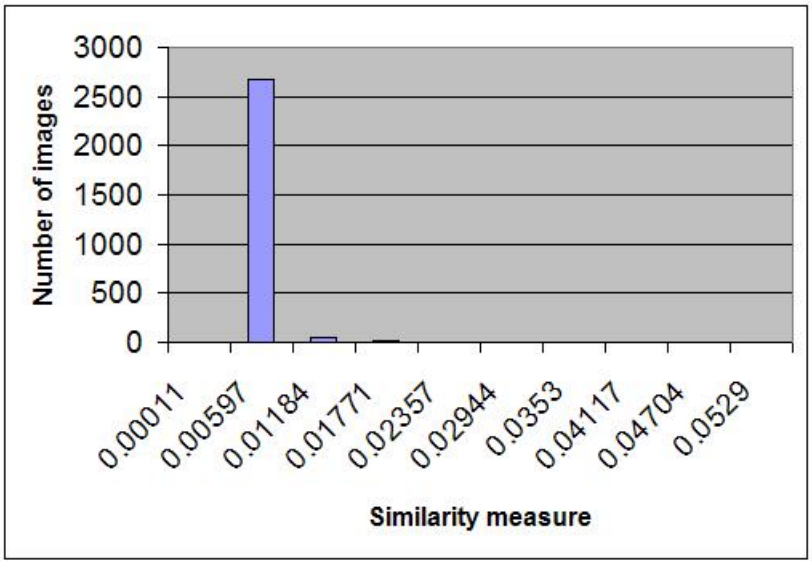
[85], implemented using the OpenCV Library[80], were tested. The algorithms were trained and tested on the same samples that were used in the optimal configuration described earlier in this section. The SVM classifier achieved an 84.03% success rate and NuSVM achieved an 83.32% recognition rate.

As mentioned in Section 3.3, the vectors, representing the shapes of eyes and mouth, which are used as input for the shape classifier contain elongations of the eyes and mouth shapes. To measure the influence of adding the elongations to the input for the shape classifier, additional experiments were performed. The shape classifier was trained and tested on the same samples that were used in the optimal configuration described earlier in this section. During this experiment, the shape classifier ignored the values of elongations of the eyes and mouth in the input vectors. The algorithm achieved a 93.01% recognition rate.

To tune the spurious image classifier of the AAM algorithm, the statistics of values of similarity measures was collected and analyzed. It was discovered that the distribution of value of similarity measure of all images is very close to those that were used to train the shape classifier. This observation allows the decrease of the number of images required to obtain the value of threshold for spurious image classifier for the AAM algorithm. It is very important from a practical point of view, because it allows the training of the classifier by using a small number of images. Figure 4.9 shows the distribution of values of similarity measures of all

images; Figure 4.8 shows the distribution of values of similarity measures of images used to train the shape classifier. As evident from the graphs, the vast majority of values are concentrated between 0.00011 and 0.00597. Due to the fact that distribution of the values of the similarity measure is very compact, the task of setting the threshold for classification of spurious images by the AAM algorithm is relatively simple and is set equal to 0.0089. This value was calculated as a middle of the histogram bin next to the bin where the majority of all similarity measure values are concentrated. After the determining of the threshold for the spurious image classifier for the AAM algorithm, all images were processed again by the AAM algorithm. There were 9011 images successfully processed, and the rest were rejected by the AAM algorithm. There were 2711 shapes, which is 30% of the successfully processed images, selected to train the shape classifier and the rest, 6300 shapes, for the testing of the performance of the classifier. The AAM algorithm successfully processed 234 out of 440 images, which is 53.18%. Such a high false positive rate is unacceptable for the system that is proposed in this work. Therefore, the shape classifier should be able to reject spurious shapes.

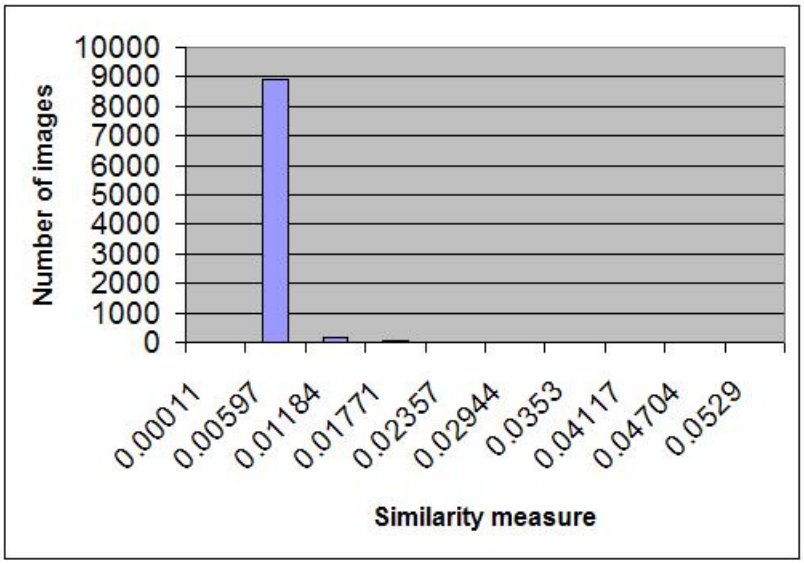
To determine the threshold for the shape classifier, the statistics of values of distances between the input shape and the nearest training shape were collected and analyzed. Figure 4.10 shows the distribution of the values of distances between the input shape and nearest training shape of all training images. As evident from



(a) Distribution of similarity measure values of images from shape classifier training set

Figure 4.8: The histogram of the distribution of similarity measure values of images used to train the shape classifier

the graph, the majority of values are concentrated below 0.000638. The distribution of the values is very compact. Hence, the task of setting the threshold for classification of spurious shapes by the shape classifier is relatively simple and set equal to 0.000958. This value is calculated as the middle of the histogram bin next to the bin where the majority of all values are concentrated. After engaging the spurious shape detector, the algorithm successfully recognized 5703 out of 6300 valid images, which is a 90% success rate. The algorithm recognized 27 out of 440

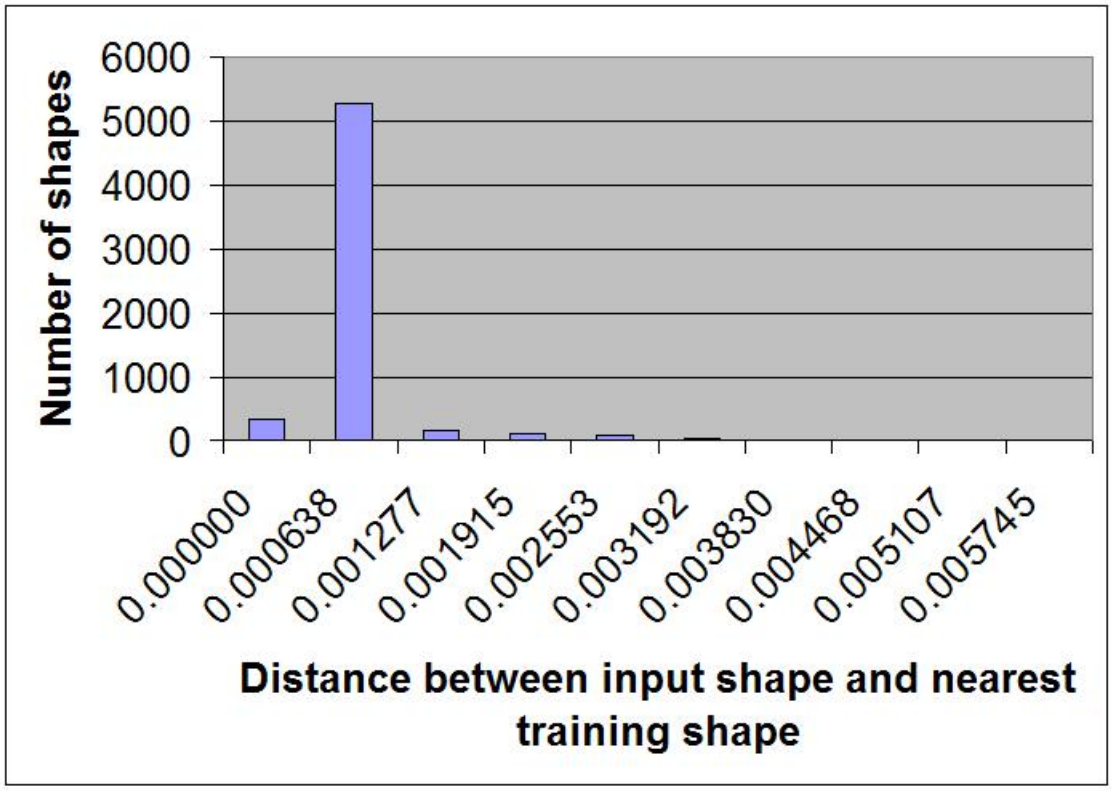


(a) Distribution of similarity measure values of all images

Figure 4.9: The histogram of the distribution of similarity measure values of all images

spurious images which is a 6% false positive rate. The shape classifier rejected 266 valid images and the AAM algorithm rejected 129 valid images. Therefore, in total the algorithm rejected 395 valid images, which is a 4% false negative rate.

Detailed results, showing the performance of the algorithm on each particular facial gesture, are shown in Table 4.2. Facial gestures are denoted by letters a,b,c,. . . ,j. Spurious gestures are denoted by letter s. The axes of the table represent the actual facial gesture (vertical) versus the classification result. Each cell (i,j) in the table holds the number of cases that were actually i, but classified as j.



(a) Distribution of distances between input shape and the nearest training shape

Figure 4.10: The histogram of the distribution of distances between input shape and the nearest training shape of all training images

The diagonal represents the count of correctly classified facial gestures. Table 4.3 summarizes performance of the algorithm on a set of spurious images. The details about rejected images are presented in Table 4.4.

	a	b	c	d	e	f	g	h	i	j
a	659	0	8	2	1	0	1	0	8	2
b	0	509	68	0	0	16	1	4	1	2
c	3	1	601	0	1	2	4	8	2	3
d	6	0	2	432	0	0	3	0	1	11
e	0	0	2	7	425	2	2	1	0	4
f	0	2	6	0	0	628	2	3	2	1
g	0	1	6	1	3	0	635	2	1	3
h	0	0	5	1	1	10	0	642	0	0
i	8	0	6	1	0	9	5	1	528	47
j	2	1	0	13	4	0	2	1	2	644

Table 4.2: Facial gesture classification results.

4.4 Discussion

The experiment was conducted on data consisting of ten facial gestures images, produced by ten volunteers. The images were typical indoor images of a human

a	b	c	d	e	f	g	h	i	j
0	0	2	3	4	12	2	4	0	0

Table 4.3: Spurious images classification results.

a	b	c	d	e	f	g	h	i	j
9	27	30	95	109	41	20	15	30	19

Table 4.4: Images rejected by the algorithm.

sitting in a wheelchair. The volunteers were of different origin and gender; some of them wore glasses. The location of the volunteer face relative to the camera could not be fixed due to the mechanical design of the wheelchair. Moreover, the volunteers were allowed to move during the experiment. The experiment was conducted according to the following procedure. First, the pictures of the volunteers were taken and stored. Next, a number of images were selected to train the first stage of the algorithm, to detect the contours of the eyes and mouth. After training, all images were run through the first stages of the algorithm to obtain the compact representations of facial gestures detected in the images. Some of these representations were used to train the last stage of the algorithm. The rest were used to test the last stage of the algorithm. The results of this test are presented in this chapter.

In addition, multiple facial gestures, produced by five volunteers, were collected to test the ability of the algorithm to reject spurious images.

Naturally, misclassification of a facial gesture by the system can occur due to the failure to accurately detect the contours of the eyes and mouth in the input image or misclassification of the detected contours to facial gestures. The reasons for the failure to detect the contours of the eyes and mouth include a large variation in the appearance of the face and insufficient training of AAMs. The great variation in the appearances can be explained by excessive distortion, caused by movements of the volunteers during the experiment, as well as natural variation in the facial appearance of the volunteer when producing a facial gesture. The reasons for inaccurate classification of the detected contours into facial gestures include inaccurate reproduction of the gestures by volunteers, insufficient discriminative ability of Fourier descriptors used in this work, and non optimal training of the classifier. The poor performance of SVM classifiers can be explained by the inability of such classifiers to separate the contours by a hyperplane with satisfactory accuracy. In addition, some images are misclassified by the spurious image classifier, which classifies the images by using a simple thresholding

Overall, the results demonstrate the ability of the system to recognize correctly, the facial gestures of different persons and suggest that the proposed approach can be used in automatic wheelchairs to obtain feedback from a user.

5 Conclusions

This report presented a new approach in monitoring a user of an automatic wheelchair and performed a feasibility analysis on this approach. Many approaches have been proposed to monitor the user of an automatic wheelchair. However the majority of these approaches monitor the user in dealing with low level direct control of a wheelchair or activating a limited set of simple automatic operations. Such approaches are usually inconvenient for the user and not suitable for the modern intelligent wheelchair. The approach proposed in this work suggests monitoring the user to obtain information about intentions and then using this information to make decisions automatically about the future actions of the wheelchair. The approach has a clear advantage over other approaches in terms of flexibility and convenience to the user. The report examines feasibility and suggests the implementation of a component of such a system that monitors the facial gestures the user. The results of the evaluation suggest applicability of this approach to monitoring the user of an automatic wheelchair.

5.1 Summary of implementation

The monitoring of facial gestures in the context of this work is complicated by the fact that due to the peculiarity of the mechanical design of the automatic wheelchair, it is impossible to obtain frontal images of the face of a person sitting in the wheelchair. Using a set of ten facial gestures as a test bed application, it is demonstrated that the proposed approach is capable of robust and reliable monitoring of the facial gestures of a person sitting in a wheelchair.

The approach, presented in this work, can be summarized as follows. First, the input image which is taken by a camera, installed on the wheelchair, is processed by AAM algorithm in order to obtain the contours of the eyes and mouth of a person sitting in the wheelchair. Then, Fourier descriptors of the detected contours are calculated to obtain compact representation of the shapes of the eyes and mouth. Finally, obtained Fourier descriptors are classified to facial gestures, using the k Nearest Neighbors classifier. To reject the spurious images, the matches obtained by the AAM algorithm and results of k Nearest Neighbors classifier are analyzed and rejected if the results are likely to be spurious images.

Over the experiments conducted in this work, the system that has implemented this approach is able to recognize correctly 90% of facial gestures produced by ten volunteers. The implementation demonstrated a low false positive rate of 6% and

low false negative rate of 4%. The approach has proved to be robust to natural variations of facial gestures, produced by several volunteers as well as to variations due to inconstant camera point of view and perspective. The results suggest applicability of this approach to recognizing facial gestures in automatic wheelchair applications. However, the computational performance of the current implementation of the system is not sufficient for most of the real world applications. This drawback can be corrected, using more efficient implementations of the AAM algorithm that will be described in the next section.

5.2 Future Work

Suggestions for future work can be classified into two categories: improvements of the current work in terms of performance and accuracy, and extensions of the current work in terms of functionality.

An immediate improvement of the current work is to remove the assumption that the face of a person sitting in the wheelchair is located approximately in the same region for all images. To eliminate the assumption, future work may incorporate some type of face detector into the algorithm. The majority of known face detectors, for example, the face detector proposed by Viola and Jones [98], may be used in the algorithm. However, from a practical point of view, face detectors that do not require extensive training, such as the one proposed by Bauckhage et al.

[8], should be preferred.

The computational performance of the proposed algorithm is very important for practical applications. Currently, the algorithm does not provide real time performance and can not be used in some applications. The most important factor, contributing to the poor performance of the algorithm overall, is the performance of the AAM fitting. To improve the performance of the AAM fitting, modifications of the AAM algorithm, proposed by Matthews and Baker [71] or Xiao et al. [102], can be used. The reported performance of these modifications to the original AAM algorithm is enough to make the proposed algorithm extremely efficient in terms of computation time.

To improve the accuracy of classification results, improvement in AAM fitting and discriminative ability of Fourier descriptors may be considered.

The variability of perspective of the camera and movement of a person sitting in a wheelchair, may affect the accuracy of the AAM fitting. The main reason for such a phenomenon is that in these cases, variations of the shapes of facial features become too great to be described by a single AAM. The naive approach in dealing with this phenomenon is to increase the number of models that describe the variation. However, this approach may be not applicable for some applications due to the high computational cost. The approaches, proposed by Kanaujia and Metaxas [48] and Kanaujia et al. [47], might be suggested instead.

The classification results can also potentially improve by enhancing the representation of facial features contours. The improved contour representation describes more details of the contour and leads to better classification results. One of the simplest ways to improve the contour representation, proposed in this work, is to increase the number of points that describe the contour. However, in the case of short contours, the number of points that represents the contour is limited, and further increase in the number of points does not lead to improved contour representation. Kunttu et al. [59] proposed a technique to improve the contour representation by Fourier descriptors without increasing the number of points, representing the contour. Such an approach can also be used in this work.

To improve the robustness of the algorithm to spurious facial gestures more sophisticated classifier, e.g. Bayesian classifier [36], of spurious images may be used. To improve the performance of such a classifier, the facial gestures of the user that are not meaningful should be used for training.

The most logical extension of this work is to extend the implementation of the algorithm to detect the gaze direction of a person sitting in a wheelchair. Since the existing algorithm obtains contours of the eyes as an intermediate result, it is possible to analyze the image inside the contours, and obtain the direction of a gaze of a person, sitting in the wheelchair. Such an extension allows the system to determine where the user is looking, and use this information in various applica-

tions. One application would be to enhance the operational safety of a wheelchair by preventing movement in the direction which is currently the blind spot of the user.

Another possible extension of the work is to train the algorithm to classify human emotions, especially, happiness and distress. Such an extension provides the automatic wheelchair with valuable feedback about the reaction of the user to the actions of the wheelchair.

A Appendix I

Consent for research participation

I, the undersigned, voluntarily and without undue inducement or any element of force, deceit, or other form of constraint or coercion, consent to be a participant in the on going research of Dr. John Tsotsos.

I understand the risks in this study are not greater then those ordinarily experience in daily life, but participation in this study may be terminated at any time by my request or at the request of the investigator. Even after I give my permission, I am free to withdraw at any time without explanation.

The researcher has clearly explained the purpose and procedure of the experiment. The purpose of the experiment is to collect data for facial gesture recognition algorithm. I will make facial gestures according to instructions. The experiment will be conducted in a single session, the session will last 5-15 minutes.

I understand that responses will be gathered in such a way to insure the greatest anonymity and kept in the strictest of confidence. Any published results will either

be presented in a group format or without any form of identification.

After finishing the experiment there will be debriefing session in which I will be informed in more detail about the experimental question that the study intends to answer as well as how my results contribute to it. I will have the possibility of acquiring information about the kind of research the investigators are conducting as well as their methods and the implications of their results for the development of science.

I have been given a copy of this consent form for my records. I understand that if I have any question, I can contact the researcher at 416-7362100(ext. 3313) or Dr. John Tsotsos at 416-7362100(ext. 70135). Any concerns about the ethical aspects of the study can be addressed to the University's ethics committee (c/o Office of Research Administration, 416-736 5055).

Subject consent

I have been read the above description, it has been explained to me verbally, and my questions have been adequately addressed. I understand that as a subject in these experiments , I am free to withdraw from the experiments at any time, without penalty. I understand that if published in any form, only a letter (e.g. 'A','B', etc.) or number (e.g. '1','2', etc.) will be associated with my data.

Further I will have complete access to my results once the experiments are

completed. In other words, if I choose, the experimenter will show and explain my results for each experiment.

Subject:	Witness:
Signature:	Signature:
Telephone:	Date:
Email:	

Bibliography

[1] Facts for features: Americans with disabilities act: July 26. URL `http://www.census.gov/Press-Release/www/releases/archives/cb08ff-11.pdf`.

[2] Y. Adachi, Y. Kuno, N. Shimada, and Y. Shirai. Intelligent wheelchair using visual information on human faces. *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, 1:354–359 vol.1, Oct 1998.

[3] David W. Aha. Editorial. *Artificial Intelligence Review*, 11(1-5):7–10, 1997. ISSN 0269-2821.

[4] María Elena Algorri and Alejandra Escobar. Facial gesture recognition for interactive applications. In *ENC '04: Proceedings of the Fifth Mexican International Conference in Computer Science*, pages 188–195, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7695-2160-6.

[5] Christian Balkenius. Elastic template matching as a basis for visual landmark recognition and spatial navigation. submitted. Technical report, Department of Computer Science, Report number UMCS-97-4-1. Manchester University, 1997.

[6] R. Barea, L. Boquete, M. Mazo, and E. López. Wheelchair guidance strategies using eog. *J. Intell. Robotics Syst.*, 34(3):279–299, 2002. ISSN 0921-0296.

[7] Marian Stewart Bartlett, Gwen Littlewort, Ian Fasel, and Javier R. Movellan. Real time face detection and facial expression recognition: Development and application to human computer interaction. In *In CVPR Workshop on CVPR for HCI*, pages 139–157, 2003.

[8] Christian Bauckhage, Thomas Kaster, Andrei M. Rotenstein, and John K. Tsotsos. Fast learning for customizable head pose recognition in robotic wheelchair control. In *FGR '06: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, pages 311–316, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2503-2.

[9] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970. ISSN 00034851.

[10] Yoshua Bengio and Paolo Frasconi. An input output hmm architecture. In *Advances in Neural Information Processing Systems 7*, volume 7, pages 427–434. MIT Press, 1995.

[11] L.M. Bergasa, M. Mazo, A. Gardel, R. Barea, and L. Boquete. Commands generation by face movements applied to the guidance of a wheelchair for handicapped people. *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, 4:660–663 vol.4, 2000.

[12] L.M. Bergasa, M. Mazo, A. Gardel, J.C. Garcia, A. Ortuno, and A.E. Mendez. Guidance of a wheelchair for handicapped people by face tracking. *Emerging Technologies and Factory Automation, 1999. Proceedings. ETFA '99. 1999 7th IEEE International Conference on*, 1:105–111 vol.1, 1999.

[13] William H. Beyer. *CRC Standard Mathematical Tables*, pages 123–124. CRC Press, 28th edition, 1987.

[14] Zeungnam Bien, Myung-Jin Chung, Pyung-Hun Chang, Dong-Soo Kwon, Dae-Jin Kim, Jeong-Su Han, Jae-Hean Kim, Do-Hyung Kim, Hyung-Soon Park, Sang-Hoon Kang, Kyoobin Lee, and Soo-Chul Lim. Integration of a rehabilitation robotic system (kares ii) with human-friendly man-machine interaction units. *Auton. Robots*, 16(2):165–191, 2004. ISSN 0929-5593.

[15] N.D. Binh, E. Shuichi, and T. Ejima. Real-time hand tracking and gesture recognition system. In *ICGST International Journal on Graphics, Vision and Image Processing*, pages 31–39, 2005.

[16] Michael J. Black and Anand Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Int. J. Comput. Vision*, 19(1):57–91, 1996. ISSN 0920-5691.

[17] F. Bley, M. Rous, U. Canzler, and K.-F. Kraiss. Supervised navigation and manipulation for impaired wheelchair users. *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, 3:2790–2796 vol.3, Oct. 2004. ISSN 1062-922X.

[18] F.S. Chen, C.M. Fu, and C.L. Huang. Hand gesture recognition using a real-time tracking method and hidden markov models. *Image and Vision Computing*, 21(8):745–758, August 2003.

[19] H.Y. Chen, C.L. Huang, and C.M. Fu. Hybrid-boost learning for multi-pose face detection and facial expression recognition. *Pattern Recognition*, 41(3): 1173–1185, March 2008.

[20] Qing Chen, N.D. Georganas, and E.M. Petriu. Real-time vision-based hand gesture recognition using haar-like features. *Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE*, pages 1–6, May 2007.

[21] Thomas Coogan, George Awad, Junwei Han, and Alistair Sutherland. Real time hand gesture recognition including hand segmentation and tracking. In *International Symposium on Visual Computing*, pages 495–504, 2006.

[22] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models - their training and application. *Comput. Vis. Image Underst.*, 61(1): 38–59, January 1995. ISSN 1077-3142.

[23] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *PAMI*, 23(6):681–685, June 2001.

[24] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[25] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, January 2000.

[26] K.G. Derpanis, R.P. Wildes, and J.K. Tsotsos. Hand gesture recognition within a linguistics-based framework. In *ECCV04*, volume 1, pages 282–296. Springer, 2004.

[27] G. J. Edwards, C. J. Taylor, and T. F. Cootes. Interpreting face images using active appearance models. In *FG '98: Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, page 300, Washington, DC, USA, 1998. IEEE Computer Society. ISBN 0-8186-8344-9.

[28] P. Ekman. Methods for measuring facial action. *Handbook of Methods in Nonverbal Behavioral Research*, pages 445–90, 1982.

[29] P. Ekman and W.V. Friesen. The facial action coding system: A technique for the measurement of facial movement. In *Consulting Psychologists*, 1978.

[30] Beat Fasel and Juergen Luettin. Automatic facial expression analysis: a survey. *Pattern Recognition*, 36(1):259–275, 2003.

[31] A. Fazekas and I. Santa. Recognition of facial gestures based on support vector machines. In *IbPRIA*, page I:469, 2005.

[32] E. Fix and J.L. Hodges. Discriminatory analysis, nonparametric discrimination: Consistency properties. Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas, USA, 1951.

[33] William T. Freeman and Edward H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13: 891–906, 1991.

[34] William T. Freeman and Michael Roth. Orientation histograms for hand gesture recognition. Technical Report TR-94-03a, Mitsubishi Electric Research Laboratories, Cambridge Research Center, Massachusetts, USA, 1994.

[35] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *EuroCOLT '95: Proceedings of the Second European Conference on Computational Learning Theory*, pages 23–37, London, UK, 1995. Springer-Verlag. ISBN 3-540-59119-2.

[36] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition, Second Edition (Computer Science and Scientific Computing Series)*. Academic Press, September 1990. ISBN 0122698517.

[37] D. Gabor. Theory of communication. *Journal of the IEE*, 93:429–457, 1946.

[38] Takashi Gomi and Ann Griffith. Developing intelligent wheelchairs for the handicapped. In *Assistive Technology and Artificial Intelligence, Applications in Robotics, User Interfaces and Natural Language Processing*, pages 150–178, London, UK, 1998. Springer-Verlag. ISBN 3-540-64790-2.

[39] Colin Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(2):285–339, 1991. ISSN 00359246.

[40] Jeong-Su Han, Z. Zenn Bien, Dae-Jin Kim, Hyong-Euk Lee, and Jong-Sung Kim. Human-machine interface for wheelchair control with emg and its evaluation. *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, 2:1602–1605 Vol.2, Sept. 2003. ISSN 1094-687X.

[41] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 27:417–441, 1933.

[42] H. Hu, P. Jia, T. Lu, and K Yuan. Head gesture recognition for hands-free control of an intelligent wheelchair. *Industrial Robot: An International Journal*, 34(1):60–68, 2007. ISSN 0143-991X.

[43] M.K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, February 1962.

[44] C.L. Huang and S.H. Jeng. A model-based hand gesture recognition system. *Machine Vision and Applications*, 12(5):243–258, 2001.

[45] Jr. Joseph J. LaViola. A survey of hand posture and gesture recognition techniques and technology. Technical report, Brown University, Providence, RI, USA, 1999.

[46] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME ̆2013 Journal of Basic Engineering*, Series D(82): 35–45, 1960.

[47] A. Kanaujia, Y. Huang, and D.N. Metaxas. Emblem detections by tracking facial features. In *SLAM06*, page 108, 2006.

[48] A. Kanaujia and D.N. Metaxas. Large scale learning of active shape models. In *ICIP07*, pages I: 265–268, 2007.

[49] Seong Pal Kang, G. Rodnay, M. Tordon, and J. Katupitiya. A hand gesture based virtual interface for wheelchair control. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, volume 2, pages 778–783, 2003.

[50] Yeon Gu Kang and Phill-Kyu Rhee. Head gesture recognition using feature interpolation. In *KES (1)*, pages 582–589, 2006.

[51] Ashish Kapoor and Rosalind W. Picard. A real-time head nod and shake detector. In *PUI '01: Proceedings of the 2001 workshop on Perceptive user interfaces*, pages 1–5, New York, NY, USA, 2001. ACM.

[52] N.I. Katevas, N.M. Sgouros, S.G. Tzafestas, G. Papakonstantinou, P. Beattie, J.M. Bishop, P. Tsanakas, and D. Koutsouris. The autonomous mobile robot senario: a sensor aided intelligent navigation system for powered wheelchairs. *Robotics and Automation Magazine, IEEE*, 4(4):60–70, Dec 1997. ISSN 1070-9932.

[53] H. Kauppinen, T. Seppanen, and M. Pietikainen. An experimental comparison of autoregressive and fourier-based descriptors in 2d shape classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(2): 201–207, 1995.

[54] S. Kawato and J. Ohya. Real-time detection of nodding and head-shaking by directly detecting and tracking the "between-eyes". *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 40–45, 2000.

[55] D.H. Kim, S.U. Jung, and M.J. Chung. Extension of cascaded simple feature based face detection to facial expression recognition. *Pattern Recognition Letters*, 29(11):1621–1631, August 2008.

[56] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science, Number 4598, 13 May 1983*, 220, 4598:671–680, 1983.

[57] Yoshinori Kuno, Teruhisa Murashima, Nobutaka Shimada, and Yoshiaki Shirai. Interactive gesture interface for intelligent wheelchairs. In *IEEE International Conference on Multimedia and Expo (II)*, pages 789–792, 2000.

[58] I. Kunttu, L. Lepisto, J. Rauhamaa, and A. Visa. Multiscale fourier descriptor for shape-based image retrieval. *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 2:765–768 Vol.2, Aug. 2004. ISSN 1051-4651.

[59] I. Kunttu, L. Lepisto, and A. Visa. Enhanced fourier shape descriptor using zero-padding. In *SCIA05*, pages 892–900, 2005.

[60] S.-s. Kuo and O.E. Agazzi. Machine vision for keyword spotting using pseudo 2d hidden markov models. *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, 5:81–84 vol.5, Apr 1993.

[61] M. La Cascia, L. Valenti, and S. Sclaroff. Fully automatic, real-time detection of facial gestures from generic video. *Multimedia Signal Processing, 2004 IEEE 6th Workshop on*, pages 175–178, Sept.-1 Oct. 2004.

[62] Wei-Kai Liao and Isaac Cohen. Classifying facial gestures in presence of head motion. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, page 77, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2-3.

[63] A. Licsar and T. Sziranyi. Dynamic training of hand gesture recognition system. In *ICPR*, pages IV: 971–974, 2004.

[64] Shang-Hung Lin and S. Y. Kung. Probabilistic dbnn via expectation-maximization with multi-sensor classification applications. In *ICIP '95: Proceedings of the 1995 International Conference on Image Processing (Vol. 3)-Volume 3*, page 3236, Washington, DC, USA, 1995. IEEE Computer Society. ISBN 0-8186-7310-9.

[65] P. Lu, X.S. Huang, and Y.S. Wang. A new framework for hand-free navigation in 3d game. In *CGIV04*, 2004.

[66] P. Lu, X.S. Huang, X.S. Zhu, and Y.S. Wang. Head gesture recognition based on bayesian network. In *IbPRIA*, pages 492–499, 2005.

[67] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision (ijcai). In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 674–679, April 1981. A more complete version is available as Proceedings DARPA Image Understanding Workshop, April 1981, pp.121-130. When you refer to this work, please refer to the IJCAI paper.

[68] A. Malima, E. Ozgur, and M. Cetin. A fast algorithm for vision-based hand gesture recognition for robot control. *Signal Processing and Communications Applications, 2006 IEEE 14th*, pages 1–4, April 2006.

[69] S. Marcel, O. Bernier, J.E. Viallet, and D. Collobert. Hand gesture recognition using input-output hidden markov models. In *Automatic Face and Gesture Recognition*, pages 456–461, 2000.

[70] Y. Matsumoto, T. Ino, and T. Ogasawara. Development of intelligent wheelchair system with face and gaze based interface. *Robot and Human Interactive Communication, 2001. Proceedings. 10th IEEE International Workshop on*, pages 262–267, 2001.

[71] Iain Matthews and Simon Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60:2004, 2004.

[72] Babu M. Mehtre, Mohan S. Kankanhalli, and Wing F. Lee. Shape measures for content based image retrieval: A comparison. *Information Processing & Management*, 33(3):319–337, May 1997.

[73] S. Mitra and T. Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 37(3):311–324, 2007.

[74] Inhyuk Moon, Myungjoon Lee, Jeicheong Ryu, and Museong Mun. Intelligent robotic wheelchair with emg-, gesture-, and voice-based interfaces. *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, 4:3453–3458 vol.3, Oct. 2003.

[75] S. Nakanishi, Y. Kuno, N. Shimada, and Y. Shirai. Robotic wheelchair based on observations of both user and environment. *Intelligent Robots and Systems, 1999. IROS '99. Proceedings. 1999 IEEE/RSJ International Conference on*, 2:912–917 vol.2, 1999.

[76] Mi Young Nam and Phill-Kyu Rhee. An efficient face location using integrated feature space. In *KES (2)*, pages 327–335, 2005.

[77] Pei Chi Ng and L.C. De Silva. Head gestures recognition. *Image Processing, 2001. Proceedings. 2001 International Conference on*, 3:266–269 vol.3, 2001.

[78] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1):51–59, January 1996.

[79] M.A. Okkonen, V. Kellokumpu, M. Pietikainen, and J. Heikkila. A visual system for hand gesture recognition in human-computer interaction. In *Scandinavian Conference on Image Analysis*, pages 709–718, 2007.

[80] OpenCV. Opencv library, 2006. URL `http://www.intel.com/technology/computing/opencv/`.

[81] Maja Pantic, Student Member, and Leon J. M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1424–1445, 2000.

[82] Vladimir I. Pavlovic, Rajeev Sharma, and Thomas S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):677–695, 1997. ISSN 0162-8828.

[83] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann, September 1988. ISBN 1558604790.

[84] Y. Satoh and K. Sakaue. An omnidirectional stereo vision-based smart wheelchair. *JIVP*, 2007, 2007.

[85] B. Scholkopf, A. Smola, R. Williamson, and P. Bartlett. New support vector algorithms. *Neural Computation*, 12:1207–1245, 2000.

[86] C. Shan and T. Gritti. Learning discriminative lbp-histogram bins for facial expression recognition. In *BMVC08*, pages xx–yy, 2008.

[87] Jianbo Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 593–600, 1994.

[88] R. C. Simpson. Smart wheelchairs: A literature review. *Journal of Rehabilitation Research and Development*, 42(4):423–436, 2005. ISSN 0748-7711.

[89] A.S.M. Sohail and P. Bhattacharya. Classification of facial expressions using k-nearest neighbor classifier. In *MIRAGE07*, pages 555–566, 2007.

[90] M. B. Stegmann. Active appearance models: Theory, extensions and cases. Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, aug 2000. URL `http://www.imm.dtu.dk/~aam/main/`.

[91] H. Steinhaus. Sur la division des corp materiels en parties. *Bulletin L'Acadmie Polonaise des Science C1. III*, IV:801–804, 1956.

[92] W.C. Stokoe, D. Casterline, and C.C. Croneberg. *A Dictionary of American Sign Language*. Linstock Press, Washington, DC, USA, 1965.

[93] K. Tanaka, K. Matsunaga, and H.O. Wang. Electroencephalogram-based control of an electric wheelchair. *Robotics, IEEE Transactions on*, 21(4): 762–766, Aug. 2005. ISSN 1552-3098.

[94] Jinshan Tang and Ryohei Nakatsu. A head gesture recognition algorithm. In *ICMI '00: Proceedings of the Third International Conference on Advances in Multimodal Interfaces*, pages 72–80, London, UK, 2000. Springer-Verlag. ISBN 3-540-41180-1.

[95] C.J. Taylor, G.J. Edwards, and T.F. Cootes. Active appearance models. In *ECCV98*, volume 2, pages 484–498, 1998.

[96] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report 132, CMU, Carnegie Mellon University, April 1991.

[97] P. Viola and M. Jones. 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 I–511–I–518 vol.1, 2001.

[98] P.A. Viola and M.J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, May 2004.

[99] Jian-Gang Wang and Eric Sung. Gaze determination via images of irises. *Image Vision Computing*, 19(12):891–911, 2001.

[100] Jian-Gang Wang, Eric Sung, and Ronda Venkateswarlu. Eye gaze estimation from a single image of one eye. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 136, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1950-4.

[101] Ying Wu and Thomas S. Huang. Vision-based gesture recognition: A review. In *GW '99: Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction*, pages 103–115, London, UK, 1999. Springer-Verlag. ISBN 3-540-66935-3.

[102] Jing Xiao, Simon Baker, Iain Matthews, and Takeo Kanade. Real-time combined 2d+3d active appearance models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 535 – 542, June 2004.

[103] Holly A. Yanco. Integrating robotic research: a survey of robotic wheelchair development. In *AAAI Spring Symposium on Integrating Robotic Research*, 1998.

[104] Ikushi Yoda, Katsuhiko Sakaue, and Takenobu Inoue. Development of head gesture interface for electric wheelchair. In *i-CREATe '07: Proceedings of the 1st international convention on Rehabilitation engineering & assistive technology*, pages 77–80, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-852-7.

[105] Ikushi Yoda, Junichi Tanaka, Bisser Raytchev, Katsuhiko Sakaue, and Takenobu Inoue. Stereo camera based non-contact non-constraining head gesture interface for electric wheelchairs. *icpr*, 4:740–745, 2006. ISSN 1051-4651.

[106] H.S. Yoon, J. Soh, Y.L.J. Bae, and H.S. Yang. Hand gesture recognition using combined features of location, angle and velocity. *Pattern Recognition*, 34(7):1491–1501, July 2001.

[107] C.T. Zahn and R.Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Trans. Computers*, 21(3):269–281, March 1972.

[108] Y.Z. Zhan, J.F. Ye, D.J. Niu, and P. Cao. Facial expression recognition based on gabor wavelet transformation and elastic templates matching. In *ICIG '04: Proceedings of the Third International Conference on Image and Graphics*, pages 254–257, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7695-2244-0.

[109] D. S. Zhang and G. Lu. A comparative study of fourier descriptors for shape representation and retrieval. In *Proceedings of the Fifth Asian Conference on Computer Vision*, pages 646–651, 2002.

[110] Dengsheng Zhang and Guojun Lu. A comparative study of curvature scale space and fourier descriptors for shape-based image retrieval. *J. Visual Communication and Image Representation*, 14(1):39–57, 2003.