TÉCNICO LISBOA

Clustering of movement profiles of motor-impaired people from 2D image streams

Mário André Esteves Macedo

Thesis to obtain the Master of Science Degree in

Biomedical Engineering

Supervisor(s): Dr. Manuel Ricardo de Almeida Rodrigues Marques
Prof. João Paulo Salgado Arriscado Costeira

Examination Committee

Chairperson: Prof. João Miguel Raposo Sanches
Supervisor: Dr. Manuel Ricardo De Almeida Rodrigues Marques
Member of the Committee: Prof. Duarte Nuno Jardim Nunes
Prof. Pedro Manuel Quintas Aguiar

November 2018

Declaração

Declaro que o presente documento é um trabalho original da minha autoria e que cumpre todos os requisitos do Código de Conduta e Boas Práticas da Universidade de Lisboa.

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

E assim fica concluída uma grande etapa da minha vida. No entanto, nunca teria conseguido cá chegar se não fosse toda a ajuda que me tem sido dada ao longo destes anos.

Em primeiro lugar, um grande agradecimento aos meus orientadores, Dr. Manuel Marques e Prof. João Paulo Costeira, sem eles nunca teria chegado tão longe com esta tese. Obrigado por toda a ajuda que me deram ao longo destes meses, por todas as ideias, ensinamentos e acima de tudo por se preocuparem tanto com os vossos alunos. Mais uma vez peço desculpa por ter chateado tanto!!

Gostaria também de agradecer a todos os meus amigos, em especial aos que já vêm desde o Secundário (desculpem pelo bullying :'( ), aos que enfrentaram o curso comigo ao longo destes 5 anos e também aos membros do SIPG. Estarei para sempre grato por todo o apoio dado ao longo deste percurso.

Um grande agradecimento à minha família, por apesar de todas as minhas falhas estarem sempre prontos a ajudar e quererem sempre o (que pensam ser :P) melhor para mim! Em especial, obrigado à minha mãe, que sempre se esforçou por me incutir bons valores e por me dar o melhor.

Finalmente, um grande agradecimento a ti, Sofia, que para o bem ou para o mal, ao longo destes anos estiveste sempre lá para mim! Ajudaste-me a crescer bastante quer academicamente, quer pessoalmente (mas olha que ainda há bastante espaço para improvement, por isso a ver se continuas a fazer um excelente trabalho)! És uma pessoa incrível, e só mesmo a tua teimosia consegue chegar ao nível de todas as tuas qualidades! :P Após esta longa e cansativa fase espero que o futuro te sorria, que bem mereces, e continues sempre com paciência para me aturar! ;)

Abstract

Ageing, motor disorders and accidents are some possible causes that lead people to require help from caregivers, being impossible for patients to have autonomy. Assistive robots have been developed to help them in tasks such as feeding. However, as each patient is a different case, so it is imperative to classify them and adapt the technology to better serve their needs.

The aim of this work was a methodology which focused on classifying different subjects based on the manifestation of their disabilities in a specific setup. This would enable a better adaptation of robotic systems to adapt to each patient and consequently improve their performance. To achieve this result, a data set was created with recordings of people with and without physical limitations, while performing given tasks.

Data from specific points was acquired from these videos using a facial landmark detector. Nevertheless, this data was incomplete, and so a matrix completion procedure with a Structure from Motion framework was used. Afterwards, the trajectories of the subjects were encoded onto two different features. These features were then used to build a classifier, based on the Bag of Words model, aiming to discern the different types of subjects depending on their performance.

During the clustering phase, it was possible to identify specific patterns associated to specific type of subjects. The results obtained were quite promising as they proved that the feasibility this study.

Keywords Movement Classification, Structure from Motion, Bag of Words, Matrix completion, Trajectory Feature

Resumo

Envelhecimento, doenças motoras e acidentes são algumas das causas que levam à necessidade de ajuda de cuidadores, sendo impossível os pacientes terem autonomia. A robótica assistiva foi desenvolvida para ajudar estes indivíduos em tarefas como a alimentação. Contudo, cada paciente é um caso, daí que seja necessário classificá-los de modo a adaptar a tecnologia para os conseguir servir melhor.

Nesta tese focamo-nos na criação de uma metodologia cujo objetivo é a classificação de diferentes indivíduos consoante as suas dificuldades. Com esta classificação seria possível adaptar os sistemas robóticos e consequentemente servir melhor os sujeitos avaliados. Para criar este classificador foi necessário coleccionar um conjunto de dados constituído por filmagens de diferentes sujeitos, com e sem dificuldades motoras, a realizarem um determinado conjunto de movimentos.

De seguida foram recolhidas as coordenadas de pontos específicos do rosto de cada indivíduo, a partir da utilização de um detetor já existente. No entanto, os dados obtidos a partir desta ferramenta vinham incompletos, daí que tenha sido necessário usar um modelo baseado em Estrutura a partir do Movimento para completar os mesmos. Seguidamente, as trajetórias percorridas pelos pontos da face dos participantes foram codificadas em dois tipos de descritores. Estes foram depois usados para construir um classificador baseado em Bag of Words, cujo objetivo consistia em identificar os diferentes tipos de indivíduos dependendo da execução dos movimentos.

Durante a fase de agrupamento dos dados, já era possível identificar padrões específicos de determinados tipos de sujeitos. Os resultados obtidos foram bastante promissores, pois provaram a viabilidade deste estudo.

Palavras-chave: Classificação de Movimento, Estrutura a partir do Movimento, Bag of Words, Estimação de dados omissos, Descritor de Trajectórias

Contents

List of Tables

List of Figures

Nomenclature

Greek symbols

α	Image Scale Factor
β	Scale factor
ϵ	Infinitesimal
Λ	Matrix with Images' Scale Factors
Ω	Stiefel Matrix
Σ	Diagonal Matrix with Singular Values

Math Symbols

$\circ$	Hadamard Product or Element-Wise Product
$\mathbb{I}$	Identity Matrix
$\mathbb{1}$	Matrix filled with 1
$\oslash$	Element-Wise division

Roman symbols

D	Diagonal Matrix
F	Number of Frames
M	Motion Matrix
P	Number of Points
R	Rotational Matrix
S	Shape Matrix
U	Left Singular Vectors
u, v	Pixel coordinates in image
V	Right Singular Vectors

W Data Matrix

Subscripts

c Centred

i, j, k Computational indexes

$norm$ Normalized

ref Reference

x, y, z Cartesian components

Superscripts

-1 Inverse

T Transpose

Chapter 1

Introduction

1.1 Motivation

> "Disabilities is an umbrella term, covering impairments, activity limitations, and participation restrictions. An impairment is a problem in body function or structure; an activity limitation is a difficulty encountered by an individual in executing a task or action; while a participation restriction is a problem experienced by an individual in involvement in life situations."
>
> – World Health Organization, Disabilities [1]

Nowadays, the number of people with disabilities is growing, mainly due to the ageing of the population. Thus, as people get older, the risk of having a chronic disease increases accordingly, which is commonly associated with disabilities [2].

According to the Global Burden of Disease, 3.8% of the population older than 15 years old have very significant severe disabilities corresponding to approximately 190 million people. In what concerns the younger population (up to 14 years old), it is estimated that 0.7% has severe disabilities, which corresponds to about 13 million children [2].

About 30% of this young population is affected by Cerebral Palsy [3], a disorder that is associated with damage to parts of the brain that control movement, coordination, balance and posture. This type of damage leads to symptoms such as abnormalities of muscle tone, which consequently impedes the desired movement [4].

Nevertheless, different people are affected differently by this disorder. There can be changes both in terms of locations of the body affected or the degree of the control over that region. Therefore, classifying these people is a complex task and there is still no procedure and definition to properly characterise the disease.

On the one hand, there is a topological classification, that analyses which areas of the body are affected [5]. On the other hand, there are functional classifications, which evaluate how these people can move in different scenarios. In this project, the latter classification will be focused on.

Currently, there are several types of functional classification systems for individuals with cerebral

palsy. Firstly, the Gross Motor Function Classification System, that aims to provide a standardised measurement of severity of the disorder. This system consist of a set of levels, based on how these patients performed in different tasks. There are a total of 5 levels, and these focus mainly on the locomotion capacities of the patients [6].

Secondly, there is the Manual Ability Classification System, whose aim is to classify how children with this disorder use their hands with objects during activities of daily living [7].

Thirdly, there is the Eating and Drinking Ability Classification System for Individuals with Cerebral Palsy, which analyses the safety and efficiency while these subjects are eating [8].

Overall, these classifiers consist of a set of tasks which can either be achieved or not, and based on the performance of the patient, they would be classified accordingly. However,there is a common drawback to all of these systems, which is the high subjectivity of such an analysis. So, under the same circumstances, different people might classify the same person differently.

Such classification is of extreme importance as they influence politically sensitive topics, such as helping decide if particular individuals are eligible for social insurance programs [9]. Since this classification has such great importance, then it should be properly standardised, so as to avoid subjectivity as much as possible.

In this thesis we tackle the problem of attaining objectivity, by the means of machine learning techniques. More precisely, an automatic classifier was developed in this project, aiming to differentiate the different patients based on their physical limitations.

1.1.1 Feeding Robot

The scenario behind this classification is related to assistive robotics, more precisely, using a robotic arm to help feed people with this type of disorder. Feeding, as well as bathing or dressing are defined as Activities of Daily Living [10] and are of utmost importance, as they are self-care tasks which allow an individual to properly live independently in a community.

Currently, some robotic arms have been developed with the objective of feeding people with upper arm disabilities. This would allow these people to gain some autonomy over their lives as they no longer needed the assistance of a caregiver during the meal, thus improving their quality of life.

The first fully capable robot arm to help motor-impaired people was called Handy1 [11]. This arm allowed the user to choose between 7 different food types using buttons. As the food is chosen, Handy1 would move from the plate to the mouth location, previously defined. This robotic system has some drawbacks, since it needs direct input from the user by the means of buttons, which may not be possible to everybody with cerebral palsy. Additionally, the movement is predefined, so it is not possible to adapt to each user.

Another robotic system, developed for commercial purposes, was Obi [12]. One of the major differences implemented on this one, consists on the possibility to adapt the end-effector trajectory to each user through a calibration step. Even though the trajectory is improved, there is still no real-time trajectory planning from the plate to the user's mouth.

In order to provide a real-time trajectory planning, some studies using vision and EEG have been developed [13]. In this new robot arm, the user could choose the type of food using EEG signals and based in mouth detection, acquired from the RGB output, the robot arm could track the user's mouth. Moreover, an open/closed mouth classifier was also used, so as to control the beginning of the arm's movement. The use of EEG signals to control the robot was very useful for users that could not control a joystick and/or keyboard buttons. However, using such technique required extensive training and a device would need to be attached to the user's head to measure the signals.

In summary, there is still no system that enables people with this disorder to be properly fed without any help. For instance, when the trajectory of the end-effector is predefined, the type of person is not taken into account. So, for people who can not move at all, the robotic arm would need to move the spoon all the way to the inside of the user's mouth. Whereas, if the user was able to lean the upper-body forward, then it would be preferable to leave some distance between the end-effector and the mouth of the user, so that they could reach the spoon whenever they desired.

In this thesis it was developed an automatic classifier which aims to differentiate motor impaired subjects based on their physical limitations. For instance, the proposed classifier would be able to classify the person which was using the robotic arm. After classifying the user, the robotic arm could adapt its type of movement to the specific user. Therefore, the trajectories used by the robot would take into consideration the limitations of the user and move accordingly.

In order to achieve this goal, we analyse the recordings of different people while doing some movements. Then, based on their motion patterns we were able to predict which type of user we were dealing with. This classifier does not need any type of information manually provided, all the required information is encoded on the movie.

1.2 Contributions

Overall this thesis has contributed with:

Automatic Classification for subjects with disabilities

A new approach to classify subjects according to their physical disabilities is proposed. The trajectories of facial landmarks were detected along the movie. Afterwards, based on the trajectories captured it was possible to compute the intended features. By classifying these features, one could estimate the degree of the subject's physical limitations.

Track completion

A procedure was developed in order to track the coordinates of the facial features by solving a problem of matrix completion with high-rate of missing data. Overall, this methodology consisted on assuming that the face was a rigid body and applying the Tomasi-Kanade factorization algorithm to estimate part of the missing entries. Finally, the Lucas-Kanade algorithm was used in order to track the remaining points.

Trajectory descriptors

In this thesis we propose two different ways to describe trajectories in the image. These were developed during this thesis and one of them emphasizes on the spatial properties and the other on the temporal ones.

Creation of a new Data Set

A new data set with recordings of different subjects while performing different tasks has been created.

1.3 Thesis Outline

This thesis is composed of seven chapters. The current chapter introduces the problem which we will tackle along this thesis, the classification of subjects with disabilities. An overview of our scenario is done, by describing current development of the feeding systems. Finally, the contributions of this thesis are enumerated.

Chapter 2 provides an overview of state of the art methods, related with our work. More precisely, topics such as human motion recognition, trajectory classification point detection and tracking are further discussed.

In Chapter 3, the data set developed during this thesis is described. Details regarding the setup, subjects analysed and movement performed by them are explained.

Regarding Chapter 4, an in depth explanation of how the data was acquired from the recordings in the data set and then processed is provided.

Later, in Chapter 5 the data previously processed is normalized and then encoded into the trajectons described here in further detail. After explaining this important part, we proceed to explaining how these descriptors were classified.

The results obtained are shown and discussed in Chapter 6. This one analysis the different steps in our thesis, from obtaining our data until the classification of the descriptors.

Finally, Chapter 7 provides the main thesis conclusions by giving an overview of the work done throughout the thesis. In the end, some future recommendations are proposed in order to further improve the classifier.

Chapter 2

State of the art

In this thesis, we propose to classify different subjects according to their movement ability from 2D videos.

Some works have already studied human motion recognition. However, they usually focus mostly on movement of the limbs, such as gait analysis, or the control over the upper-extremities. To the best of our knowledge, no work has been done to classify subjects and the degree of their physical limitations based on their performance in given tasks associated with a feeding scenario.

In order to classify the subjects, we used the trajectories of their facial landmarks while they were performing the given tasks. Trajectory classification is an important research topic, that has has been studied for a long time and impacts several fields, such as surveillance and traffic control.

These trajectories were obtained by detecting the coordinates of specific points over the frames. In order to acquired these points, a facial landmark detector had to be used. This detector would predict the coordinates of the points along the recordings captured.

2.1 Human motion recognition

Currently, the work regarding human motion recognition has mostly focused on discerning different types of movement, such as walking and running.

To perform this type of classification, there are works which extract the basic movement of the subject and represent it as a simple 2D human stick figure, as shown in Fig.2.1. Later, the features are extracted from the 2D model and a predictive modular neural network time series classification algorithm is applied to label the movement under analysis [14].

Other methods transform the space-time human silhouettes onto low-dimensional multivariate time series. Afterwards, they use this time series to extract features that summarise the motion properties and, finally, Gaussian Processes classification is used to learn and predict motion categories [15].

Regarding individuals with disabilities, the type of classification proposed was not associated with estimating the degree of disability, but instead with classifying types of specific motions in order to allow the operation of instruments such as wheelchairs. More precisely, it was suggested an image

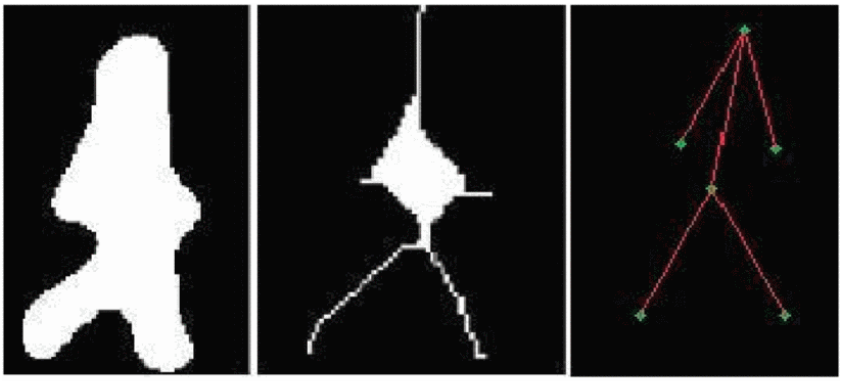


Figure 2.1: Representation of 2D human stick figure [14].

classification procedure for hand gesture classification using Artificial Neural Networks. Therefore, the wheelchair would be controlled based on the pose of the hand, which was detected by the camera [16].

The idea of operating an electric wheelchair by human body motion interface was also employed in another work, where a pressure sensor would detect the distribution of weight of the person. Based on this measure and through a Self-Organising Map, it would be possible to estimate the pose of the user [17].

2.2 Trajectory classification

In this work, we detected the points on the face of the subjects over the frames. Then, based on their position, we were able to extract their trajectory and use it to classify each one of them.

Trajectory classification has been an active research topics in several areas, such as surveillance and traffic control. However, there is no standard way to solve this problem as each technique is more suitable for each type of setup.

Most proposed methods for trajectory classification use hidden Markov models [18]. For instance, in the case of classifying human motion trajectories, one of the proposed methods relied on segmenting trajectories at points of change in curvature and representing them by their principal component analysis coefficients. Then, hidden Markov models were used so as to classify them, based on this coefficients, while capturing the temporal relationship between them [19].

Another commonly used methodology was based on neural networks to classify this type of data. Some of them have already been mentioned in the previous section. In those methods, instead of using all the trajectory, they were actually encoded into a feature vector using only part of the information obtained from each frame, as can be seen in Fig.2.3. Then, a neural network was used to classify the series of features and thus predict which type of movement was present on the film among a predefined set of motions [20].

Moreover, a Self-Organising Map has also been used in a surveillance scenario. In this case, the neural network would learn the characteristics of normal trajectories, such as the one in Fig.2.3(a), and become able to detect new ones, Fig.2.3(a), which would be considered suspicious [21].

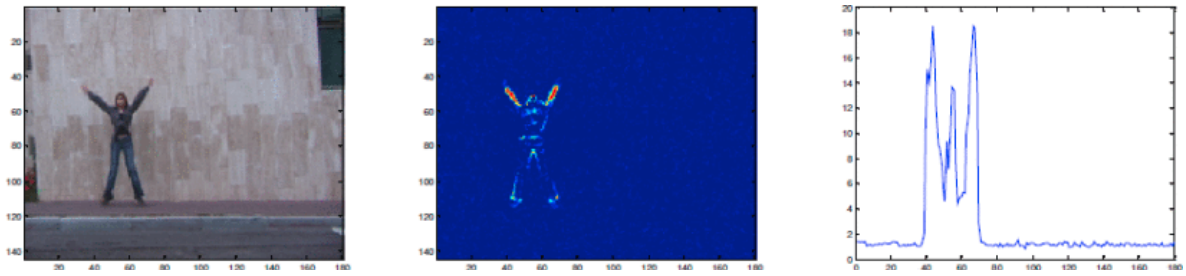


Figure 2.2: Process of capturing a feature from part of the information obtained from each frame, by [20].

Figure 2.3: Trajectories captured in the surveillance [21].

Support Vector Machines (SVM) is another popular method used to classify trajectories. In some works, the method proposed relied on extracting important features from trajectories, such as velocity, acceleration, orientation and classifying them with this algorithm [22, 23]. Moreover, similar to the surveillance case, a SVM classifier had been proposed which detected anomalies on traffic videos. In this case, the normal trajectories were used as the training data, and then the trajectories that did not fit on the trained model were detected [24].

Time-series classification can also be used to classify different trajectories. However, we are dealing with trajectories over the several frames and not exactly scalar values over the movie, so it is necessary to adjust the data and the models to fit each other [18]. One of the algorithms used to measure similarity between these features is the dynamic time warping. Then, based on the distance calculated between these time series it is possible to classify them based on their similarity [25].

2.2.1 Bag of Words

Another possible method is the Bag of Words (BoW) model, which will now be explained in detail which our thesis focus more. The BoW model is a method commonly used in language processing and computer vision to classify documents and images, respectively [26].

In the language processing scenario we have a text which is our bag, and this text is filled with words which represent the features. Each word is counted and so an histogram can be created in which the several entries correspond to the words and their respective value is associated with the number of occurrences. Then, with the help of a data set with several documents properly labelled, one can train a classifier to detect the type of future unknown documents that we might need to label.

In computer vision problems, this model is applied in a slightly different manner, the main reason being the fact that in images we do not actually have words. Therefore, another way to represent these images is required. In order to achieve this, it is usually necessary to detect some sort of feature. This is of extreme importance, as this feature will be the basis to discern the different images. Therefore, the features selected should be in some way associated with the type of classification that is under study [27].

After defining these features, it is necessary to differentiate each one of them into several classes. This is done by grouping together similar features, for instance through k-means clustering. By doing this we are creating the “codewords” which are the equivalent of the words in the language processing problem. It is important to mention that these “codewords” should be in some way countable. In the end, after analysing the different objects, we have a “codebook”, which is the analogous to a word dictionary, in which the different types of features are saved. This countable information can then be used to create to create a histogram and either build a classifier or simply classify the new images.

Some methods which also explore this idea have already been proposed. One of them analyses the trajectory of a single point and builds a “bag of segments” to store the different types of movement. Then, trajectory analysis can be decomposed by these segments and classified accordingly [28]. Similar to this idea, another method has been proposed, but now with a higher number of points. In this case, a cluster-based dictionary is constructed, based on the shape of partitions of the trajectory [29].

2.3 Point detection

In order to define the trajectories, it was necessary to identify key points of the face along the movie.

In order to detect the face of the subject, and even more important the position of these facial features, one of several algorithms could have been used.

Initially, in 2001, Paul Viola and Michael Jones proposed the first competitive real-time face tracker [30]. This detector was quite accurate and the processing time taken was very low. However, this had some setbacks, such as the fact that it was only a face detector, and so it was not able to predict the position of the different facial features. Additionally, this detector could not cope with different poses, the subjects had to be facing forward in order to be properly identified. In Fig.2.4, one can see the output after applying the face detector.

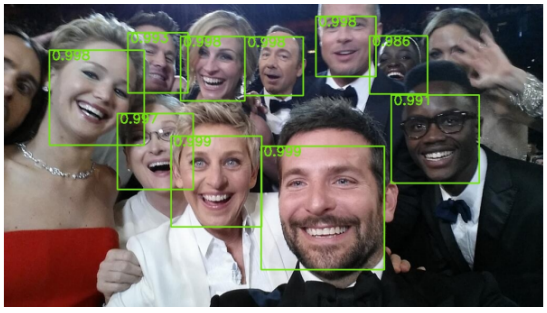


Figure 2.4: Example of the output provided by the Viola-Jones face detector [31].

Meanwhile several other face detectors started to be developed and so, currently, there are many applications which allow us to get the position of the facial features, among which the Multi-task Cascaded Convolutional Networks (MTCNN), OpenFace, OpenPose.

In the case of MTCNN, several neural networks are used in order to build a three-stage cascaded framework [32]. In this cascade first the candidate windows and their bounding boxes are estimated. Then, these candidates are filtered, by merging the similar ones and selecting the more likely ones. Finally, five facial landmarks' positions are detected inside the bounding boxes previously estimated. These five points correspond to specific points in the face of the subject, specifically, the eyes, nose and corners of the mouth, as can be seen in Fig.2.5.

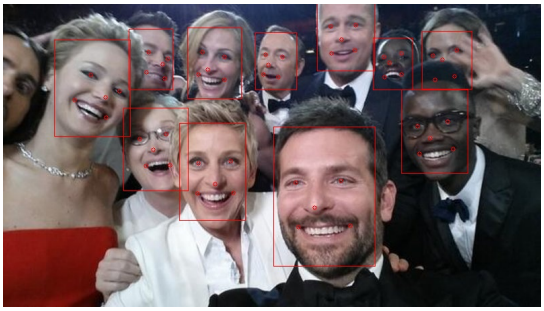


Figure 2.5: Example of the output provided by the MTCNN detector.

Recently proposed, OpenFace [33], an open source facial feature detector gained prominence. In order to predict the different facial features, this algorithm uses Conditional Local Neural Fields, which is an instance of a Constrained Local Model, for facial landmark detection and tracking. The Constrained Local Model predicts the appearance of each facial landmark individually using local detectors and uses a shape model to perform constrained optimization and, consequently, better estimate each point based on the other ones. This new algorithm has two features, firstly it uses Point Distribution Model which captures landmark shape variations, and secondly it uses patch experts which capture local appearance variations of each landmark [33–35]. With this new algorithm we can then have more facial features, namely up to 68 landmarks as shown in Fig2.6, with even higher accuracy compared to the previous one.

Finally, regarding OpenPose, this algorithm not only estimates head pose and facial landmarks, but also the position of joints of the human body. In this algorithm both the body pose estimation and the facial features detector are independent. In the case of the body detector it is able to predict the common joints in the limbs, but it also detects the nose and the ears, which is more relevant to our work. This detector is a bottom-up representation that starts by identifying the location of the different joints, then they are grouped into limbs via Part Affinity Fields and finally, these parts are associated forming, finally, the subject [37, 38]. Regarding the face detector, several cameras were used in order to apply multiview bootstrapping. This procedure consisted on using some keypoints detectors to produce noisy labels from different perspectives. Then, the labelled points were triangulated and the labels corrected on each iteration [39].

As we are interested in knowing the points from the head of the subjects, this detector would be

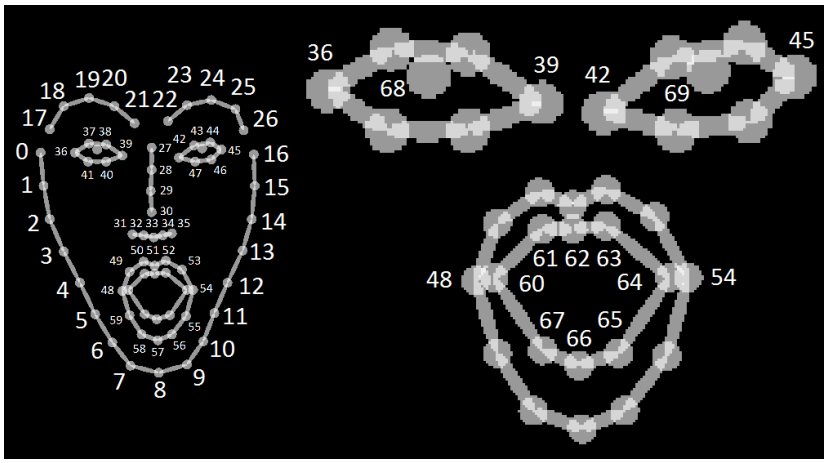


Figure 2.6: Points captured by OpenFace [36].

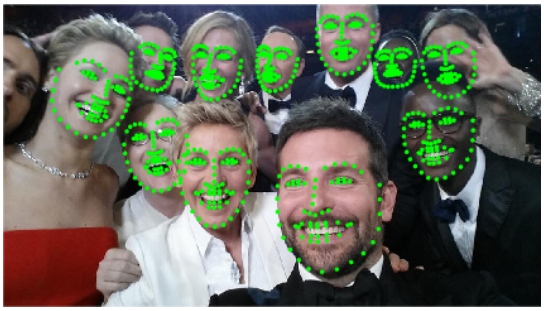


Figure 2.7: Example of the output provided by the OpenFace detector.

able to provide us with the same 68 landmarks as OpenFace, plus another 3 provided by the body estimation, as can be seen in Fig.2.8. Therefore, this one provides more data to analyse, but in contrast, is computationally more demanding and requires GPU acceleration to achieve real-time performance [40].

Taking all these different facial landmark detectors into account the one chosen to be used during this project was the OpenPose. This algorithm was chosen as it was the one that better represented the 3D structure, due to the addition of the Body Points. Moreover, this algorithm also showed each the degree of confidence of each point, which would later be used to filter the wrongly assigned points.

However, not all the points provided by the detector were used, only 19. More precisely, the points used were the nose (point number 0), eyes (14, 15), and ears (16, 17), from the body output format which can be seen in Fig.2.8. Additionally, some points of the face output format were also used, namely the corners of the mouth (48, 54), corners of the nose (27, 30, 31, 35) and outer corners of the eyes and eyebrows (17, 21, 22, 26). This format is represented in Fig.2.6.

Another important aspect of the selected points is the fact that they can express reasonably well the orientation of the face while preserving an important property, namely rigidity. Choosing other points would either introduce more non-rigidity to our model, for instance points in the neck or chin; or would usually not be detected due to the setup, such as the points in the jaw.

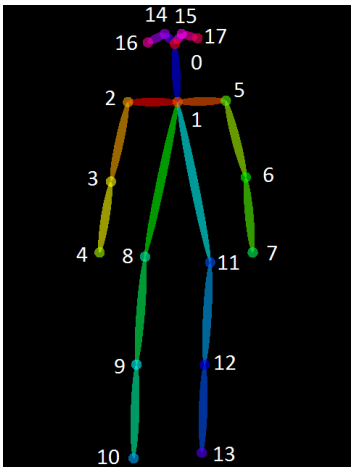


Figure 2.8: Points on the body captured by OpenPose [36].

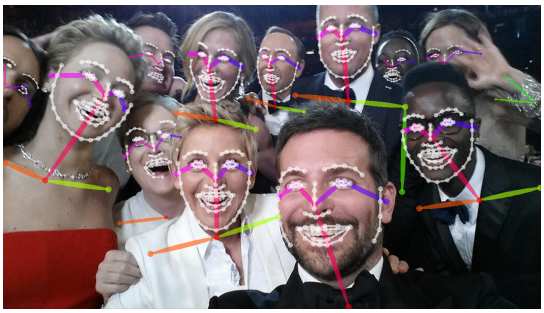


Figure 2.9: Example of the output provided by the OpenPose detector.

2.4 Point tracking

Another important topic in this project is feature tracking throughout the different frames. This problem is referred to as optical flow and, even though it has been studied for many years, it is still an unresolved problem due to its complexity. Currently this is mostly solved by the Lucas-Kanade pyramidal algorithm. This method was developed by Bruce D. Lucas and Takeo Kanade and is mainly used for optical flow estimation [41]. It consists on combining the luminosity information from several pixels nearby our point of interest, and trying to detect the new coordinates of the pattern previously identified. In other words, the point under consideration is tracked by its neighbourhood centred at the point of interest and by solving basic optical flow equations we can predict the new position of the lost point.

This method allows to track the position of specific points throughout the sequence, as long as certain requirements are met. Firstly, the points have be visible, otherwise the pattern around the point of interest is associated with a different area unrelated to it. Secondly, the point can not move too far away from its previous position. The reason behind this relies on the fact that this method searches for the points in the neighbouring pixels. Therefore, in the case of fast movement, the point we want to track will be too far away of the area where we are actually searching for the points. Finally, the contrast between the keypoints and their surrounding needs to be high so as to have a better understanding of the points' surroundings and thus better find them as they move between different frames.

Chapter 3

Data Set

In this thesis, we propose to classify different subjects according to their movement ability from 2D videos. This classification is associated with each type of movement independently, since people with disabilities behave differently depending on the task requested. Depending on the movement the subjects might classified as a person with or without disabilities. However, in order to build our classifier we needed a data set to acquire the trajectories of the subjects while performing the given tasks. From these trajectories, later, we would be able to encode it into two features and use them to differentiate the moving patterns.

In this chapter we described how the data set was obtained, more precisely, the different conditions under which the subjects were exposed while recording their movements. Additionally, the different movements that were recorded are also described in detail. Moreover, a brief description of the different subjects who accepted to participate in this project is also provided.

3.1 Data model

In order to acquire our data, ideally all the subjects should be under the same conditions. This standardization would allow to have better initial data, as we would not be so dependent on the configuration of the room, or the position of the camera relatively to the subject. For example, if the camera was placed slightly higher for a specific subject, as the data was acquired it would be as if instead of looking forward, they were looking downwards.

Overall, the environment used for recording our movie consisted on the subject sitting on a chair, with a table in front of him. This table would have some objects properly spaced between them. These corresponded to our target, in other words, what the subjects would try to reach while performing their movements. Additionally, we also had two cameras in front of the subject so as to record their movements.

We also measured the height of the chair, table and cameras, as well as the distance between these 3 major components. Finally, the distances between the objects on the table were also registered. These measurements, as well as their configuration, can be seen in Fig.3.1.

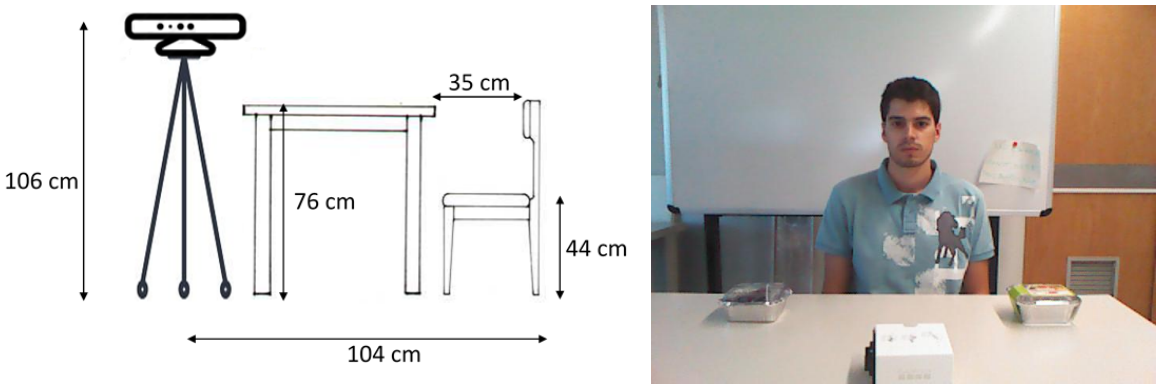


Figure 3.1: Distances in the setup (left) and output of the camera (right)

In order to record the movement of the subjects, a Kinect 360 was used, presented in Fig.3.2.

Figure 3.2: Camera used in this project - Kinect

This work mainly focused on developing a method that would work on the Kinect camera, since this camera had poor resolution. The major drawback of using a low resolution camera was associated with the difficulty behind accurately estimating the trajectories of the subjects when performing the required movements. Nevertheless, by developing a method that would work in this scenario, where the data was highly corrupted, it led to the creation of the implementation of a robust solution that works on any RGB camera.

3.2 Subjects

The subjects analysed during this project were composed by different people both with and without disabilities. In order to acquire data from different subjects with some sort of disabilities, we went to Centro de Reabilitação de Paralisia Cerebral Calouste Gulbenkian, managed by Santa Casa da Misericórdia, where it was possible to record 9 subjects. The people from this institution, due to their different disabilities, had their maneuverability affected with varying levels of severity. Another part of the recordings was done in Instituto Superior Técnico with some of their students. These last 13 subjects did not have any kind of disability that would compromise the movements asked to perform during the recordings.

All these subjects had to sign an informed consent form, in accordance to the World Health Organization.

All subjects were divided into 3 categories so as to simplify the classification process. This division depended on the degree of maneuverability. In the first group we had the subjects which were able to

perform all movements easily without any involuntary movements. In the second group, there were the ones who were able to perform the movement with some difficulty, with the existence of some involuntary movement or low movement amplitude. Finally, the last groups consisted of the subjects who had really high difficulties to perform the movement, or that couldn't achieve it at all.

3.3 Types of Movement

As noted, there are different types of movement. These movements allowed the comparison and classification of the subjects. Since every subject was doing the same type of movements, it was possible to segment the video for every subject, as it will be later explained in section 3.4. By doing this, as we analyse each movement, we were in fact comparing the different ways to perform each specific movement.

Besides standardizing our data set, by defining these movements we would also improve our data, since the selection of the movements took into account the constraints of the methodology of this work. Not only it would prevent undesired motions, but also stimulate the appearance of new types of orientations, which were of extreme importance, since we were trying to model a 3D figure based on 2D images.

Regarding the movements preformed, firstly the subjects were asked to look at the camera for a period of time, while trying not to move their head. This initial movement allowed to establish the initial distance to the camera as well as help to analyse their behaviour during a non moving phase. This one would also be considered the resting position, and an example can be seen in Fig.3.3

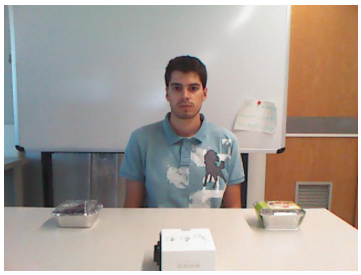


Figure 3.3: Resting Position

Secondly, the subjects were asked to move to the object on their right side and return to the initial position, then repeat this but now moving to the left. During this movement, as the subjects are trying to reach the cups, we can analyse the amplitude of their movement to each side. Moreover, this type of movement is important because it allows us to analyse both sides of the face of the subject more accurately, which is of extreme importance for the estimation of the subject's shape of the head. In practice, such movements are also quite used by people who can't properly talk or move, in order to point to what they want, so this factor increases their importance, as they almost work as a means of communication. Finally, by studying this movement we can also identify whether the subject performs this task with ease or not. In other words, we could detect if the subject was able or not to perform only the specific task, or if while trying to do it, there would be some sort of involuntary movements, such as

shacking. A subject is shown while moving to the right, in Fig.3.4.

Figure 3.4: Moving to the right

Afterwards, the subjects were asked to move as close as possible to the object in front of them and then come back, as it is shown in Fig.3.5. By doing this we could analyse how far they could move forward and how they did it. Regarding the feeding topic this movement is also quite important, since if the subject is not able to move forward, then the feeding spoon needs to go all the way to the mouth of the subject. However, in the case where the subject is able to move, they might prefer that the spoon stops moving at a certain distance and the subject them self moves forward in order to reach the spoon.

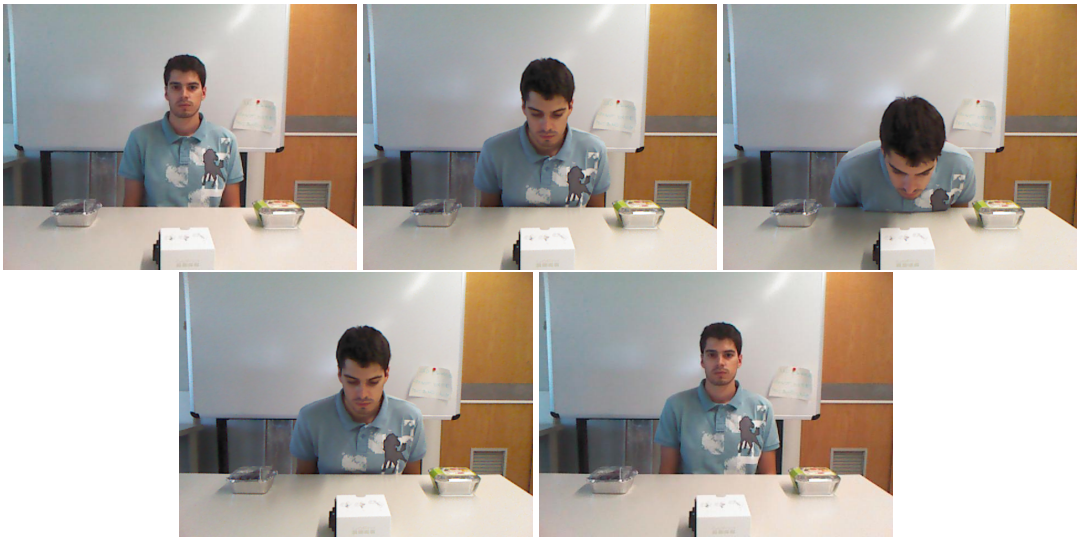


Figure 3.5: Moving forward

The final movement is shown in Fig.3.6 and consisted on going to the object on their left and then moving directly to the object on the right, without returning to the resting position. Such movement allowed to evaluate not only the amplitude of the movements of the subject, but also its fluidity.

Between each movement the subject would always go back to the resting position, since this would allow to always have the same initialization, which is of extreme importance when comparing the different

Figure 3.6: Moving to the right then directly to the left

subjects. Additionally, this also helped to later segment the video without having overlapping movements.

3.4 Segments

In order to better discern the different movements these were segmented into simpler ones.

This segmentation was done in accordance with the type of movement performed by the subjects.

In the case of the initial movement, resting position, corresponded to one full segment.

As for the second movement we had the obvious division between going to the left and right. However, based on the same data 2 segments were created for each direction. More precisely, the first segment would consist on going from the resting position up to the object, as shown in Fig.3.7.

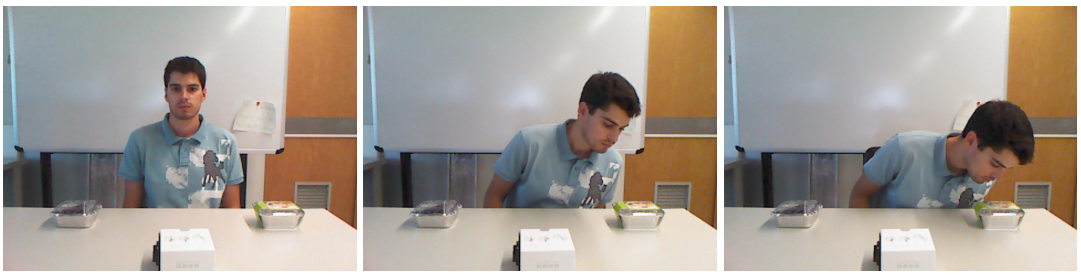


Figure 3.7: Reaching the object on the right side.

The second segment would be the full movement, so going from the initial position to the object and back, as previously presented in Fig.3.4. As mentioned previously, each segment was stored indepen-

dently and compared with the same type of movement of other subjects.

As for reaching the object in front of the subject we were also able to divide this movement into 2 segments. Similarly to the previous one, we divided by moving closer to the object (moving forward), as shown in Fig.3.8, and the full movement, previously presented.

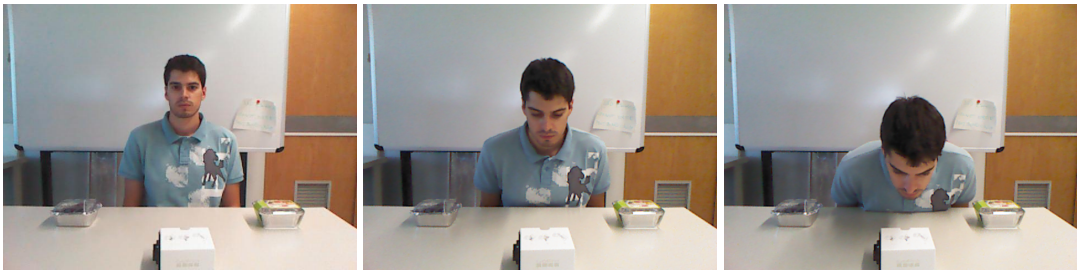


Figure 3.8: Reaching the object in front of the subject

Finally, for the last movement, this was analysed as a single segment.

In the end we had a total of 8 different segments that could be used to classify the subjects based on the movement performed.

Chapter 4

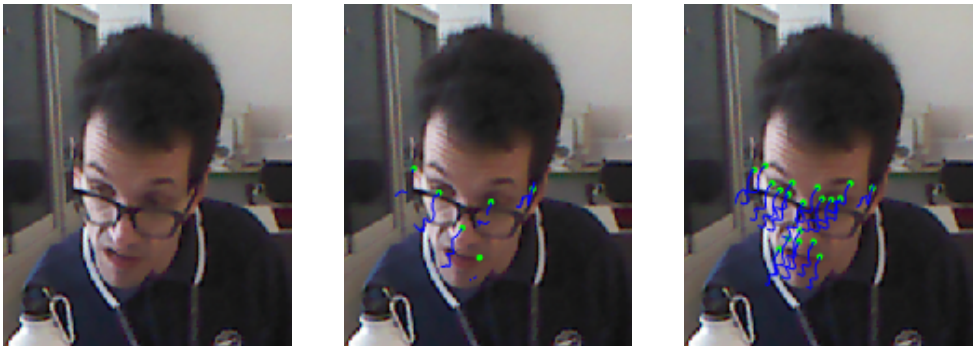
3D Reconstruction of faces from motion

The data stored on our data set consisted on movies with the subjects performing several actions. In order to classify the different subjects, it was necessary to find the trajectories of the points along the frames of the films. This was achieved by detecting the different facial landmarks throughout the movies with a facial landmark detector previously explained in section 2.3.

However, not all this information could be recovered with OpenPose, so we were left with incomplete trajectories. Therefore, in this chapter we explain the way used to estimate these entries.

This completion procedure has 3 phases. In the first one, the simpler coordinates are estimated based on their position in consecutive frames. Secondly, since the object we are analysing has some structure behind, by computing this model we can infer the coordinates of part of the points based on the known ones. Finally, the known points are tracked along the frames and their position is recovered.

In Fig.4.1, one can see an example of one frame and the trajectory of the points detected by OpenPose before and after estimating the unknown coordinates.



(a) Original figure (b) Incomplete Trajectory (c) Complete Trajectory

Figure 4.1: Example of one frame stored in the data set (a). The OpenPose output of the points in that specific frame (in green) and trajectory up to that frame (blue) is shown in (b). Full trajectories and points coordinates after estimation process (c).

4.1 Data matrix

The data obtained and stored in the data set consisted of RGB videos with the subjects performing several predetermined motions. In order to obtain some data regarding those movies, we used the OpenPose algorithm, previously described in section 2.3. This information consisted of the coordinates of all the points detected throughout each video and it would be stored in a matrix that later would be analysed. This data matrix, W, is $2F \times P$ and is shown in Fig.4.2.

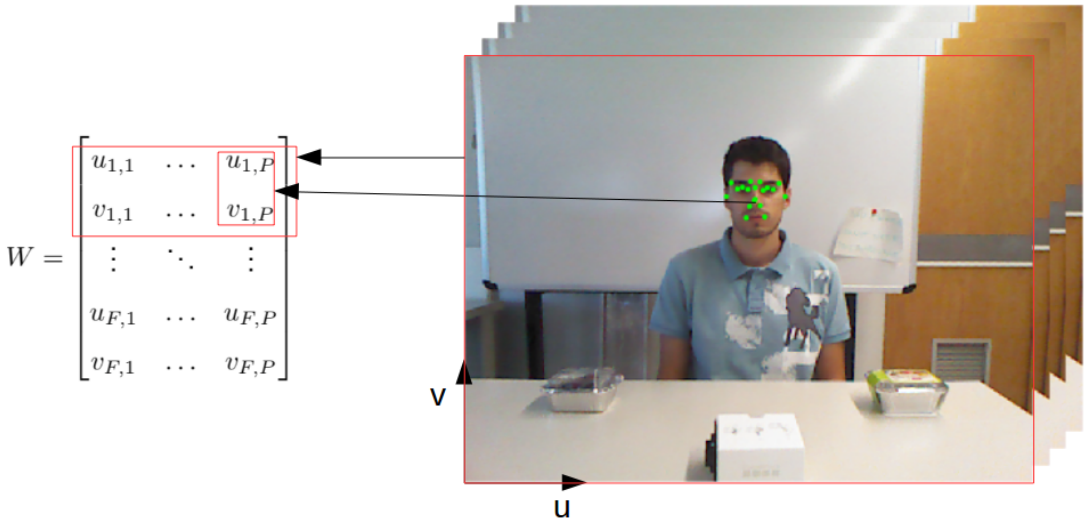


Figure 4.2: Data Matrix and original point in frame

In this case, the variable F corresponds to the number of frames and P the number of points tracked. More precisely we get the horizontal and vertical coordinates of each point for every frame, $\{(u_{fp}, v_{fp}) | f = 1, ..., F, p = 1, ..., P\}$ and then we combine and store it in W.

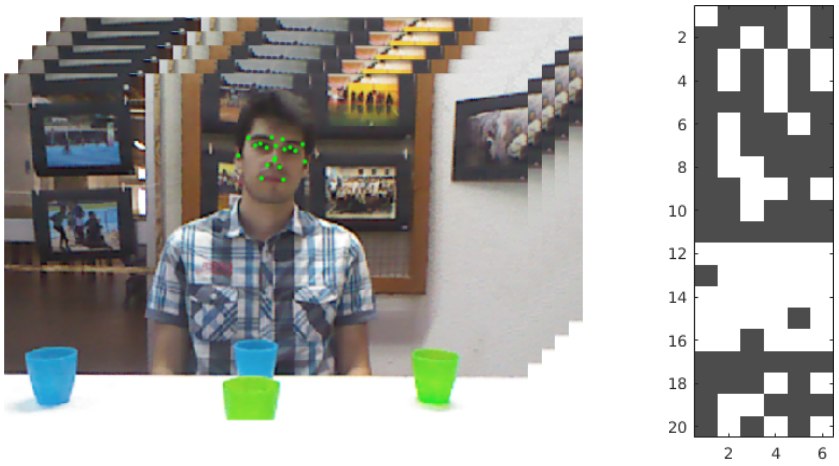
Regarding the number of points tracked, P, as mentioned in section 2.3, we did not use all of the points provided by the detector. Actually, out of the 71 possible points, only 19 were tracked in every subject. The points selected had some physical meaning and were easily detected. This way one could look at the estimation of the points and infer if their location seemed reasonable.

This detection algorithm not only provided the pixel coordinates of the different points throughout the sequence of frames but also the degree of confidence of each point. Therefore, one could use this information and store only those entries with a confidence above 80%. Consequently, the points used to determine the orientation of the face would have high reliability preventing misreadings.

Due to the omission of points with low confidence, the existence of points not detected by OpenPose and possible occlusions, our data matrix W was only partially filled, as there was a high percentage of missing entries.

Fig.4.3 presents an example of part of a data matrix that will be used to show how the missing entries are filled.

Therefore, we were left with a matrix with missing entries which we needed to fill in order to estimate not only the shape of the head of the subjects, but also the orientation throughout the frames.



(a) (b)

Figure 4.3: Frame with the tracked points (a). In (b) an example of a data matrix associated to the frames shown. In this data matrix white entries correspond to the ones OpenPose was not able to provide their position, so missing entries, and the known ones are shown in black. Each line of the data matrix corresponds to a frame and each column to one of the points.

4.2 Temporal constraint

As the data is acquired, it is saved in the data matrix in a sequential way, and so in every 2 lines of the matrix we have the new coordinates of every point at the current frame. As the frequency of the frames was relatively high, namely 17 frames per second, it was possible to assume that in a small time period the velocity of the points was constant.

In order to complete our data we considered that in the cases where some point was not detected only during 1 or 2 frames its coordinates could simply be predicted in a naive way. In other words, in those missing entries we would simply assume that the projection of the 2D trajectory of the points was linear and so we could interpolate the unknown entries based on the known ones, assuming that the time interval between frames was constant.

It was important to complete these entries in this way, as the algorithms that will be further explained lead to slight worse estimations for these points. Additionally, one can notice that this was only possible due to the fact that W was sorted by the frames, so consecutive frames would correspond to consecutive instants of time.

Fig.4.4 shows the previous example of part of a data matrix before and after the completion step just described. As one can see, the missing entries which were missing during 1 or 2 frames were recovered

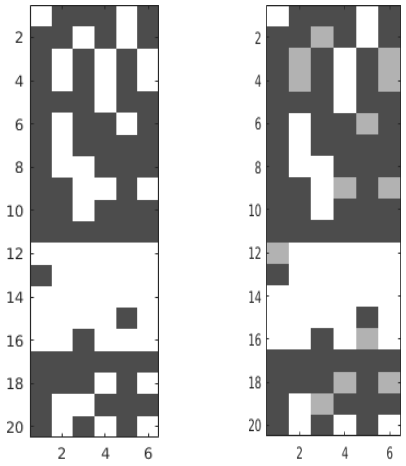


Figure 4.4: Example of part of data matrix before (a) and after (b) the first completion step, in which grey entries correspond to the completed ones.

4.3 Shape matrix estimation

In order to further estimate the missing entries different methods could have been used.

For instance, [42] has proposed a method of estimating the row and column spaces of the solution matrix, in two alternate steps. In the case of [43], a method for calculating the low-rank factorization of a matrix which minimizes the L_1 norm in the presence of missing data is proposed. Another method consisted of solving a problem of non-rigid structure-from-motion with rank imposition, proposed in [44].

In this problem, a factorisation algorithm was used to estimate the missing entries. This algorithm is able to solve the Structure from Motion (SfM) problem while handling degenerate data and missing entries [45]. This algorithm was chosen as it allows the estimation of the overall shape of the head of the subject, which would be used later. The algorithm proposed in [46] would also be applicable to our missing data, but [45] was used instead due to speed constraints.

This new algorithm is based on the Tomasi-Kanade algorithm, which is a factorisation method that allows us to recover the shape and motion of a given rigid object throughout a stream of images [47]. Actually, these images present the rigid object on different poses under orthographic projection, as can be seen in Fig.4.5. Given these characteristics, this algorithm assumes that all the keypoints of the rigid body under analysis is at a constant depth.

In order to apply the Tomasi-Kanade algorithm, first we must centre our data matrix, W, by subtracting to each row its mean, $W_c = W - \frac{1}{P} W \mathbb{1}_{P \times P}$. In this case $\mathbb{1}$ corresponds to a matrix filled with 1 in all its entries. Therefore, during this analysis, the translation of the face throughout the frames is not taken into account, and only the rotation is saved.

One of the key aspects of this matrix is the fact that it is rank deficient. Actually, the data matrix can be factorised into two different matrices, M and S, which is presented in equation (4.1).

$$W_c = MS \tag{4.1}$$

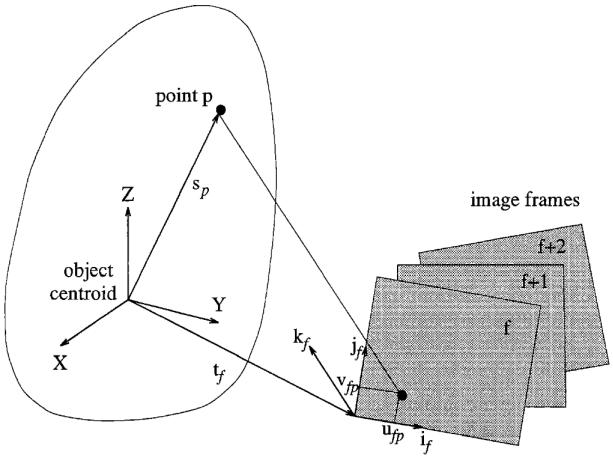


Figure 4.5: The two systems of reference used in the Tomasi-Kanade problem formulation [47].

In this equation, M is the Motion matrix, a matrix that encodes the camera rotation throughout the video. Additionally, M is $2F \times 3$ so, for every frame we have a 2×3 matrix, M_f, which encodes the camera rotation for the f^{th} frame.

$$M = \begin{bmatrix} M_1 \\ \vdots \\ M_F \end{bmatrix} \quad M_f = \begin{bmatrix} i^f_{1\times 3} \\ j^f_{1\times 3} \end{bmatrix}$$

The orthonormal vectors i^f and j^f point along the scanlines and the columns of the image of frame f, respectively, and are defined with reference to the world reference system, as shown in Fig.4.5. Therefore, with these vectors we can determine the orientation of the camera reference system at each frame. Additionally, we can conclude that M is the composition of several independent matrices each associated with its frame.

Regarding S, it is a $3 \times P$ centred matrix which encodes the positions of the different points of object under study relatively to each other. This matrix has on each column the 3D coordinates of each one of the P points, which in this case is 19.

$$S = \begin{bmatrix} p_{1x} & \dots & p_{19x} \\ p_{1y} & \dots & p_{19y} \\ p_{1z} & \dots & p_{19z} \end{bmatrix} \tag{4.2}$$

The estimation of S is of extreme importance, as it will help us filling the majority of those missing entries. Based on the coordinates of the points found out we are able to fit this shape into those and then we can predict where the unknown ones would be located. This means that based on the information of the structure and by knowing any 3 points, we can predict all remaining ones.

By analysing the dimensions of both M and S, we can see that their rank is at most 3. Therefore, this means that the rank of the data matrix W is also at most 3, without any noise.

In order to find these matrices we have to solve the minimisation problem presented in (4.3).

$$(M^*, S^*) = \underset{M,S}{\text{argmin}} \quad \|W - MS\|_F^2 \quad \text{subject to} \quad M_i M_i^T = \mathbb{I}_{2\times 2} \tag{4.3}$$

This optimization problem has a closed form solution which consists of applying the Singular Value Decomposition (SVD) to the data matrix in order to decompose it and obtain the M and S matrix, as shown below. More precisely, by applying SVD to W we can obtain 3 others matrices corresponding to the left-singular vectors, U, diagonal matrix with the singular values, Σ, and right-singular vectors, V. As this is a factorisation method, by multiplying these matrices as shown in equation (4.4) we can obtain W_c again.

$$W_c = U\Sigma V^T \tag{4.4}$$

Relatively to dimensions, U is $2F \times 2F$, Σ is $2F \times P$ and V is $P \times P$, by selecting the first 3 principal singular values and the associated left and right-singular vectors. We then have the previous matrices with the following dimensions: $\hat{U}$ is $2F \times 3$, $\hat{\Sigma}$ is 3×3 and $\hat{V}$ is $3 \times P$. By multiplying these matrices as explained before, we obtain $\hat{W}_c$ which corresponds to the best rank 3 estimation of W_c.

$$\hat{W}_c = \hat{U}\hat{\Sigma}\hat{V} \tag{4.5}$$

However, as stated previously, we want to factorize W into only two matrices. Thus we can arbitrarily define

$$\hat{M} = \hat{U}\sqrt{\hat{\Sigma}} \tag{4.6}$$

$$\hat{S} = \sqrt{\hat{\Sigma}}\hat{V} \tag{4.7}$$

and so, we can rewrite $\hat{W}_c$ as:

$$\hat{W}_c = \hat{M}\hat{S} \tag{4.8}$$

In fact, another step is required to truly estimate the true value both M and S, as $\hat{M}$ and$\hat{S}$ are no more than a linear transformation of those matrices. Further information regarding this detail is provided in Appendix A.

As the images are expected to be orthographic projections, the effects of camera translation along the optical axis are not accounted for. Additionally, different distances to the camera of different points are also ignored as the model expects that the object is so far away that those distances are irrelevant, so everything is assumed to be on the same plane, which leads to degenerate data. However, the greatest problem is the amount of missing data.

One of the ways to cope with the existence of degenerate data relies on the addition of the scaling factor [48]. More precisely, the motion constraints are updated, i^f and j^f instead of being orthonormal they are actually orthogonal, so the equations (4.9) are used to determine the new M:

$$
\begin{aligned}
i^f i^f &= \alpha^f \\
j^f j^f &= \alpha^f \\
i^f j^f &= 0
\end{aligned}
\tag{4.9}
$$

This new parameter α^f will correspond to the scaling of the shape for each frame, so different images at different distances can be acquired with no further problems.

Nevertheless, the problem created by the existence of the missing entries, can only be solved through the use of algorithm 2 presented by [45]. In this case the overall idea is the same as the one presented above. However, this new problem does not have a closed form solution and needs to be solved iteratively.

To sum up, with this method one can easily factorise the data matrix and obtain both the shape matrix of our subject and also partially complete the data matrix. The completion of the matrix was only partial since throughout the project there were some frames with so little visible points that the coordinates of the missing points in these frames could not be determined by relying only on this tool. Therefore, only the frames with at least 6 points visible out of the 19 tracked points were completed and used to estimate the overall shape matrix. If the problem was noise free only 3 would be required. However, the setup was too susceptible to noise due to the low quality of the camera. Moreover, the points were too likely to fall into a subspace, leading to highly degenerate frames. So, a higher number of points had to be used in order to maintain the reliability of the points' estimations.

Fig.4.6 shows the previous example of part of a data matrix before and after the completion step that has just been described above.

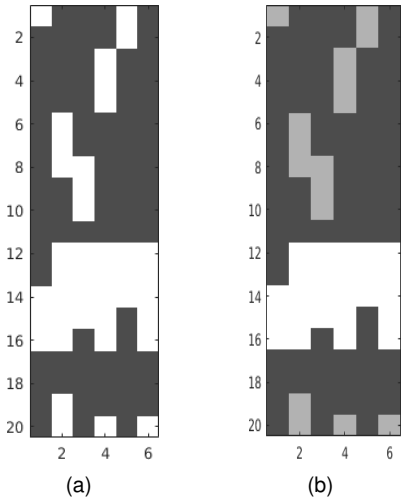


Figure 4.6: Example of part of data matrix before (a) and after (b) the using the algorithm described.

Below, in Fig.4.7, one can see the final result of the shape matrix obtained from one person, from different perspectives.

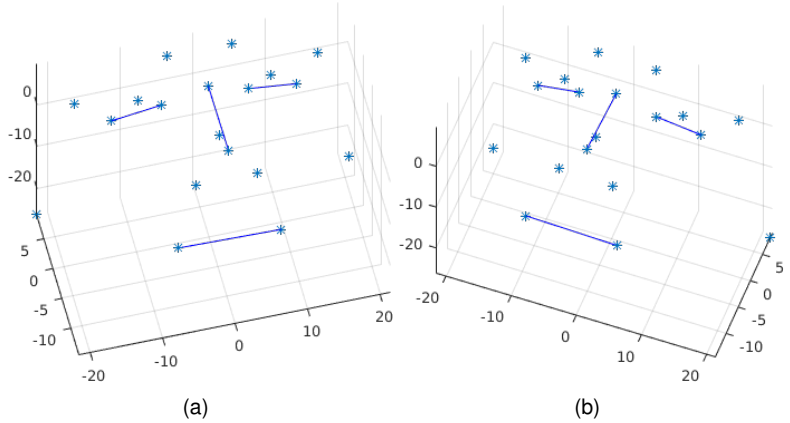


Figure 4.7: 3D Shape matrix in different poses, representing the shape of one subject.

4.4 Motion matrix estimation

Even though part of W had already been filled and S acquired, this was not enough to proceed to the classification of the subjects, as we were also interested on the motion of the subjects while they were performing the required movements. Therefore, it was imperative to acquire the position of the remaining points that had not been detected yet.

As it was possible to see, W was quite heterogeneous regarding the observations. At the moment, this matrix either had large blocks completely filled with the expected entries, or blocks full of missing entries. Moreover, these blocks had few detected points which was unfortunate as there was no clear way to predict these points coordinates.

One could think about applying the same methodology as in section 4.2. However, during a longer time period the person is less likely to move linearly. So, since the gaps were too large, this type of estimation would most likely give unsatisfactory results.

In order to estimate the position of these points, the Lucas-Kanade method for feature tracking, explained in 2.4, was used for each block of consecutive missing entries.

The points previously detected in the completed frames were used as initialisation for the missing points we wanted to track. This means that we would start in the known frames, and try to predict where the points had travelled to based on the optical flow equations. In the specific case where a point was already known then we would simply use that stored point, otherwise we would rely on this tracking method.

After tracking the new positions of the missing data then we would make use of S, obtained in the previous section, and impose it into the new points. In other words, we wanted to find the best approximation of the data points, which actually also shared the same structure of the Shape matrix.

This was quite important as the points obtained through the Lucas-Kanade algorithm would be simply the estimations of the new positions of these points, but there was no real interaction between them, so there could be the case where geometry of these points would not be in accordance with S.

Actually, this imposition problem consists on solving an orthogonal Procrustes problem with scaling [49]. This can be expressed as finding the matrix, Ω, that was able to align both S and our data.

Actually, we want to rotate S, but then as we are treating 2D data we want the object rotated, but only the 2D result is of interest. Therefore, Ω will be a 2×3 Stiefel matrix and it will be composed by two orthonormal vectors which better rotate S onto our 2D data. Additionally, we also wanted to find the scale factor, β, that would correspond to the scale between both matrices. To sum up, this problem could be mathematically expressed as a minimisation problem, as shown in equation (4.10).

$$\begin{aligned} (\beta^*, \Omega^*) = \underset{\beta,\Omega}{\text{argmin}} \quad & \|\beta\Omega S - X\|_F^2 \\ \text{subject to} \quad & \Omega\Omega^T = \mathbb{I}_{2\times 2} \\ & \beta > 0 \end{aligned} \tag{4.10}$$

Where X corresponds to the centred coordinates of in study and has dimensions of $2 \times P$.

This problem has already been solved and its closed form solution boils down to computing the SVD. More precisely, if we define $A = XS^T(SS^T)^{-1}$, then Ω will correspond to the multiplication of the left and right singular vectors and β will correspond to the average of the singular values.

$$A = XS^T(SS^T)^{-1} \tag{4.11}$$

$$A = U_{2\times 2}\Sigma_{2\times 2}V_{3\times 2}^T \tag{4.12}$$

$$\Omega = U_{2\times 2}V_{3\times 2}^T \tag{4.13}$$

$$\beta = \frac{1}{2}\mathbb{1}_{1\times 2}\Sigma\mathbb{1}_{2\times 1} \tag{4.14}$$

Therefore, by knowing Ω and β we can estimate the best position for the points given a certain shape. This way we were able to get a more realistic approximation of the points we were trying to track which we could then use to help us fill our data matrix. Moreover, the application of this method would even allow to predict the position of missing points for the cases where the Lucas-Kanade method was not able to predict the final position.

One of the biggest disadvantages of this procedure relies on the fact that as the number of frames analysed increases we are subjected to higher error. So, when the last frames of the block with missing entries was reached, the points' coordinates would highly diverge from the expected result. In order to avoid this problem the same idea was used twice. More precisely, the points were tracked along the normal appearance of the frames, and then tracked again, but backwards, so starting in the last frame, and finishing on the first one. After tracking the points both ways, we would apply a weighted average where in the first case the weight decreases as the number of frames increases and in the second case the opposite is performed.

Just like in section 4.2, this was only possible to apply due to the fact that W was organised with the frames in accordance with the natural flow of time.

After running this procedure in each block of missing entries we finally reached our goal of having our data matrix W completed. Then it was possible to estimate W over all the frames of the movie, as can be seen in Fig.4.8.

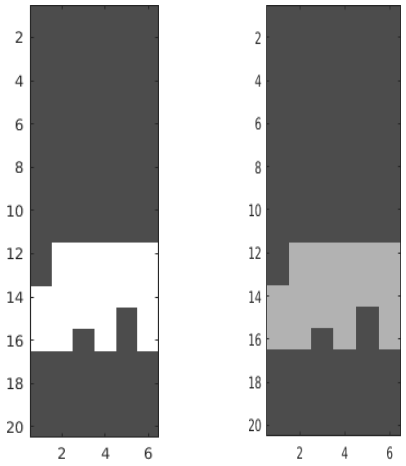


Figure 4.8: Example of part of data matrix before (a) and after (b) the using the algorithm described.

4.5 Correcting high confidence entries

After recovering all missing data one might think that this would be enough and we could proceed to the classification of the movement. However, there was still one error that had not been accounted for so far, which was the errors from the facial landmark detector. This means that even though some points provided by the detector have high reliability associated to them, they might actually be wrong.

This can lead to some problems since during this process we are assuming that we can fully trust these points with high confidence, which is not ideal.

Another step was added to the overall process to prevent this problem. More precisely, after calculating the initial estimation of S, we corrected the raw data obtained by trying to fit the shape into the known data. The way to do that was by trying to fit the shape into the known data. Similarly to the idea explained in the previous section, we also try to impose the shape on the data, and we confirm if there is any point on the data which is too far away from where it should be. If that is the case then that point is also eliminated in order to improve our final result.

This method could only be applied in frames which had at least 4 points, otherwise the points available were not even enough to align both structures.

After filtering the raw data provided by the detector, all the process of recovering missing entries would be run again in order to complete our data matrix one last time.

Chapter 5

Movement Classification

In this chapter, we explain the way we extract features from the subjects and later how these are used not only to classify the different subjects, but also to create a classifier to label the subjects.

More precisely, two different types of features were developed in this thesis, which aim to encode the trajectory of the users head. These features were independent of the subject, so our data had to be pre-processed before being encoded into those descriptors.

This pre-processing step consisted on removing the influence resulting of the different head shapes, while keeping in the data matrix the pose of the head over the frames analysed.

Then, the features which encode blocks of images were used to classify the subjects. This classification relied on the BoW model, but since we were using the features, we would be classifying blocks of images, instead of independent images.

5.1 Pre-Processing Data Matrix

In this section the pre-processing of our data is further discussed.

In order to compute the trajectons, further described, we needed to make the data matrices invariant to the shape matrix of each subject. This was achieved by defining a reference shape and then finding the transformation between this one and the shape associated with the data matrix under analysis.

As the transformation between this shapes is defined, we can adapt the data matrix in order to remove its influence.

In the end, the data matrix would only depend on the orientation of the head throughout the movie without being influenced by S.

5.1.1 Reference Shape

As mentioned previously in section 4.3, from the data matrix, W, it was possible to extract both the Motion matrix, M, and the Shape matrix, S. Since we wanted to analyse only the type of movement, one might wrongly think that analysing directly M would be enough. However, that is not the case, since each subject would have a different S, with its own orientation and dimensions.

Actually, we do know that for each movie S was constant, and that for each frame, M_f had two orthogonal vectors that expressed the orientation of that specific shape matrix, while taking into consideration its orientation.

In order to avoid this problem, a Shape matrix of reference was created, S_{ref}. Then, every S was mapped onto the one of reference, and later, this linear transformation would be used to adapt M, and normalise W.

Regarding S_{ref}, it was used a S from one subject, where tracking went particularly well. Then this matrix was normalised so as to have its first singular value equal to 1.

After defining our standard shape matrix, we had to find the linear transformation between these matrices, which would encode both the scaling and rotation between them.

Regarding, the scale factor, different people have their own head dimensions and these differences can not be summed up to a single value. In this case, it was assumed that the variations in the head's shape mostly fall to different scale factors along the 3 principal directions. Therefore, we defined a 3×3 diagonal matrix, D, to register these differences.

In the case of the rotation between S_{ref} and S, a simple rotational matrix, R, was required to align the two of them.

With both R and D it would be possible to align S_{ref} and any S. In order to determine these new matrices we would then need to solve the optimisation problem presented in equation (5.1).

$$\begin{aligned} (R^*, D^*) = \underset{R,D}{\text{argmin}} \quad & \|RDS_{ref} - S\|_F^2 \\ \text{subject to} \quad & RR^T = \mathbb{I}_{3\times 3} \\ & D_{ii} > 0,\ i = 1,2,3 \\ & D_{ij} = 0,\ i = 1,2,3,\ j = 1,2,3, \text{ and } i \neq j \end{aligned} \tag{5.1}$$

Actually, this problem consists on an anisotropic Procrustes with post-scaling. It is similar to the problem presented in equation (4.10), but now instead on having a simple scale factor we have another matrix, which greatly increases the degree of complexity of this problem.

However, the formulation presented for problem is not in the form we currently have a solution for [49]. Therefore, it had to be rewritten so as to adjust its form. This reformulation is further explained in Appendix B.

After rewriting our problem we had to solve the optimisation problem shown in equation (5.2).

$$\begin{aligned} (R^*, D^*) = \underset{R,D}{\text{argmin}} \quad & \|X_1RD - X_2\|_F^2 \\ \text{subject to} \quad & RR^T = \mathbb{I}_{3\times 3} \\ & D_{ii} > 0,\ i = 1,2,3 \\ & D_{ij} = 0,\ i = 1,2,3,\ j = 1,2,3, \text{ and } i \neq j \end{aligned} \tag{5.2}$$

However, according to [49], this new problem does not have a closed form solution, but can still be solved iteratively as shown in Algorithm 1.

Algorithm 1 Anisotropic Procrustes Algorithm

Initialise D, D_{aux} and R
while $\|D - D_{aux}\|_F^2 \geq \epsilon$ **do**
  $D_{aux} = D$
  $D = (\mathbb{I} \circ (R' X_1' X_2))(\mathbb{I} \circ (R' X_1' X_1 R))^{-1}$
  $[U, S, V] = SVD(X_1' X_1)$
  $\mu = S_{11}$
  Initialise R_{aux}
  while $\|R - R_{aux}\|_F^2 \geq \epsilon$ **do**
    $R_{aux} = R$
    $Z = D(X_2' X_1 + D' R' (\mu \mathbb{I} - X_1' X_1))$
    $[U, S, V] = SVD(Z)$
    $R = VU'$
  end while
end while
return D and R matrices.

In this case, $\circ$ corresponds to the Hadamard product, or element-wise product.

Moreover, in Appendix B, besides explaining the reformulation done to the optimisation problem, some simplifications done to the Algorithm 1 are also mentioned.

By repeatedly updating both D and R in this algorithm, it was possible to determine their final values and with them we finally had the required elements to transform S_{ref} into S.

5.1.2 Standardise Data

As the shape has been standardised, then we needed to update each subject's W, so as to remove the influence of the subject's S.

As we saw in the previous subsection, our data matrix can be factorised into S and M. Based on this information and the assumptions made on the previous section, we can then write the following equalities:

$$W_c = MS = MRDS_{ref} \tag{5.3}$$

In fact this is not true, since RDS_{ref} will only be an approximation of S. Nevertheless, the differences between the two are neglectable, and in practical terms these will not influence the final result.

Regarding D, this matrix was used to get the S_{ref} with the same dimensions as S. However, we want to focus on the orientation of the head throughout the different phases of the movements, and not to classify the different subjects based on their head's dimensions, so this term could be neglected.

In order to perceive the relative distance between the subject and the camera itself it was used the scale factor, previously described in section 4.3. For instance, if the scale factor in one frame is twice the value in other frame, we know that in the first frame mentioned the distance between the subject and the camera is half than in the second frame. Increasing the distance leads to a smaller shape which is then associated with a smaller scale factor.

Due to the fact that the recordings were performed with the same standardised measures, we knew that in the first frame of the different recordings, the subjects were approximately at the same distance

to the camera. Therefore, the scale factors along the video were normalised based on this value. This way, while the subjects were on the resting position the scale factor would be around 1 and as they move closer to the camera, this value would increase accordingly. These values were saved in a column matrix, Λ.

Regarding the orientation of S_{ref}, since R was a rotational matrix, by multiplying both M with R, we got a matrix whose properties were similar to the ones of M. Therefore, in order to store the rotation applied to S_{ref}, it was created M_{norm}, which corresponded to:

$$M_{norm} = MR \oslash \Lambda \tag{5.4}$$

This matrix stored both the information from both rotations in M and R, while removing the influence of the scale factor. Regarding the symbol $\oslash$, this corresponded to an element wise division.

So, with M_{norm} and S_{ref} we were able to reformulate the data matrix of each subject, creating then a new one, as expressed in equation (5.5). This new normalised data matrix, W_{norm}, would be comparable with the other ones acquired and would only hold information regarding the actual orientation of S_{ref}.

$$W_{norm} = M_{norm} S_{ref} \tag{5.5}$$

In the end, we got the data matrix normalised and the relative scale factor stored independently on Λ. With this information it was possible to extract the features further explained and to classify the different subjects.

5.2 Features

After normalizing the data matrix we were left with the relative position of the points of interest over the several frames, as shown in Fig.5.1.

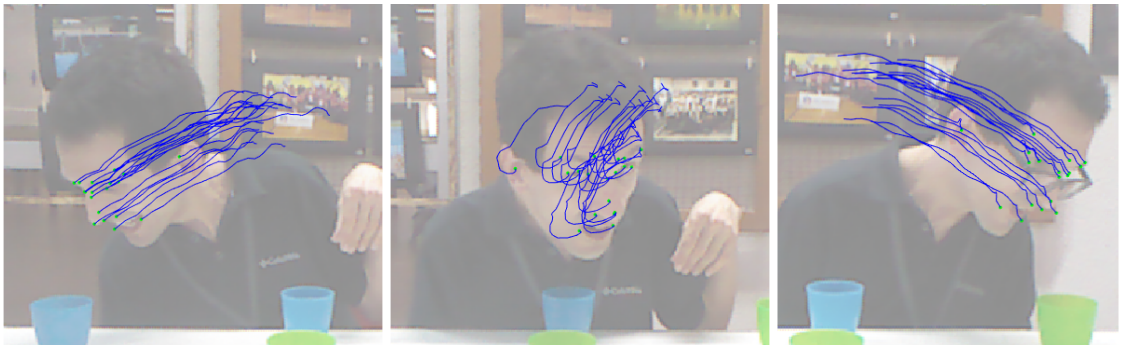


Figure 5.1: Examples of the projection of the trajectory of different points over a block of frames

In order to classify this data set, two different features were developed and tested. These mainly focused on analysing the orientation of the head's shape in two different ways.

Moreover, the scale factor, which encoded the distance to the camera, was also taken into consideration. However, the translation along both perpendicular axis between the camera and the subject was

not taken into consideration.

As we were trying to analyse the behaviour of the subject while performing a certain task, only evaluating independent frames would not be of much interest, so blocks of frames were analysed as a whole. The process to select these blocks was similar to the sliding window method. More precisely, we would look at the first block of 25 consecutive frames, then we would look at the next block with the same dimension, but now starting in another position. In Fig.5.2, there is an example of the frames selected each step.

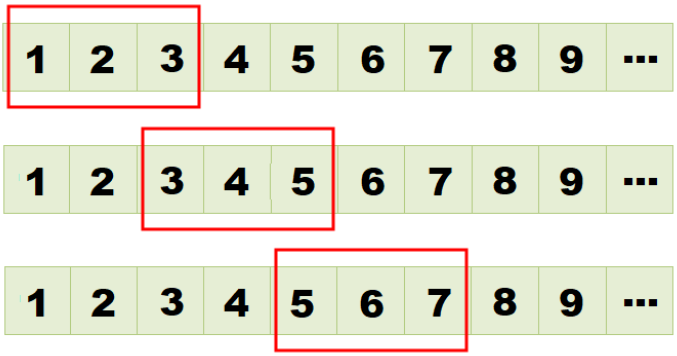


Figure 5.2: Example of sliding window with a step of 2 and width of 3

For each subject several blocks of frames would be created until all the movie had been covered. In the end, each one of these blocks would lead to an independent trajecton.

These features focused on analysing the behaviour of the points during blocks frames. However, they were not the same as the actual trajectory of the points analysed. Therefore, they will be referred as trajectons, inspired by [50].

5.2.1 6P feature

This feature consisted on creating a discretized 3D cube, where we would project the reference shape according to the pose of the subject each frame. This cube would be voxelized and each voxel would be scored as the facial landmarks passed by them. Overall, this feature would focus on the spatial distribution of the points over the frames.

Therefore, this feature focus more on the space occupied by the points over the frames.

Regarding this special cube that was designed, it had 3 axis with two different types of information. In the first two axis, the position of the points tracked was stored and the final axis would store the information relative to the distance between the head of the subject and the camera, which corresponded to the scale factor. These axis can be seen in Fig.5.3.

In Fig.5.4, one can see the representation of a single point, in 25 consecutive frames, inside the cube currently being described. This image shows the case of subjects with and without disabilities, in order to emphasize the difference between the two subjects.

One of the most important factors of this feature relied on the fact that actually not all the 19 points tracked were used in this configuration. Since we did not label the score of the voxels then we had no

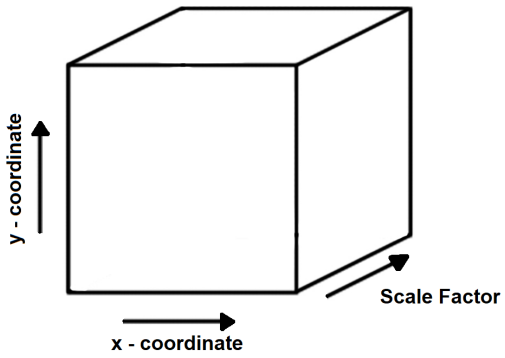


Figure 5.3: 6P trajecton cube with respective axis

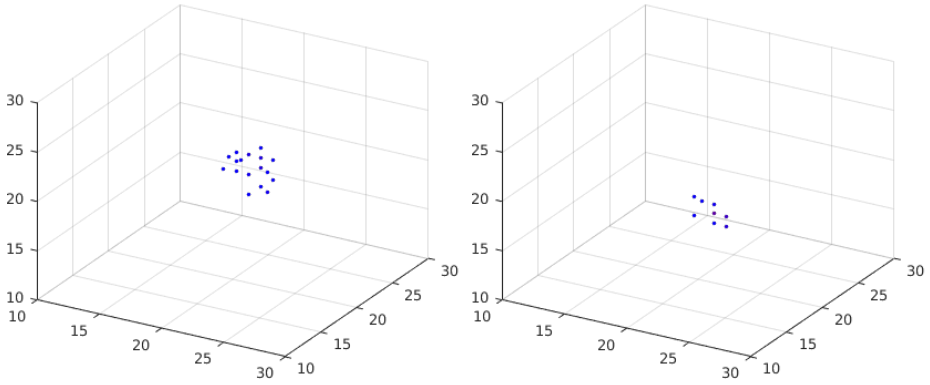


Figure 5.4: Example of the 6P trajecton formed by one single point of a subject with (a) and without (b) disabilities, during 25 frames

way to know which point passed at each place, so we only use 6 points.

These corresponded to the corners of the mouth, outer corners of the eyes and the vertical corners of the nose. As it is possible to see in Fig.5.5(a), all the selected points are reasonably well-spaced between them. In contrast, in the case of Fig.5.5(b) we can see that with the 19 points, they can easily move on top of each other.

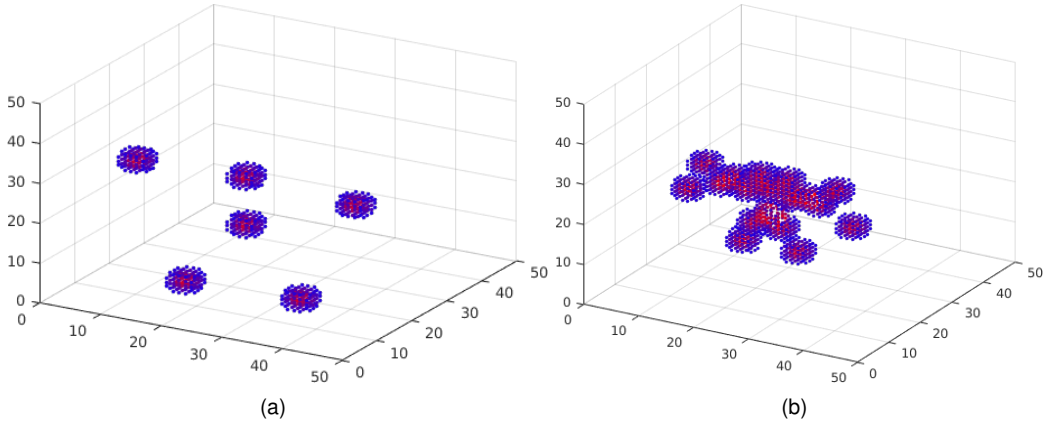


Figure 5.5: Representation of one frame with (a) 6 points, (b) 19 points, using the 6P methodology.

As the voxels are inside the cube it is reasonable to assume that neighbouring voxels have similar

meaning. However, as we pass this information to a vector their relationship is lost. So if we try to cluster two different subjects with similar behaviour, where for some reason the voxels occupied by their points slightly differ, these two will never be grouped together, as they do not share the same voxels.

In order to obtain a more robust descriptor, the score of the adjacent ones were also improved. By increasing the score of the neighbouring voxels, we induce a connection between them, that later on during clustering can help to better classify the different subjects.

We consider a small cube, with $5 \times 5 \times 5$ voxels, which we denominated by Gaussian Box, where the score of each entry is given by the following equation.

$$score(i, j, k) = e^{(\frac{-\|[i,j,k]-\mu\|^2}{2\sigma^2})} \tag{5.6}$$

This cube would be added to the voxel where the points were present, as well the neighbouring ones.

In this equation i, j and k correspond to the entries of the cube, relative to the actual position of the point. Regarding μ, this is the centre of the cube. Finally, σ was a parameter used in order to control how much the values of the cube would decrease as the distance to the centre increases. After testing different values, σ was imposed the value of 1.5.

The overall Gaussian Box can be seen in Fig.5.6, with the score of each entry of the cube coded in its respective colour.

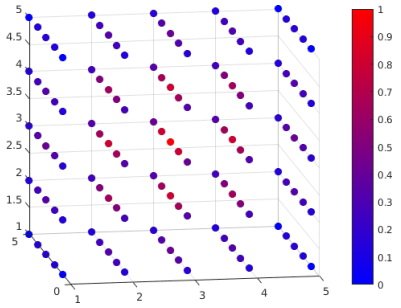


Figure 5.6: Representation of the Gaussian Box described. The values on entry of the cube can be estimated based on their colour.

5.2.2 Shaky feature

This second feature focused on analysing the degree of the oscillations in the different movements. On the one hand, one would expect that in the case of subjects without disabilities their movement would be fluid with little to no shacking, so the rate of change would be essentially constant.

On the other hand, people with disabilities have lower control over their movement. Therefore, while they try to perform a certain action, there are still a lot of involuntary movements. So, the movement performed by these subjects will have the tendency associated with the movement requested, as well as several other random changes in other directions.

Therefore, this feature focus more on the temporal changes of the coordinates of the points.

It is important to remember that W was a $2F \times P$ and every 2 rows would encode one frame. These frame would have both directions (horizontal and vertical) encoded on each line of the matrix.

In order to analyse the rate of change of the movement first the position of the points was discretised. After this step, in order to determine the direction where each point was moving, it was subtracted, to each frame, the values from the previous one. Therefore, now we would have for each point and each frame 2 different values encoding the direction of the movement.

These could take any value inside the range $\{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$. More precisely, a positive value would be associated with an increment of a specific point in the direction under analysis; 0 in case the point did not move substantially; and a negative value when the coordinate of the point would decrease. Therefore, for each direction, each point would take one of 9 possible values.

This increment meant that instead of only detecting the direction of the movement of the point based on the signal, we were also able to know by how much it actually moved.

As each point could move along 2 directions and each direction had 9 different intensities, by combining them all, we would have a total of 81 different alternatives for each point's rate of change, as shown in Fig.5.7.

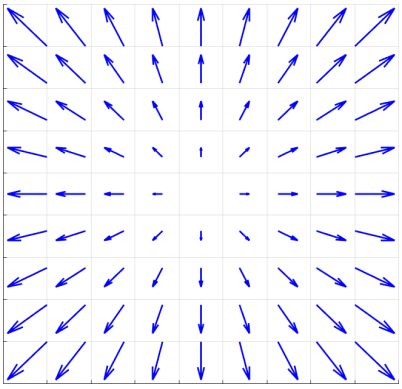


Figure 5.7: Possible directions and given intensities that each point can take, while moving between 2 frames

Due to the fact that the points on both ears were too unstable, the rate of change of these points were included in this trajecton. The lack of reliability of these two points, came from the fact that during the movie, the subjects were almost always facing forward. Therefore, the amount of good readings associated with these points was too small as they were not detected that many times. So fewer readings, inevitably, led to worse estimations.

In order to encode the rate of change of the scale factor, a similar process was applied. As this value is referent to the shape as a whole, we only had nine possible cases which would encode whether this value increased or decreased, and the intensity of this change.

Similarly to the previous feature, a procedure comparable to the Gaussian box, which is described above, was implemented so as to approximate similar trajectories. In this case, it was only required a 3×3 square, centred in the direction where each point was moving each frame. For example, if a point was moving to the left, then the correspondent entry would increase its value by one and, due to this propagation, entries such as moving diagonally to the left and up and down, as well as not moving would also receive a small increment, as they are the directions immediately next to the predicted one.

In the end, each frame would store the changes of its points and by combining several frames we

would be able to get our feature, as shown in Fig.5.8.

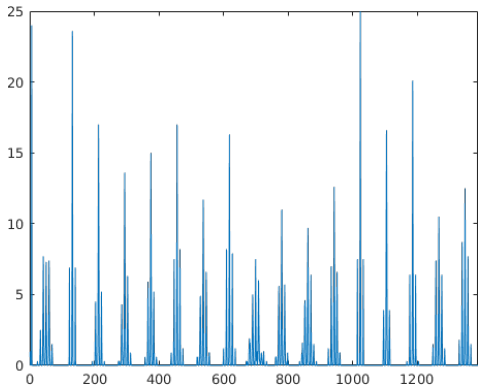


Figure 5.8: Example of the Shaky trajecton.

5.3 Classification

After creating the different features associated to each subject, we were able to create our codewords. These would correspond to the centres of the clusters formed through k-means clustering. This algorithm aims to partition our data into k clusters, while assigning to each observation a label based on the closest cluster. This assignment would be based on euclidean distance and would try to preserve similar trajectons in the same cluster.

This clustering algorithm was applied independently to each type of Segment analysed, so that different movements would have different clusters associated. The clusters formed with this algorithm would correspond to the codewords and with them we were able to create a codebook or dictionary associated to each segment.

Regarding the number of clusters used, this was a difficult problem as there was no optimal value. In this case, k, was tested with the values from 3 to 10. The reason behind these values relied on the fact that below 3 clusters we were not dividing enough times our data (we would have more types of subjects than clusters). On the other hand, more than 10 clusters would lead to some clusters aggregating too few samples and most of them from a single subject. Which in turn wouldn't be dividing the different ways of performing the movement, but instead simply dividing the subjects from one another.

In order to classify a new subject, firstly, the trajectons associated to each type of movement were created. Afterwards, the trajectons were compared with the clusters stored on the respective codebook and each trajecton was then associated with the closest cluster.

As all the trajectons are associated with their respective cluster we can construct an histogram of the subject under analysis. Consequently, this histogram reflected the percentage of trajectons associated with each cluster.

As we already knew the subjects classification of the training set, for each segment, we could find the closest one using the nearest neighbour algorithm.

In other words, we could see which known subject resembled the most the subject under analysis and simply assume they had the same degree of disability.

In the end, this process was repeated twice, once for each type of feature presented in 5.2.

Chapter 6

Experimental Evaluation

In this chapter we will look at the results obtained during this project.

More precisely, the data obtained from the facial landmark detector is evaluated. Moreover, the data matrices obtained from OpenPose output as well as their rate of completion were analysed. This analyse was merely qualitative, as there was no ground truth which could be used to support the results obtained.

Following this analysis, we proceed to compare, for both trajectons, the different the result from different people while performing the several movements.

Finally, the results obtained after the classification of the subjects will be presented.

6.1 3D reconstruction

6.1.1 Filling Missing Data

As it was explained in chapter 4 the data matrix obtained from the facial landmark detector was filled with missing entries. In the cases of subjects without any type of involuntary movement and with the full control over its movements, this matrix would have about 40% of missing entries. In contrast, a subject whose movements were not as controlled as intended this value would be about 55%, while reaching, in some cases, values of 73%. One would expect that in both cases the amount of missing data would be the same, since all the conditions were the same (setup, movements requested and facial landmark detector used). However, that was not the case, so we can predict that the neural network behind the detector was trained with people without disabilities. Consequently, in the case of subjects with disabilities, the points detected would overall have lower confidence, which led to the increment of the difficulty that this challenge presented.

Additionally, one can also emphasise the fact that by simply using glasses, then the overall percentage of missing entries would increase by at least 5%.

Presented in Fig.6.1 one can see the overall distribution of missing entries throughout the data matrix, of two different subjects, after these were collected by the facial landmark detector.

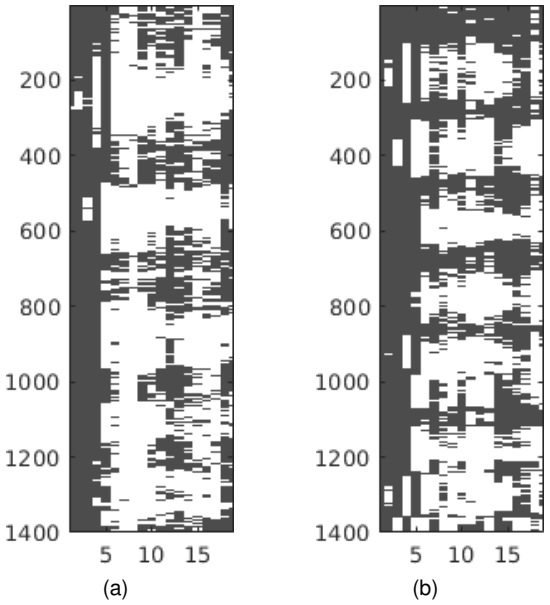


Figure 6.1: Examples of different data matrices. Image a) corresponds to a motor-impaired person and b) a person without any disability. In this case the white area corresponds to missing entries and the grey one to the entries with values obtained from OpenPose.

By applying the procedure described in chapter 4, after completing the predictable missing entries and using the algorithm proposed by [45] we could further complete our target matrix and build the Shape matrix.

In Fig. 6.2 it is possible to see the previous matrices after this initial completion phase.

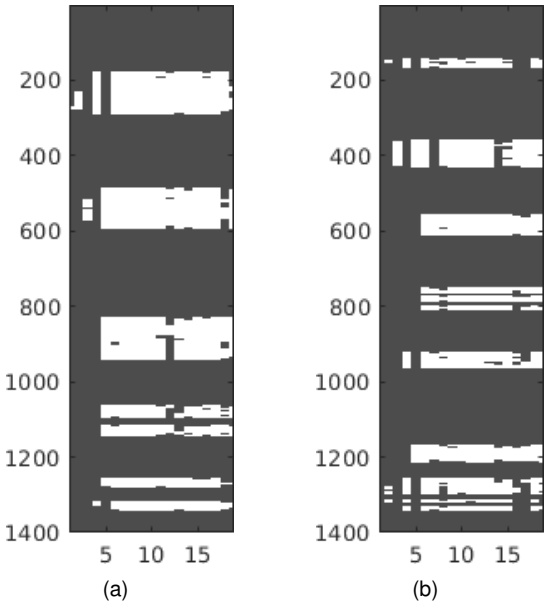


Figure 6.2: Examples of different data matrices partially completed. Image (a) corresponds to a person with disabilities, (b) to a person without any disability.

As is is possible to see, the data matrices obtained after this completion step still had big gaps of missing entries. Moreover, these gaps with almost no information were much bigger, in the case of the subjects with disabilities. This means the tracker would loose their points during much more frames.

Part of this problem relied not only on the fact that the detector had more difficulties to detect the features from people without disabilities, but also because these subjects would take more time to perform the different movements. Therefore, these subjects would spend more time in poses which were harder to detect and so the corresponding points of the face would be unknown for a longer period of time.

After using the algorithm presented in section 4.3 to complete part of the data matrix, only the frames with too little points detected would still be incomplete. Moreover, there were different types of gaps throughout the data matrix. Some of them would consist only of 3 frames, but others would be longer than 50 frames.

Regarding the smaller gaps, the method proposed was quite reliable, it would easily detect the new position of the points of interest and properly adjust them to the shape matrix. However, the larger ones were quite alarming, since we were tracking the key features during so many frame, we were bound to accumulate errors and to have our points deviating from their true position.

Nevertheless, due to the fact that we run the Lucas-Kanade algorithm in the two directions, and also with the help of the imposition of S, we were able to successfully complete all the gaps independently of their size.

In Fig.6.3 we show one example of the points tracked during this last phase. In case of Fig.6.3(a), the tracker only had to follow the subject for 15 frames, whereas in Fig.6.3(b) the tracked followed for 63 frames. As it is possible to see, in both cases the tracker is able to provide a good estimation of the points.

(a) (b)

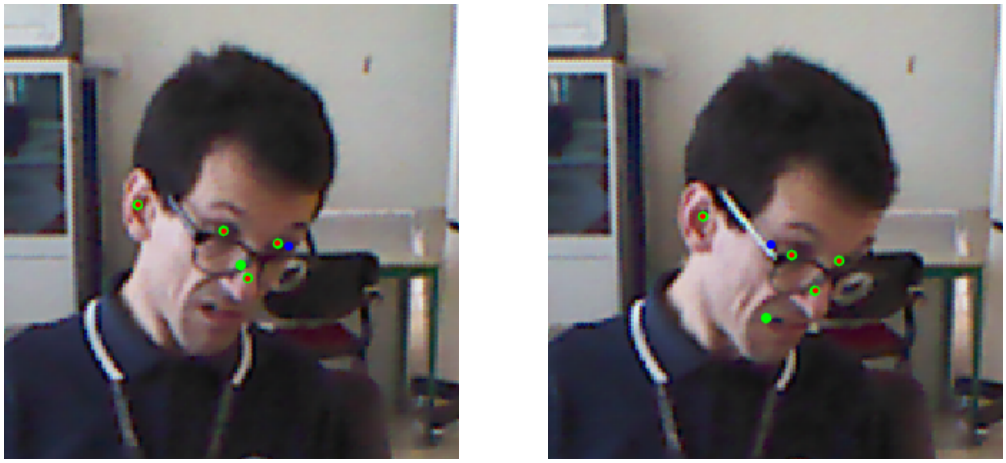
Figure 6.3: Examples of the points tracked during the third phase of the estimation of missing entries. Image (a) corresponds to a detection for a short duration (15 frames), (b) corresponds to a detection for a longer duration (63 frames). More precisely, the image correspond to the frame at the center of the gap, as this would be the one with highest error associated

One might point out the fact that in the figure with the bigger gap, the points of the mouth corner's are not reliable. However, this discrepancy is mainly due to the fact that we are not dealing with a rigid body. So, since the tracker also uses the previously determined shape of the subject's head, face expressions are not taken into account during the detection, which led to a poorer detection of those exact points. Nevertheless, the remaining points position is reasonably good.

6.1.2 Improving the face detector output

As it was explained previously in section 4.5, after completing the data matrix, S was used to filter the output provided by OpenPose. This way points with high confidence but wrongly placed could be detected and later corrected.

In Fig.6.4 it is shown two different cases where points were eliminated due to the fact they were badly assigned. More precisely, in Fig.6.4(a) the green point next to the nose was labelled by the OpenPose as the eye on the right (in blue). Regarding Fig.6.4(b), what seems to be the left corner of the mouth, in green, actually is labelled as the outer corner of the eye on the left (in blue).



(a) (b)

Figure 6.4: Points wrongly assigned by OpenPose. The points in green correspond to the ones provided by OpenPose. The points in red correspond to the ones that are actually well identified.

These deletions allowed to improve our data matrix, as there would be less errors in what was considered the ground truth.

Fortunately, this type of errors mainly occurred in frames where the number of points detected was quite low. Therefore, most of the time these frames were not used in the first phase of the completion of the missing entries, so the shape matrix wasn't influenced those corrupted entries, as can be seen in Fig.6.5.

Nevertheless, even if there was the case where the points wrongly assigned were used in the first completion phase, their quantity would be much lower than the ones considered well defined. Therefore, as we were building our Shape matrix, their influence would be negligible to the overall structure.

Overall, after this correction, the data matrix would suffer almost no impact from this misclassification. Essentially, the improvement would be noticed mainly in the detection of those specific points, and partly in the points of the same frame, or surrounding ones.

In Fig.6.6, it is shown the same figure as in Fig.6.4, but now, with the estimation of the remaining points. As one can see, the points obtained from the corrupted entries (in green) are similar to the ones without corrupted entries (in red). The major difference are essentially present in the wrongly detected points, while the others are actually close to the desired position.

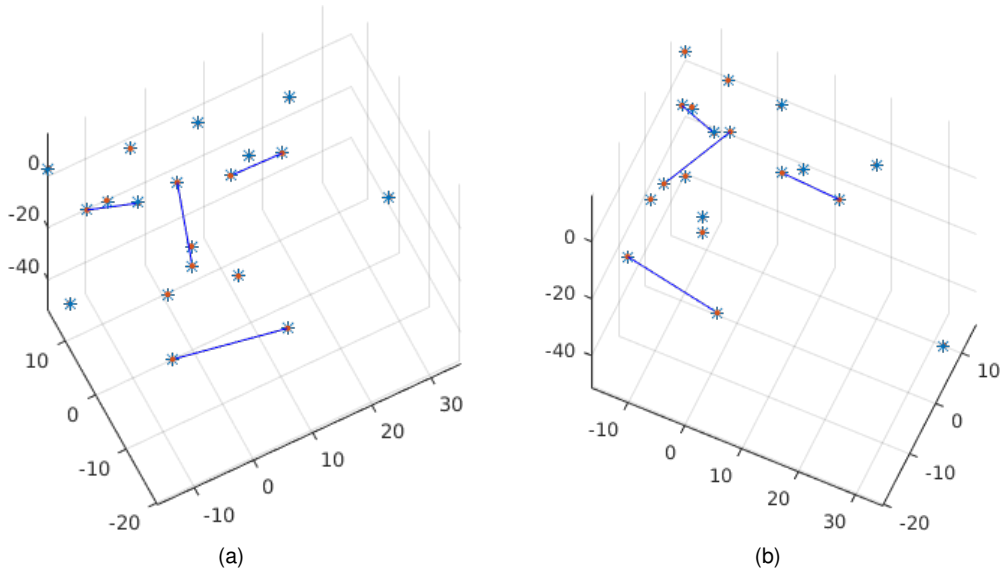


Figure 6.5: Shapes obtained from data with (orange dots) and without corrupted data (blue stars).

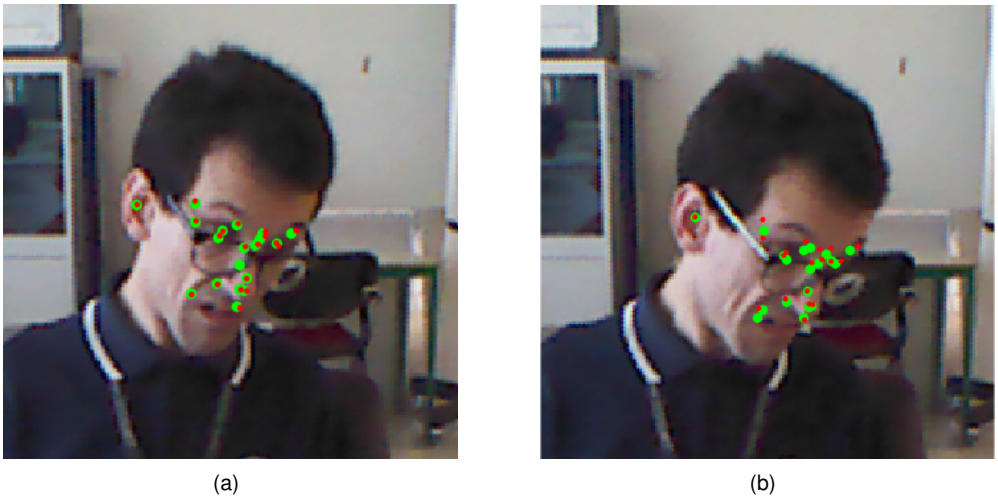


Figure 6.6: Points estimation with (green) and without (red) corrupted entries, at the frames presented in Fig.6.4.

6.1.3 Comparison with Depth sensor

As the Kinect camera has a depth sensor, it is possible to compare our results with the ones actually obtained from Kinect, by overlapping them with face detector output.

In order to do this we simply mapped the points from the facial landmark detector to the depth output. This way we could select the points of the face in the depth image based on the processed RGB output.

As it is possible to see in Fig.6.7, especially in the figures with a top view, Fig.6.7(b) and Fig.6.7(e), the depth camera does not have enough resolution to properly discriminate the small details on the subject's face and body. Actually, it almost looks as if the face itself is completely planar.

On the other hand, as we impose our shape matrix it is possible to see that in this one the depth has a much bigger impact, up to the point that, it seems as if the dimensions in this direction are slightly exaggerated.

This can be explained by the fact that the model behind the point estimation and the creation of the Shape matrix was an orthographic model. So each frame was analysed as a plane and depth was not taken into account. Additionally, in order to properly detect the distances in this axis the subject can't be looking forward, but to one of the sides. However, as it has been discussed previously, the performance of the face detector used decreases in such cases, so we have less data in such scenarios which leads to estimations less accurate.

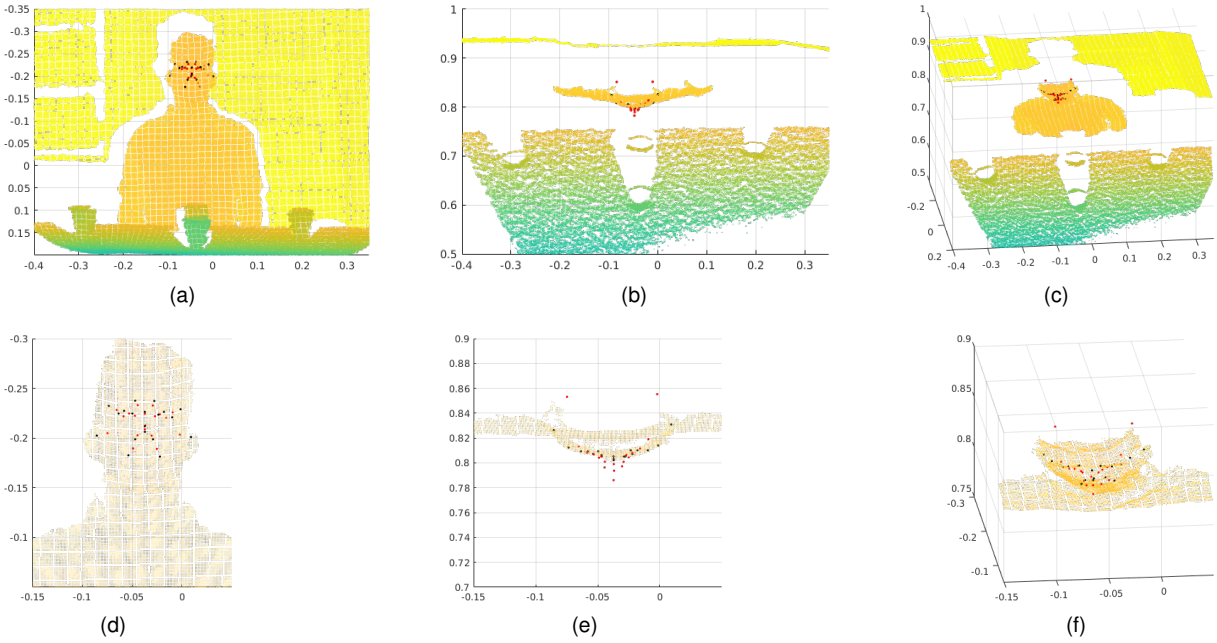


Figure 6.7: Examples of different poses with respective Shape imposed

6.2 Classification

6.2.1 Standardise Shape matrices

After obtaining all the data regarding each subject, then we had to standardise it, as discussed in section 5.1.1. In order to achieve this we had to adjust the shape of each subject onto a standard one, by means of the anisothropic Procrustes [49].

As one would expect, each Shape matrix would be quite dependent on the subject itself as different people have different head structure. However, this was not the only thing that influenced this matrix, as the degree of disability also had its own impact.

This could happen due to the fact that, some people with disabilities end up not being able to move exactly as requested. Therefore, if they could not properly turn their head as requested and get closer to the target, then the data matrix would be lacking this type of frames. So, as we are determining our shape matrix we might not have enough information from different angles to properly estimate it.

In Fig.6.8, a Shape matrix, obtained from one subject who almost could not move their head, imposed onto our reference Shape is presented. As it is possible to see, the shape does not adjust too well to

S_{ref}, due to the fact that this one initially was quite deformed, due to the lack of frames captured from different angles.

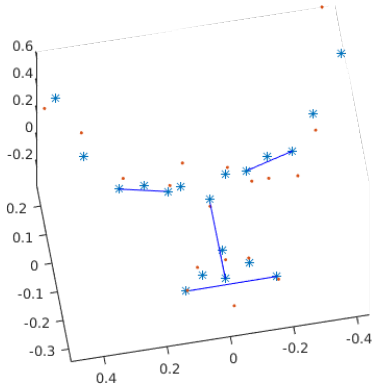


Figure 6.8: S from a subject who was not able to properly turn their head (orange dots), imposed onto S_{ref} (blue stars).

In contrast, subjects without any disabilities had their S better adjusted to S_{ref}. As it is possible to see in Fig.6.9, the points of one of these subjects S (when imposed to S_{ref}) and S_{ref} are quite close as expected, which shows the importance of acquiring data from different angles.

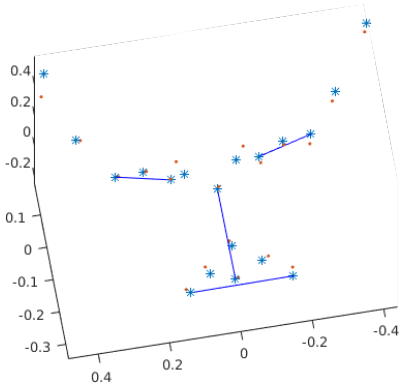


Figure 6.9: S from a subject without disabilities (orange dots), imposed onto S_{ref} (blue stars).

Even though the subjects with some disabilities had their S less well adjusted to S_{ref}, the values of the motion matrix were still used, as we were mainly focused on the orientation of the head throughout the movie, and not on the actual shape of each subject.

6.2.2 Trajectons

In this section, the different trajectons obtained for the different segments will be presented.

More precisely, in both subsections, we analyse the segments where the subjects were asked not to move, move forward and move to the sides.

Additionally, the results shown in this section come from the same subjects during the same frames.

6P Trajectons

As previously explained in subsection 5.2.1, this type of feature creates a discretized cube which gets each entry scored as the points pass on each voxel.

In Fig.6.10, one can compare how one point is represented. More precisely, in this picture, we are tracking one point for 25 frames while the subjects were trying not to move while facing the camera. In Fig.6.10(a) we have a subjected with disabilities and in Fig.6.10(b) another subject without disabilities.

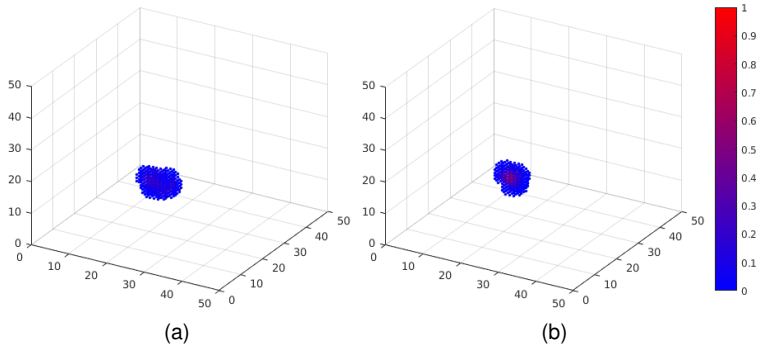


Figure 6.10: One point, corresponding to the tip of the nose, over 25 frames in the 6P trajecton. Performed by (a) a person with disabilities and in (b) without, while they were trying not to move.

As it is possible to see, it is harder for the subject with disabilities to completely restrain the head movements. Firstly, the cloud originated by the position of the point tracked is broader when compared to the second subject. Additionally, one can also look at the intensity of each voxel, by analysing this parameter we can see that in the case of the first subject the voxels on the cloud have low values, which means that the subject did not spend too much time in the same position. However, in the case of the second subject not only the size of the cloud is smaller, but we can also see higher intensity in the centre, which means that stability was more easily achieved by this subjected.

Regarding the full trajecton, with all 6 points, in the case of the movement which consisted on not moving at all, we have similar results. As it is possible to see in Fig.6.11, even before turning the trajectories into the trajecton, we could already see clear differences between the different subjects, as an increasingly control of the movement led to much smaller trajectories. Regarding the trajecton itself, by analysing the ones presented on the image, one can see that by increasing the control over the movement we get point clouds more intense and less broad.

In the case of moving to one of the sides, we can detect different tendencies during the 25 frames registered. In the case of the subject with severe disabilities, shown in Fig.6.12(a), it is clear that the subject is not able to properly reach the object placed on his side and is simply moving forth and back.

On the other hand, the subject with mild disabilities, shown in Fig.6.12(b), is actually able to reach the object, but the trajectory of the points is not as smooth as in the case of the subject with no disabilities, shown in 6.12(c). Actually, the subjects without disabilities were able to get closer to the object much faster than the other subjects, while also rotating less their heads, when compared with subjects with mild physical limitations.

This behaviour can be seen in the 6P Trajecton representation of each subject, which is shown in the

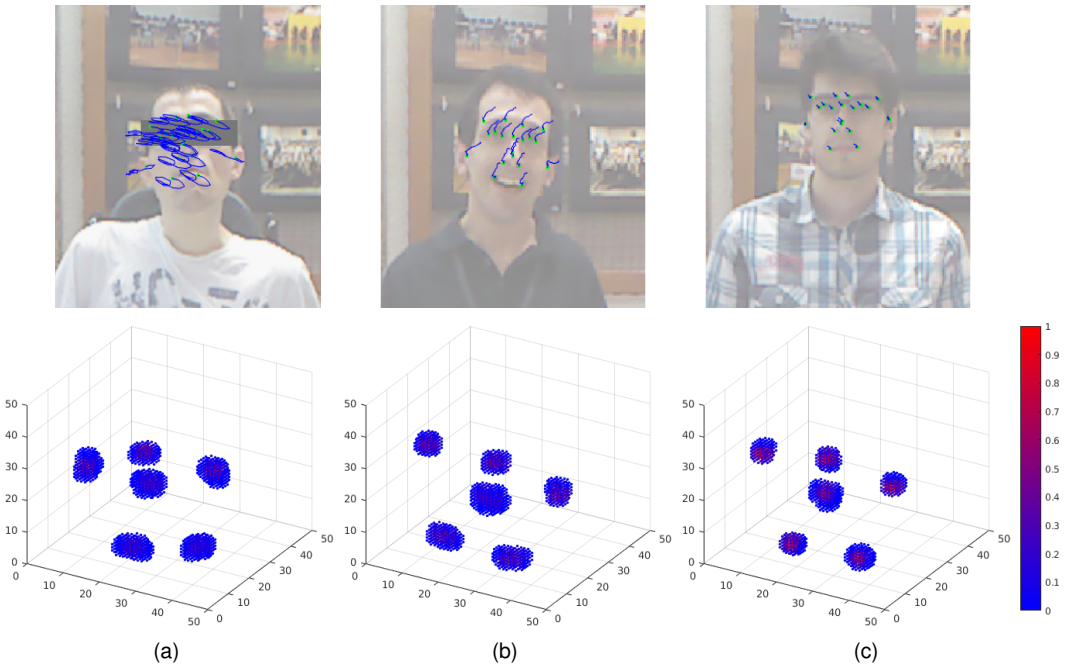


Figure 6.11: Full 6P trajecton, over 25 frames, during the Stable movement, associated with a person with severe (a), mild (b) and no (c) physical limitations

same figure, below each subject, the trajecton obtained when they were moving to the left and to the right. Regarding people with slight disabilities, Fig.6.12(b), we see long trajectories but the increment on the scaling factor is smaller when compared to subjects with no disabilities, Fig.6.12(c). On the other hand, subjects with high physical limitations were not able to move so much, as their movements were highly constrained. Therefore, by analysing Fig.6.12(a), one can clearly see the irregular and restricted movement associated with this subjects.

Another movement analysed consisted on moving forward. Similarly to the previous movement, in this case the subject with severe disabilities, shown in Fig.6.13(a), isn't able to properly approach the object in front of him as instructed. This is shown in the trajecton associated with this subject, which clearly reflects the low movement amplitude and high irregularities.

In contrast, the subject with mild physical limitations, in Fig.6.13(b), can get closer to the object, but while trying to achieve the task we can detect some involuntary movements, characteristic of this type of subjects. In this case, the trajecton presents a movement which is not so fluid and has its amplitude slightly restricted, as one would expect.

Finally, a subject without any type of disabilities is shown in Fig.6.13(c), the trajectory obtained after moving forward is exactly what one would expect. Regarding the trajecton computed, this is quite constant and the changes occur mainly along the scaling factor.

Regarding the images shown, these correspond to the trajectons taken as the movement being performed was halfway done. Therefore, these trajectons enclosed the most important characteristics of each subject while performing a specific movement.

Additionally, the subjects started to do their task in the resting position, so part of our trajectons had both frames with the subject resting and performing the task. With these type of trajecton classifying

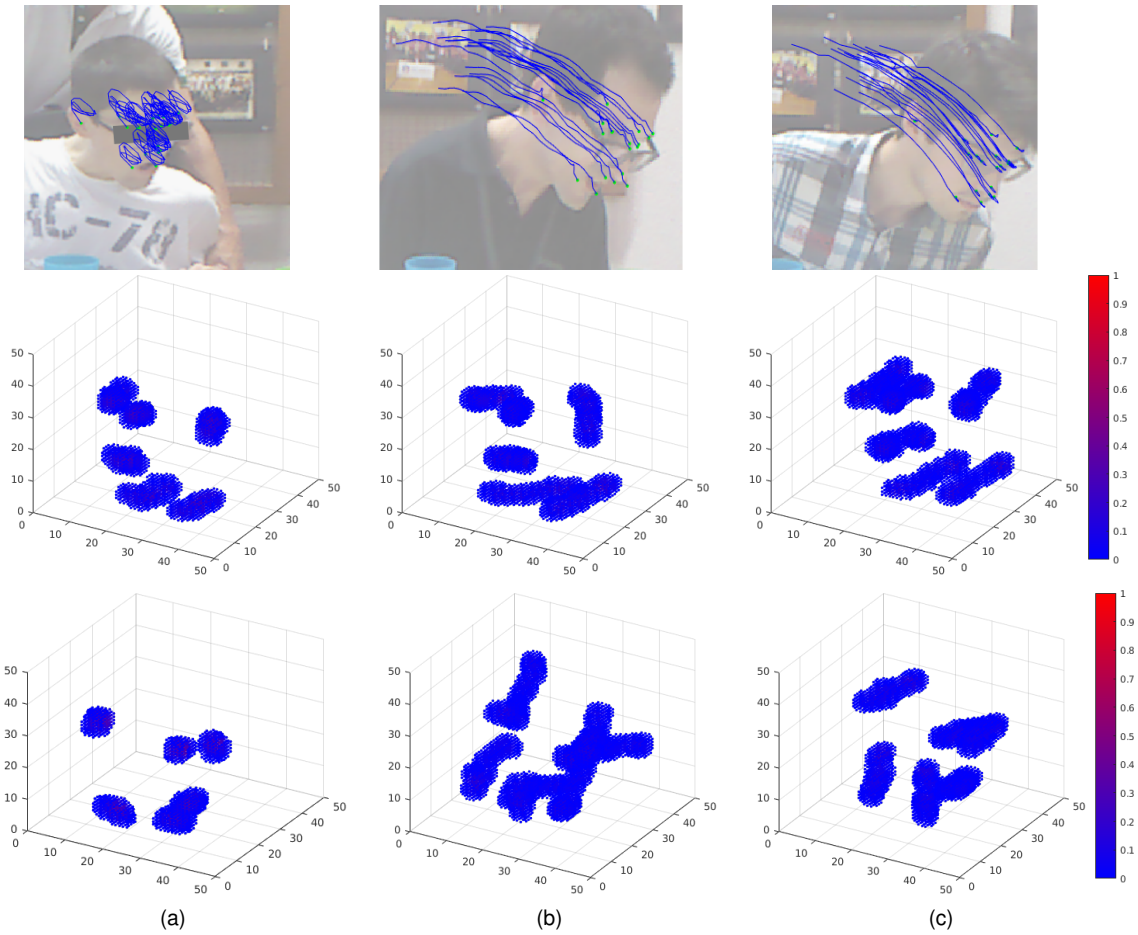


Figure 6.12: Trajectory of the points of each subject while going to one of the sides. Full 6P trajecton, over 25 frames, while moving to the left (second row) and to the right(third row), performed by a person with high (a), mildly (b) and no (c) physical limitations.

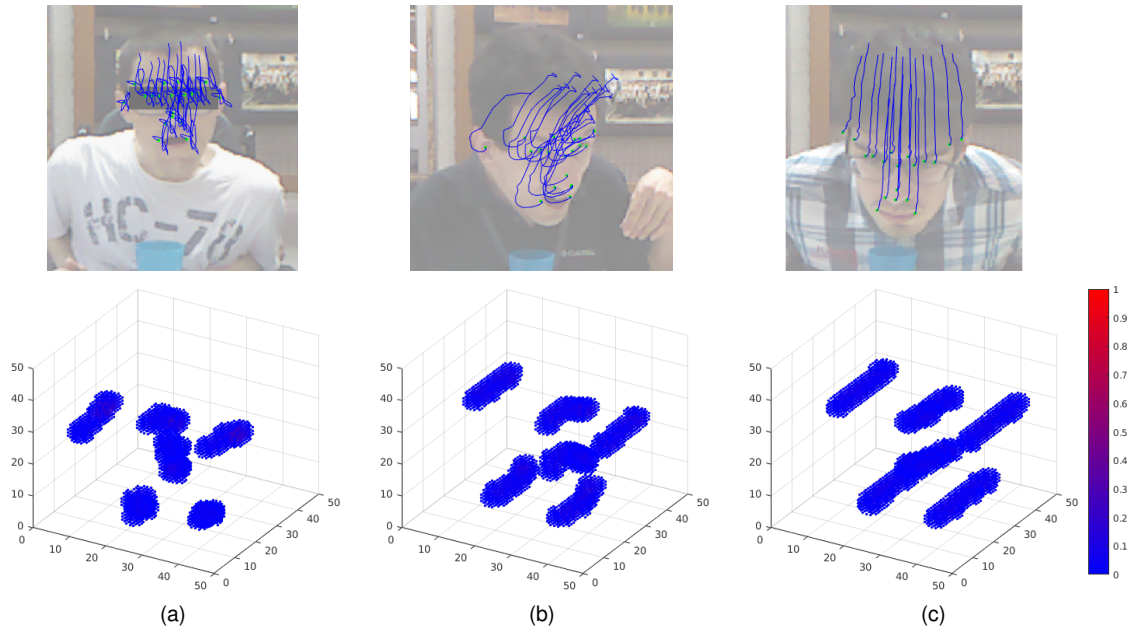


Figure 6.13: Full 6P trajecton, over 25 frames, while moving forward, performed by a person with high (a), mildly (b) and no (c) physical limitations.

the subjects is a bit harder as they have more traits in common. Nevertheless, these were also used to classify all the subjects.

Overall this type of trajecton seemed quite good and the simple fact of being able to discern the different types of subjects just by looking at the feature is quite promising.

Shaky Trajecton

In this second feature, previously explained in subsection 5.2.2, we focus on the changes of the coordinates of each point as the head of the subject rotates.

Firstly, when looking at one point in a single frame we will always have the same result, one big spike and 4 smaller ones. This big spike, corresponds to the entry associated with the type of movement detected between two frames, the remaining 4 match with the closest ones, derived from the Gaussian Box, previously explained. Since, this spike is associated with the movement itself, the shape is always constant and the only difference between different movements will be in the actual position of this spike. In Fig.6.14, one can see the output of one point which is not moving.

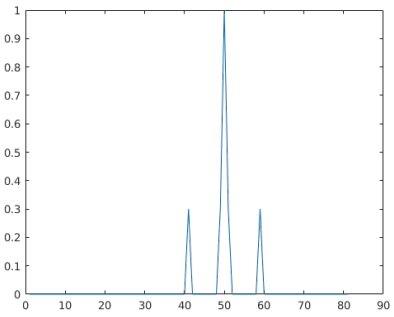


Figure 6.14: Example of the representation of a single point of a single frame.

Actually, for each frame we actually analyse 17 points, as well as the scale factor between two frames. Therefore, we will end up having more data to evaluate for every frame, which can be seen in Fig.6.15.

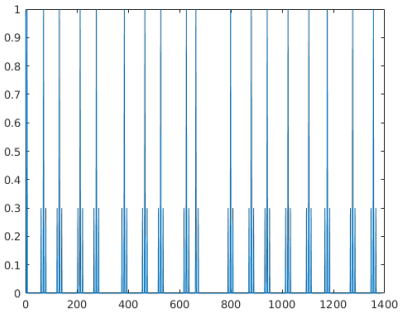


Figure 6.15: Example of the representation of all the points under analyse, of a single frame. All have the same intensity, but they are differently spaced, which means that different points are moving at different rates

Besides looking at all the points we also include the influence of the different frames. In Fig.6.16, one can see an example of the behaviour of one point along a block of 25 frames.

As we put all the information obtained for a block of frames together, we can then obtain our trajecton.

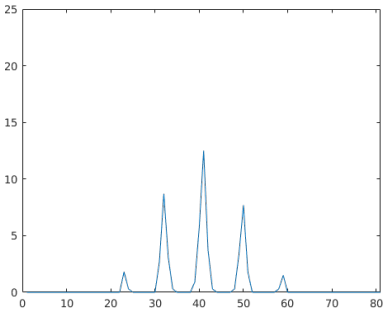


Figure 6.16: Example of the representation of one the point, throughout 25 frames.

Regarding this trajecton, as we analyse the resting position it is possible to predict in some way the resulting trajectons for the different subjects. In the case of the subjects with no disabilities, Fig.6.17(c), as they can better control their movements then they will stay immobile more easily. Therefore, this type of subjects would have bigger spikes since the output would be always constant, as can be seen in Fig.6.17(b). As for subjects with some physical limitations, these were not able to restrict as easily their movements. So, as they try to stay immobile, the unavoidable shacking of the head leads to a trajecton with more entries different than 0, and these had a lower amplitude. Such pattern can easily be seen in Fig.6.17(a).

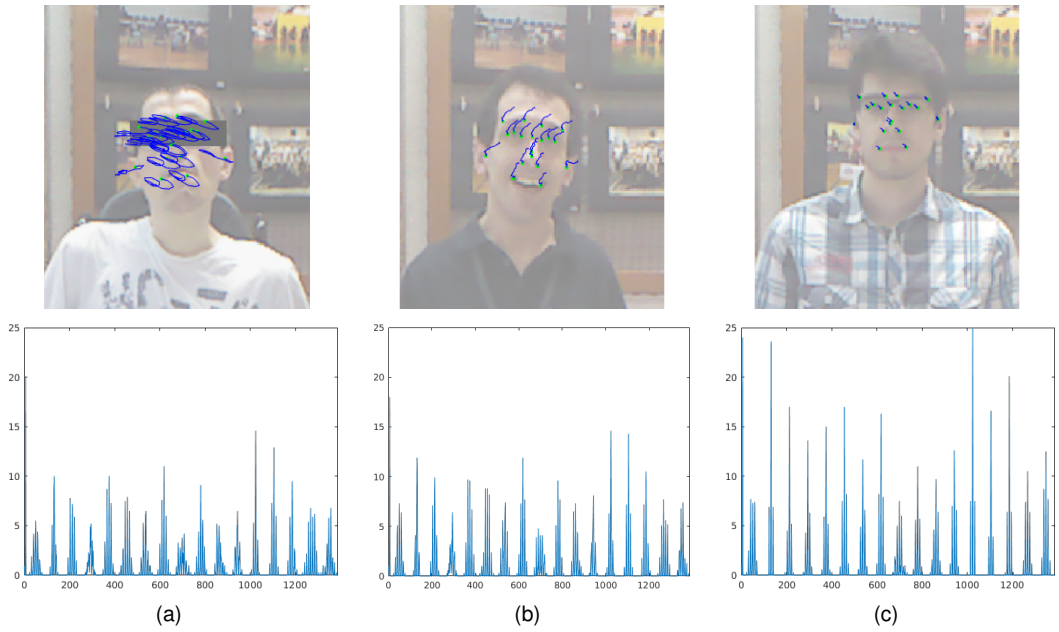


Figure 6.17: Example of a Shaky trajecton as a subject with (a) and without (b) disabilities tries not to move.

Actually, in Fig.6.18 another way to represent these trajectons is shown. In Fig.6.18(a), the actual ideal trajecton for someone not moving at all is presented. Regarding the remaining ones, Fig.6.18(c) and Fig.6.18(b), we can see the output provided by a subject without and with disabilities, respectively. In this new configuration, we are simply storing the directions where each point in each frame translates to. Each square is associated with moving in one direction in respect to the centre one, which corresponds to being stable.

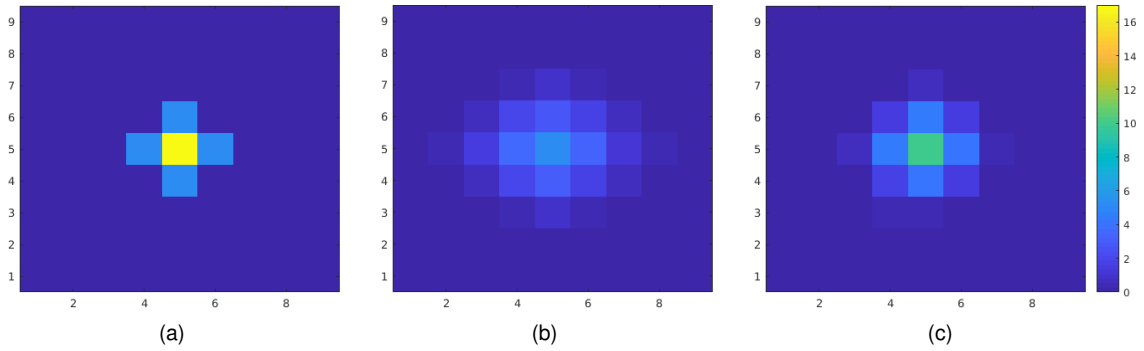


Figure 6.18: Representation of the ideal Shaky trajecton during the resting position (a). Example of a Shaky trajecton as a subject with (a) and without (b) disabilities tries not to perform any movement.

As the subjects should not be moving the translation should be constant and nil, as shown in the ideal case. This type of configuration only makes sense for the resting position, as during the remaining movements we are expecting changes on the points positions, which would not be possible to analyse using this type of visualisation.

Regarding the other types of movements, it is harder to analyse the trajecton obtained, since we are expecting movement to happen, and due to the dimensions of the problem it is not easy to evaluate this trajecton.

Intuitively, one would expect that independently of the type of movement, subjects without any type of physical disabilities would have their movements more fluid. This fact would imply that their points trajectories would be constant along consecutive frames. Therefore, as the block of frames integrated in each trajecton consists on 25 consecutive frames, then it is expected that this type of subjects would have associated bigger spikes, than subjects who cannot control their movements as well. Additionally, we can also add, that this spikes aren't necessarily associated with the constant position, as we can have the points moving at a constant velocity.

In Fig.6.19, it is shown the resulting trajecton for the three type of subjects as they move to the right and forward. According to the previous statement, we can see the size of the spikes increasing as the physical limitations decrease.

Regarding this type of trajecton, now, time was not such a serious problem as dealing with it was much faster, due to the fact that the dimension of this new trajecton was much lower. In contrast this trajecton had another problem intrinsically associated with the way it was designed. More precisely, the problem relied on the fact that sometimes the tracker would not follow the points as reliably as we wanted, which led to slight variations of the output. In the case of the 6P trajecton this was not a big problem, as we were storing the relative positions of the points along all frames, so in the end we would have a cloud around the true coordinates of these points. In contrast, this features stores the changes of the points coordinates during the movement. Therefore, if these points have slight destabilization in their estimation, then we will detect more false positives which are stored and seen as the subject moving their head, even though that may not be the case.

Nevertheless, as one can see by the information presented here and by the results obtained during

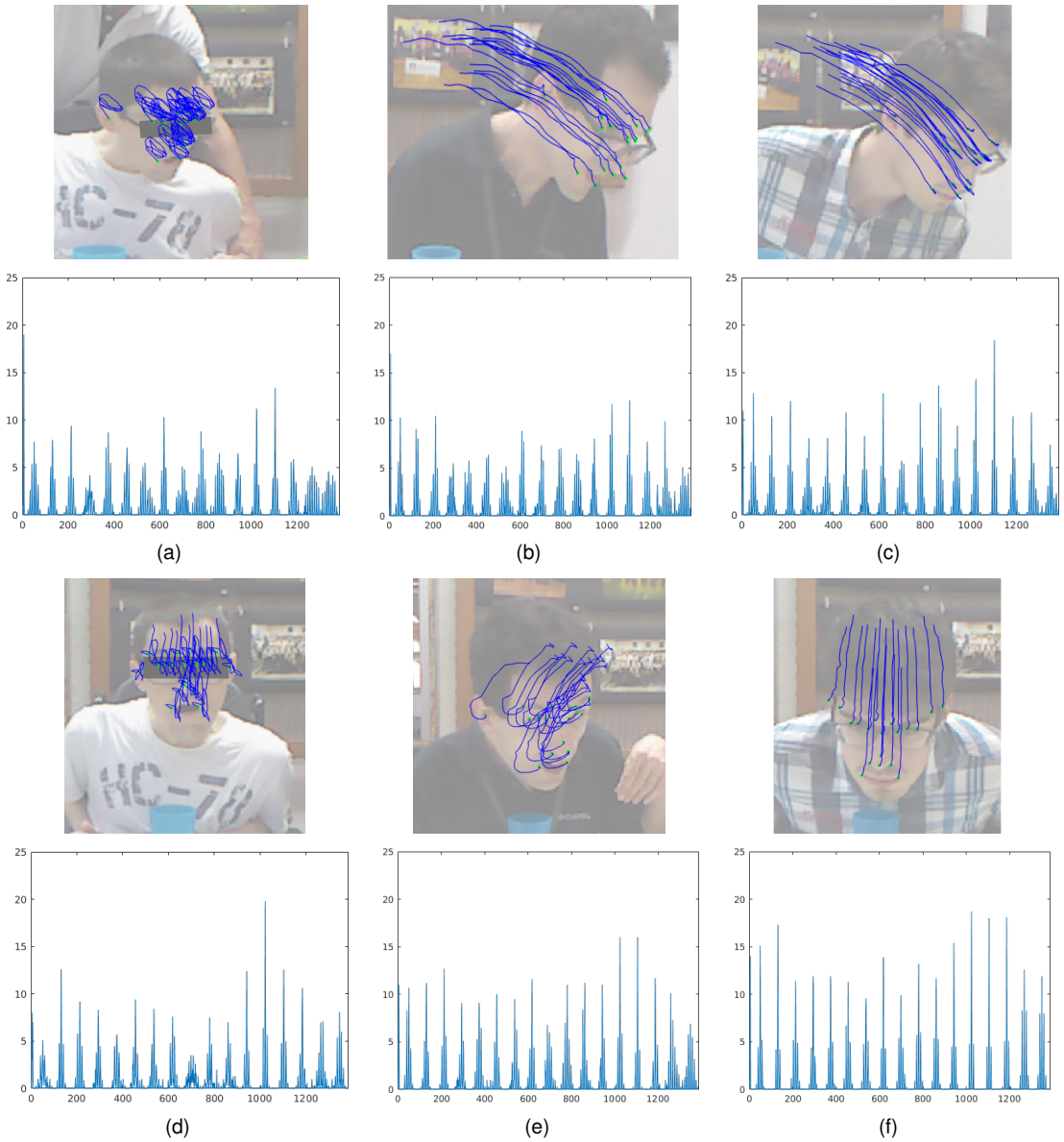


Figure 6.19: Full Shaky trajecton, over 25 frames, while moving to the right (a)(b)(c) and forward(d)(e)(f), performed by a person with high (a)(d), mildly (b)(e) and no (c)(f) physical limitations.

the classification stage, we can conclude that these trajecton was good enough to achieve the proposed task.

6.2.3 Classification

In order to classify the subjects, first it was necessary to create the codewords. These were determined by applying the k-means clustering algorithm to all the trajectons obtained in each type of segment.

Actually, the trajectons used were not all the possible ones. During the sliding window method, explained in 5.2.1, a step of 5 was used. Therefore, only one out of every 5 possible trajectons was used. This way we wouldn't have so many trajectons almost equal to each other.

During this phase a different number of clusters were tested, but only those from 4 to 6 clusters are presented

After creating the codewords these were grouped together and later used to classify future new subjects given.

6P Trajecton

In order to create the clusters, it was imperative to choose the number of clusters we needed.

In order to make this decision it was tested the elbow method in order to determine the optimal number of clusters. In order to apply this method, it was necessary to cluster the data with a varying number of clusters and plot the sum of squared errors for each value. Then, by analysing this plot it would be possible to detect the optimal k where a further increment of this value would not, in theory, improve substantially the performance.

However, as one can be seen in Fig.6.20, the decreasing of performance only happens at high values of k, more precisely when using around 10 clusters.

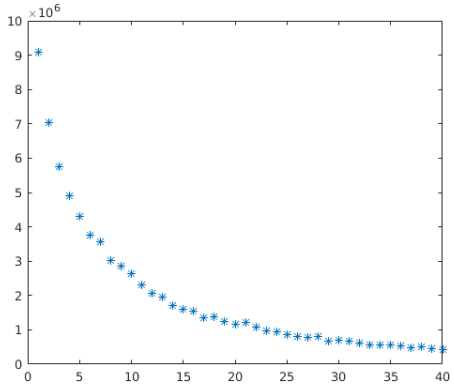


Figure 6.20: Plot of the error associated with 2 to 40 clusters for the Stable movement, in order to apply the elbow method.

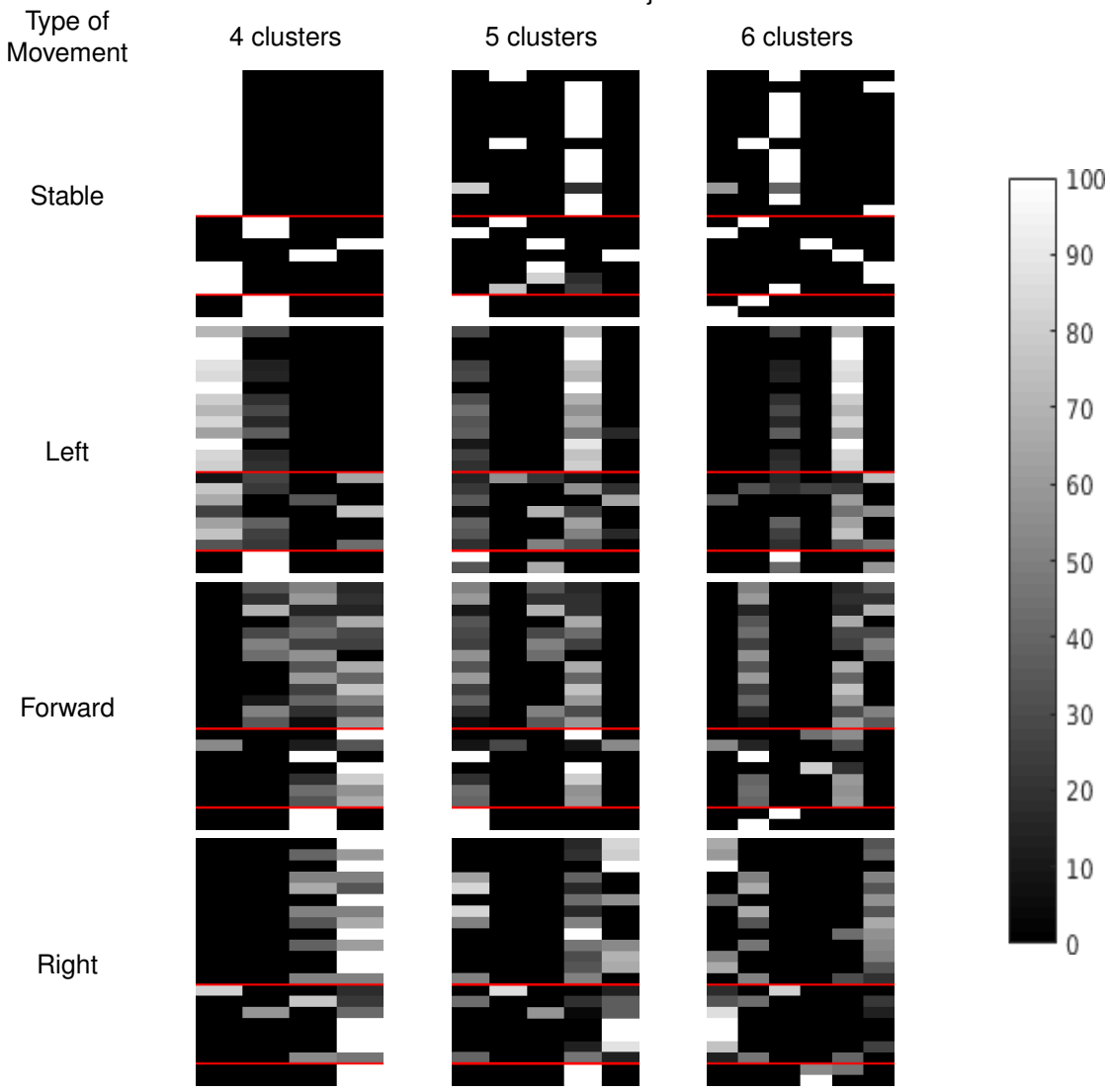
Given the number of subjects in our data set, using such a high value in the umber of clusters would most likely separate the subjects among themselves. So the idea of clustering subjects based on their performance would not be so clear.

Therefore, a lower number of clusters was used, so as to better discern the similarities and differences between the several subjects.

In Table 6.1, it is shown the results after clustering our data, for different types of movement, with different values of k, $\{4, 5, 6\}$. In each row of the table one type of movement is analysed and in each column a different number of clusters is used. Then on each cell of the table it is shown an image representing the distribution of the subject's trajectons, while they were doing a certain movement, over a defined number of clusters. More precisely, in the images, each line corresponds to a subject histogram and each column with a cluster. Brighter colour corresponds to a higher percentage of that subject's trajectons associated with a certain cluster. The first subjects (until first red line) don't have any disability, second group of subjects have some physical disabilities and in the last one are the subjects with severe physical disabilities

As it is possible to see, the results after using different number of clusters lead to slightly different combinations of clusters. Nevertheless, the subjects who share the same type of behaviour, most of the

Table 6.1: Clusters obtained from 6P trajecton



times, are grouped together under the same cluster and occupy a different one, when compared with subjects with different degree of disabilities.

For instance, in the case of the Stable movement, independently of the number of clusters used, there is one cluster which mainly expresses all the individuals without disabilities, while the remaining clusters better represent the subjects with disabilities. Therefore, we can conclude that almost all the subjects without disabilities behave the same way, and this behaviour is captured by one of the clusters.

This means that the simple aggregation of similar trajectons leads to the formation of clusters with semantic. In other words, we are not simply aggregating the similar trajectons, but actually differentiating the movements based on their performance. Consequently, if we know how to classify one trajecton we can already infer with some confidence which type of subject we are analysing. This is only possible, since this feature can express reasonably well the differences between the different types of subjects.

However, sometimes this classification divides the subjects in a way that someone one would not expect given their label.

In some cases, subjects without disabilities might get associated with other groups. Such case can

be seen in the clustering with 5 and 6 seeds while the subjects were Stable, where we have two subject completely out of the expected cluster. Actually, the reason behind this misclassification relies on the subject performance and not on the classifier. These subjects were asked not to move, but due to exterior influences they were not so focused on the task at hand, as can be seen in Fig.6.21. So, instead of trying to be immobile, they were actually moving like an individual with disabilities. Therefore, even though they did not have any physical limitation, they were classified as having it.

Figure 6.21: Trajectory of the points during the stable movement of a subject without disabilities, which was classified as a subject with physical limitations

Just like the first group can be associated with the other ones, the same can happen the other way around. Actually, regarding the subjects with disabilities analysed these were quite different between them. No one had the same type of physical limitation, so their behaviour would be quite inconsistent. Therefore, while we were grouping them as a certain type of subjects, depending on the task asked to perform, they would show different behaviours, in some cases even similar to subjects without any physical limitation.

For instance, the last subject with mild during the Stable movement, independently of the number of clusters used, this subject would have, at least partly of its trajectons associated with subjects without disabilities. This can be explained by the fact that this subject had a high control over her movements, so she was able to stay immobile, as shown in Fig.6.22, just like we would expect from a subject without any physical limitation.

Figure 6.22: Trajectory of the points during the stable movement of a subject with disabilities, which was classified as a subject without physical limitations.

Therefore, even though the label might indicate one type of subject, since the method used was non supervised, it is able to classify how the subjects behaved during the recordings. So, it detects whether the given movement was associated with the group of subjects without disability or not, just as intended.

Below in Fig.6.23, the clusters obtained for the Stable movement with 4 clusters are shown. Their order is in the same as in the scheme shown on the first cell of Table 6.1. Therefore, the first cluster is associated with the individuals without any type of disabilities and the remaining clusters encode the subjects with physical limitations.

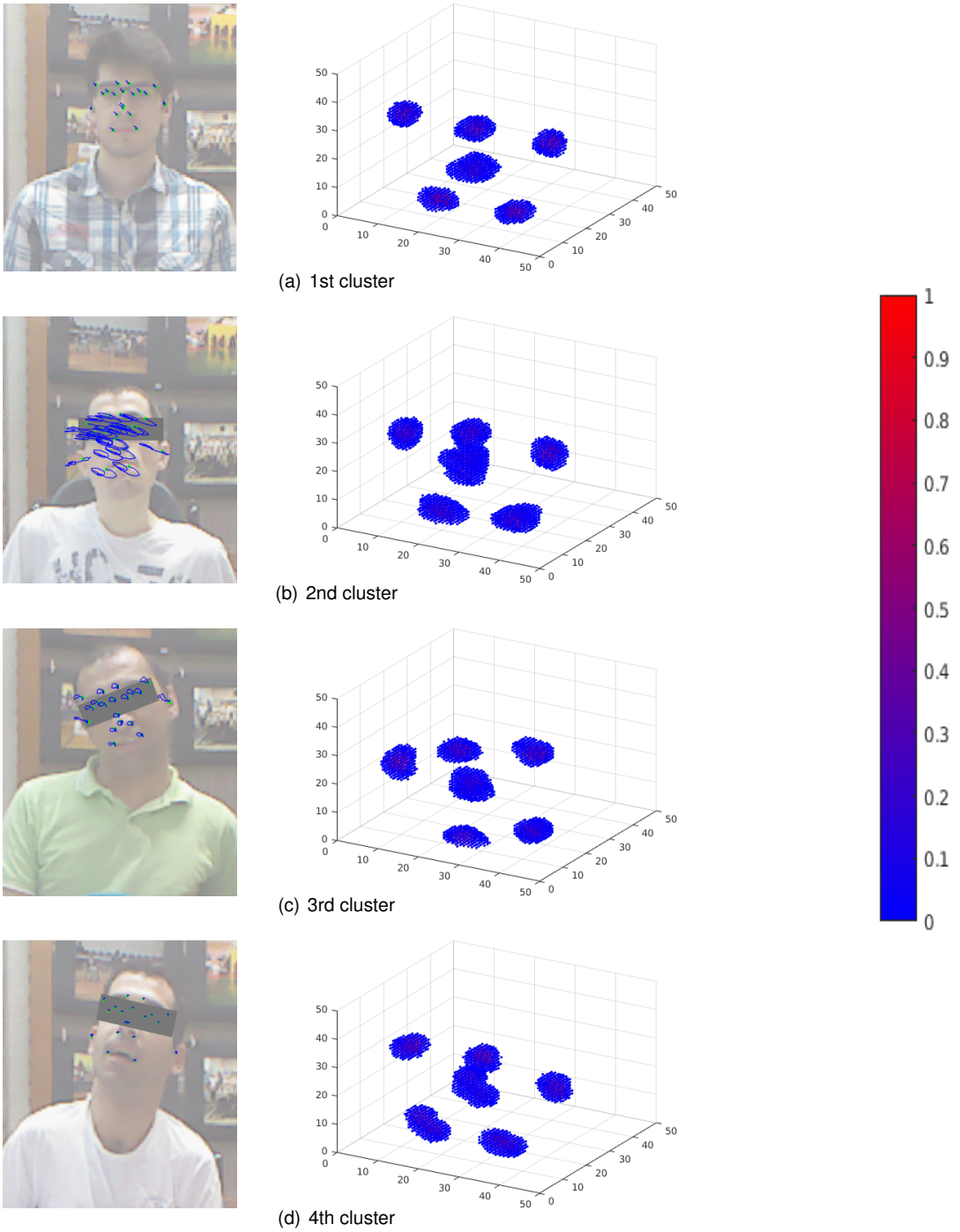


Figure 6.23: Clusters of the 6P trajectons obtained from k-means algorithm with the Stable data while using 4 seeds.

As it is possible to see in Fig.6.23, the representation of the clusters describes quite well our data. Regarding the first cluster, if we analyse the result from Table 6.1, we can conclude that this one is associated with subjects without any physical limitations. The second cluster, as one can see, encloses

the subjects who were not able to stabilize their head.

In the third cluster, we can see that the subject associated is able to stabilize his head reasonably well. However, the 6P trajecton focus more on the pose of the head, and since this subject has his head pending to one of the sides, then he actually led to the creation for trajectons with this type of pose.

Regarding the other types of movements, one can see that, most of the times, there are two clusters enclosing the different subjects. This can be explained due to the fact that during these movements there were different phases. For instance, in the moving forward movement we start with the head in the resting position. Then we need to rotate our head down and start moving towards the object. As the 6P trajecton is focusing on the pose of the head, during this phase, we will have trajectons with a mix of poses, as the subject looks forward and starts rotating the head towards the object. As the movement develops the remaining trajectons will have a constant pose with a varying scale factor, as the subject gets closer to the camera while looking at the object.

In contrast, the subjects with severe disabilities were not able to move as well as the remaining subjects. Therefore, due to this lack of manoeuvrability, there were no apparent phases while performing the movement, when compared with the other subjects. Therefore, instead of having their trajectons spread along the clusters, they are concentrated on one of the clusters. When they try to perform the movement there is little development, there are no clear phases to discern.

Below in Fig.6.24, the clusters obtained for the Forward movement with 4 clusters are shown. Their order is in the same as in the scheme shown on the first cell of Table 6.1.

By analysing the figures obtained it is possible to see the different phases of the movement. In the Third cluster cluster, it is possible to see a big cloud in the area with small scale factor and then a thin trail. By comparing this feature with one of the frames associated with this subject we can clearly see that this cluster corresponds to the initial phase of the movement, when the subject is in the resting position and then suddenly moves forward.

Since this cluster encodes the initial phase of the movement, then it also led to aggregating the subjects with severe disabilities. This happened due to the fact that these type of subjects could not move properly, as explained before, so they would not be able to move forward, which would simulate an initial phase of the movement during all the trajectons. Therefore, this subjects would only be associated with this cluster and never with the ones associated with the other parts of the movement.

The following phase of the movement is depicted in the fourth cluster. In this one we can see a big trail caused by the different trajectories performed by the several subjects. Since we are using a small amount of clusters, we only have a sufficient amount to describe the different phases of the movement. Therefore, the several subjects, while moving forward are represented here, which explains the degree of the size of the cloud. In order to better differentiate the performance of the different subjects along this movement, a bigger number of clusters would be required.

The final phase of the movement can be seen in the second cluster. This cluster encodes the moment when the subjects were reaching the furthest point, as well as the frames where they could not go any further. Therefore, this cluster presents us the faint traces of the final part of the trajectory, as well as the final position where they spent some time.

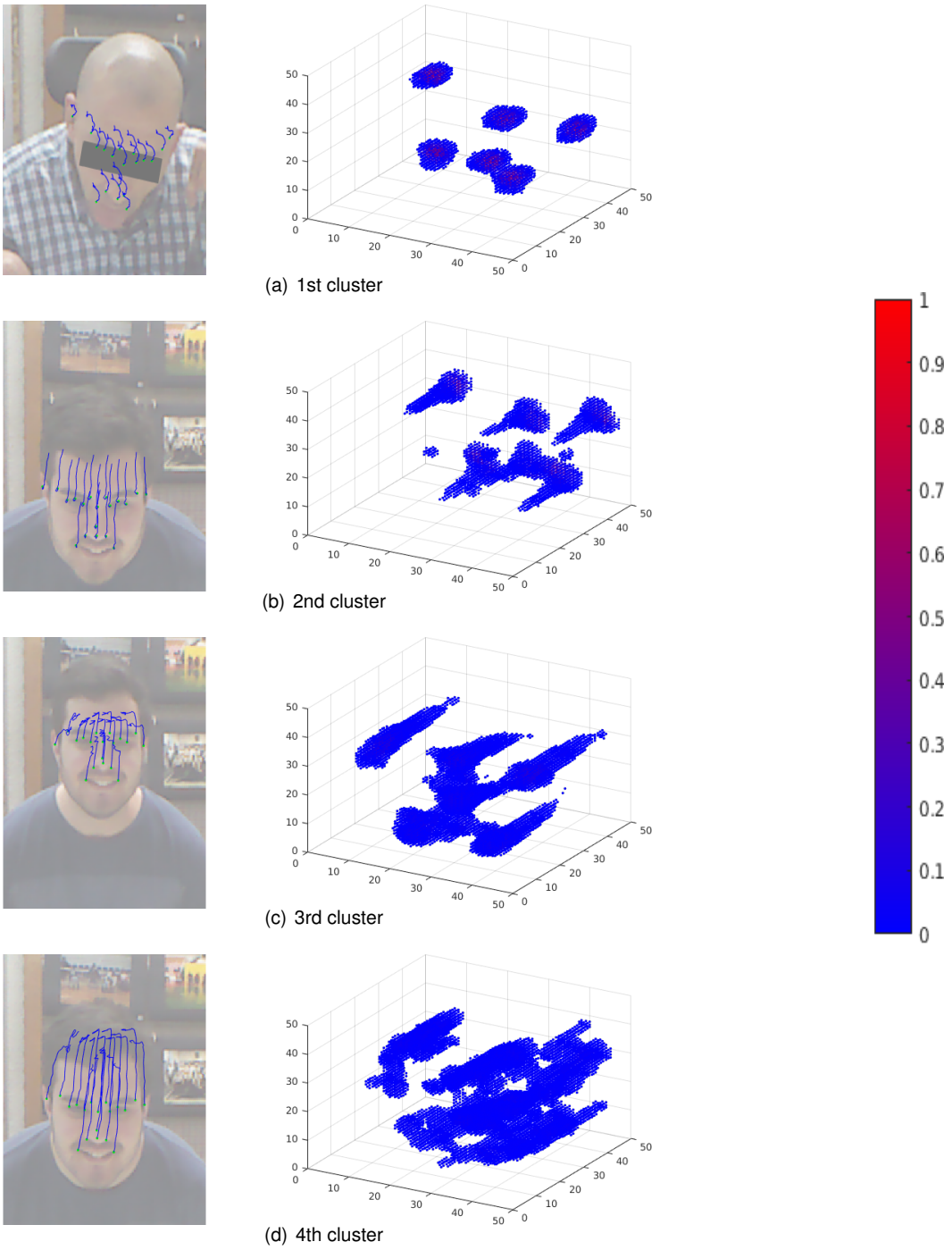


Figure 6.24: Clusters of the 6P trajectons obtained from k-means algorithm with the Forward data while using 4 seeds.

The first cluster was formed by the trajectons of one subject who reached the object in front of him, but then spent too much time in that position creating then a faint cluster dedicated only to his points.

Overall, this trajecton presented a great tool to encode the subjects and detect the way they perform different movements. Based on this performance, we were able to detecting whether the subject behaved like a subject without disabilities or not. In the feeding robot scenario this type of information would be extremely helpful, more than knowing if the subject actually had any type of disability. Since, it would allow to adapt the different states of the feeding process to the subjects themselves and their

ability to perform different tasks.

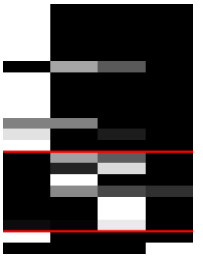
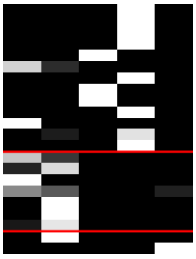
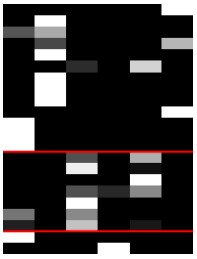
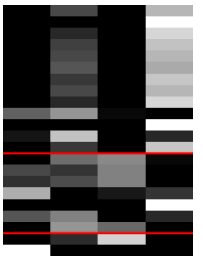
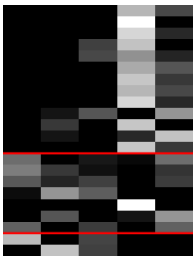
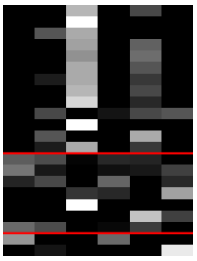
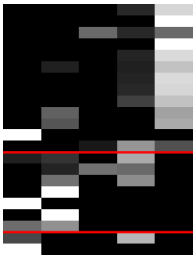
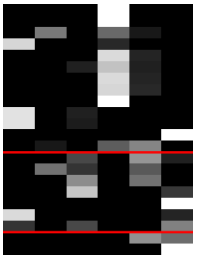
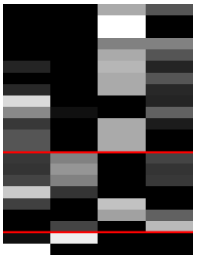
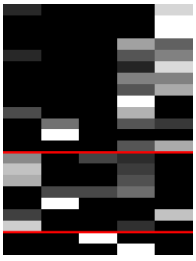
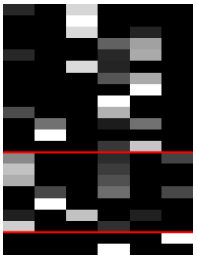
One might also point out the fact that our data is unbalanced. However, that is not the case, since even though we have more patients without disabilities, these type of people would move faster than the ones with physical limitations. Therefore, we might have a higher number of subjects with disabilities, but the number of trajectons of each type (with and without disabilities) is quite similar.

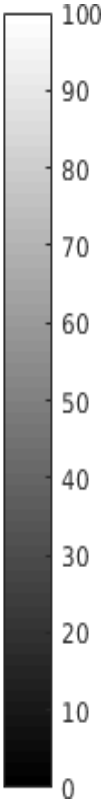
Shaky Trajecton

Similarly to the 6P trajecton, the same procedure was applied to the Shaky trajecton.

In Table 6.2, we have a table similar to the one presented in the previous section. The same movements and number of clusters are shown, the only difference is the type of trajectons clustered.

Table 6.2: Clusters obtained from Shaky trajecton

Type of Movement	4 clusters	5 clusters	6 clusters
Stable			
Left			
Forward			
Right			



Overall the results obtained are similar to the ones shown in the 6P trajecton. We can detect some clusters predominantly associated with subjects without disabilities and other clusters with motor impaired subjects.

Regarding incorrect labelling, one may point out one interesting subject. This was one of the subjects previously defined as having severe disabilities, because the amplitude of her movement was almost nill. Since she could not move and had great control over her movements, as shown in Fig.6.25, one would expect her to classified as a subject without disabilities, during the Stable movement. However, in the 6P trajecton that was not the case, since her posture was not as straight as the remaining individuals without disabilities. Nevertheless, the Shaky feature focus on the rate of change of the points of the face, so this trajecton was able to identify her pattern as one of those from a subject without disabilities, as can be seen in Table.6.2 (she corresponds to the first subject of the ones with severe disabilities).

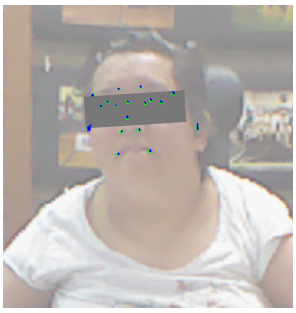


Figure 6.25: Trajectory of subject with severe disability, while not moving.

So, in this trajecton we are able to detect again the subjects whose behaviour is similar to to the one common in the subjects without disabilities. Therefore, this trajecton also achieves the proposed objective as it is able to differentiate the subjects based on their performance on each movement.

Like in the 6P trajecton, it is possible to show again the different clusters formed for each type of movement. However, as one was able to see in previous examples, the output of this trajecton is not so intuitive, so it is harder to derive conclusions from them. In Fig.6.26, we show again the output of 4 clusters obtained as the subjects were not moving.

As it is possible to see, the first cluster was associated with subjects without disabilities and as we would expect, it has almost all points with high peaks. This means that almost all points along the major part of the frames have exactly the same type of constant behaviour, in this case they are almost always not moving.

On the other hand, subjects with disabilities were not able to maintain their face immobile so well. Therefore the clusters associated to this type of subjects had peaks slightly smaller. One can see this pattern in the second and third cluster, where the subjects with disabilities were predominantly assigned depending on their behaviour.

The last cluster was associated to only one subject who had severe difficulties. Unlike the one previously described, this subjected had extreme difficulties controlling his movement, as can been seen in Fig.6.26. His behaviour was so irregular that the just by simply clustering our data we are already able to separate this subject from the remaining others.

Regarding other movements, in Fig.6.27, the same analysis is done to the Forward movement. The different clusters obtained when using 4 seeds for the k-means algorithm and an example of a trajectory belonging to these clusters is shown.

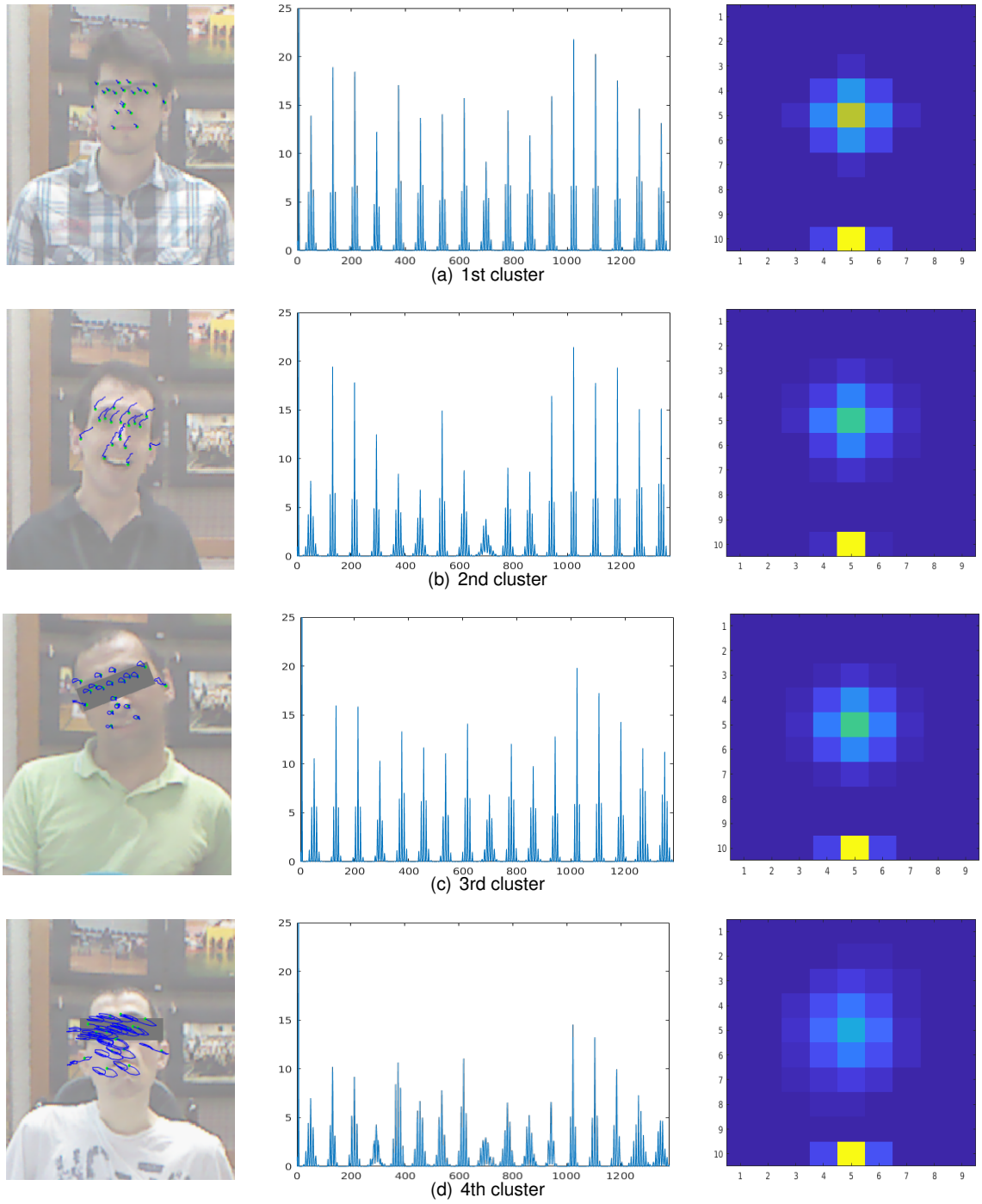


Figure 6.26: Clusters of the Shaky trajectons obtained from k-means algorithm with the Stable data while using 4 seeds. Last image corresponds to an alternative representation of the trajecton, where the rate of change of the coordinates of all points are combined together, as well as the change of the scale factor (depicted in the last line)

In this new movement the third cluster is associated with the subjects without disabilities. As we can see, the movement associated with this individuals is clear an direct. When we analyse the trajecton it is possible to see high values in the rate of change of the scale factor, which means the subject is approaching the camera reasonably fast while rotating the head towards the object.

On the other hand, in cluster four it is presented a large cluster which encoded most of the subjects

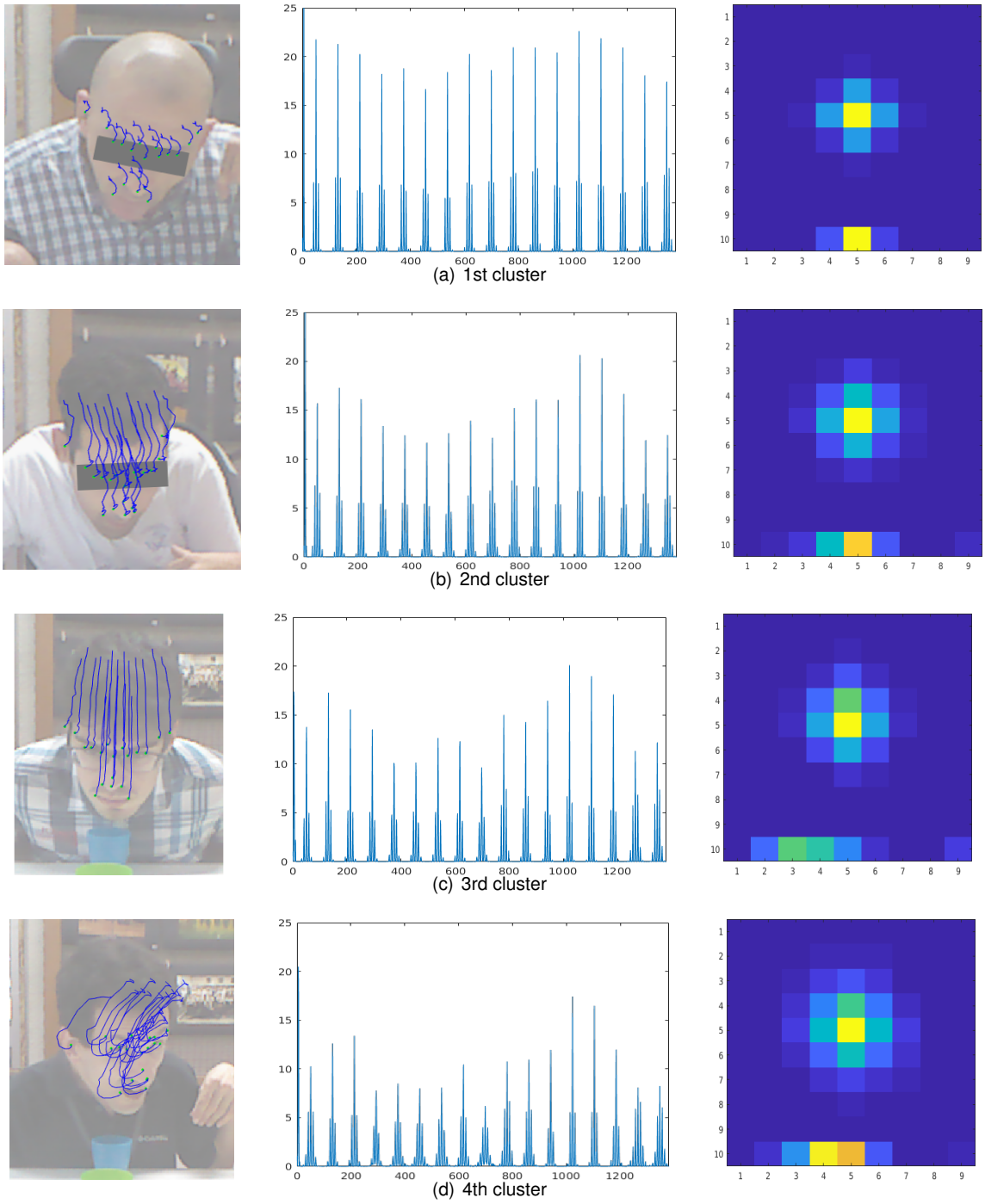


Figure 6.27: Clusters of the Shaky trajectons obtained from k-means algorithm with the Forward data while using 4 seeds.

without disabilities. Compared to the previous cluster, we can see that the movement is less fluid, the subjects shake more the head while performing the given task. Additionally, we can also confirm that the movement is also performed at a slower rate, since the values of the rate of change of the scale factor are closer to 0. This might be due to the fact that subjects without disabilities don't have such constraints over their movements and, as they have higher control over their movement, are more confident on the outcome of their actions. Another explanation, might be due to the fact that as subjects move, their actions are all in accordance with the final outcome. In contrast, subjects with disabilities might

try to achieve the given task, but due to their physical limitations they require more time and auxiliary movements to perform it.

The first cluster is associated with the final part of the movement. Similarly, to the last movement, this one also had a cluster associated with the last trajectons of the movement. More precisely, this trajectons would correspond to the ones when the subject had finished the movement and was standing near the objective, without getting any closer nor moving significantly the head.

Finally, the first cluster is more closely associated with a subject who could move his head but was not able to properly get closer to the object in front of him. Therefore, he rotated his head according to the desired movement, but then he was not able to actually lean forward. Therefore, there was no clear increment of the scale factor, as the distance of the subject to the camera did not decrease substantially.

6.2.4 Classification

As the clusters have been defined, it is now possible to classify new subjects. In order to simulate this step, we simply divided our data set into a training and a test set.

This way, we were able to create clusters based on our training set and then classify the ones from the test set based on this new clusters. However, before doing this classification, the subjects were labelled based on their performance. So, for each movement each subject was analysed and it was decided, whether the subject moved as a subject with disabilities or without disabilities, since this was what we were actually trying to define. For instance, the subject presented in Fig.6.25, was able to stay immobile but could not perform the other movements, so she was labelled as a subject without physical limitations relatively to the first movement, but the remaining ones labelled as a subject with disabilities.

In order to create these two sets 2 subjects without disabilities and 2 subjects with disabilities were chosen at random and formed the test set. The remaining ones were associated with the training set. Then, the clusters of the test set would be formed, which would provide an output similar to the ones presented in the previous section.

After testing 20 combinations of test sets, on average these subjects would be classified accordingly 65.3% of the times with the 6P trajecton and 66.5% with the Shaky trajecton, with the respective confusion matrices shown in Tables 6.3 and 6.4. In those matrices, WD and WoD stands for ”With disability” and ”Without disability”, respectively.

Table 6.3: Confusion matrix 6P trajecton with full subject removal

	True WoD	True WD
Predicted WoD	399	249
Predicted WD	52	268
Draw	32	72

Table 6.4: Confusion matrix Shaky trajecton with full subject removal

	True WoD	True WD
Predicted WoD	400	251
Predicted WD	61	218
Draw	16	20

As it is possible to see most of the misclassification fall on classifying subjects as one without disabilities, when in fact that was not the case.

There are several reasons which might be behind this misclassification. One of the most important reason relies on the lack of data. Since, in our data set each subject had different degrees of disability, then, when removing these subjects from the training data, we might not have another subject with a similar behaviour on this set. Consequently, there was no cluster close to their data as there was no subject with a similar behaviour, and the classification of these subjects would end up not being realiable. As we can see, most of the misclassifications fall on assigning the label on person without disability to the ones which show behaviour influenced by it. The other way around does not happen as much since the number of subjects with that pattern was much bigger when compared to the large variety of patterns associated with people with disabilities as each one moves almost in their own way.

Another proof that supports this argument, relies on the fact that running our k-means algorithm several times leads to slight different results. This happens due to the fact that we don't have enough data to properly represent the clusters we are trying to determine. Moreover, by increasing the amount of data it would allow to have more subjects would help to break the ties in the classification.

Additionally, this results might also be biased by the original classification, as the labels were not done by a professional of the area.

This problem could be solved by increasing our data set. More specifically,by getting more data from different subjects with similar disabilities.

This problem was simulated again, but now instead of completely removing the subjects of the test set from the training set, only a random part of their movement was removed. This way both sets would have parts of the same subjects, but these parts would be associated with different frames, so that there was no information repeated on both sets.

In order to test these partitions, another 20 tests, with different combinations subjects and parts removed were tested. Since we had in this setup a subject with the same type of disability, the outcome was better, more precisely, on average these subjects would be classified accordingly 71.5% of the times with the 6P trajecton and 93.1% with the Shaky trajecton. Similar to the previous example, the respective confusion matrices are shown in Tables 6.5 and 6.6. In those matrices, WD and WoD stands for "With disability" and "Without disability", respectively.In this case the same classification was applied, each subject was classified depending whether they were able to perform each movement as a subject without any disability.

Table 6.5: Confusion matrix 6P trajecton with partial subject removal

	True WoD	True WD
Predicted WoD	90	45
Predicted WD	18	68
Draw	9	10

Table 6.6: Confusion matrix Shaky trajecton with partial subject removal

	True WoD	True WD
Predicted WoD	112	11
Predicted WD	5	103
Draw	3	6

As one can see, by having similar patterns in the training set greatly improves the classification of the new trajectons, which supports the idea that increasing the data set would lead to a better performance.

Additionally, it was also tested the distance matrices between the different subjects, one for each trajecton. These matrices consisted on the distances between the different subjects, based on the histograms presented in Table 6.1 and Table 6.2 for the 6P and Shaky trajecton, respectively. These matrices are shown below in Fig.6.28.

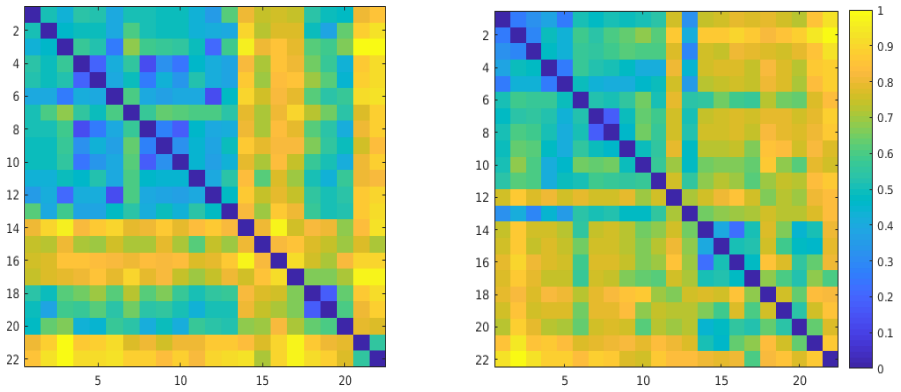


Figure 6.28: Distance matrices of the 6P (left) and Shaky (right) trajecton. These measure the distance between the histograms of the different subjects, according to the results shown in the tables of the previous section.

As one can see the trajectons presented can group together the several subjects without disabilities. However, the other type of individuals are not as close to one another as one would expect. This divergence also supports the idea that the data set requires more data, as there were no subjects with the same type of disabilities being grouped together.

Chapter 7

Conclusions and Future Work

In this thesis we focused on creating a methodology aiming to classify different subjects based on their disabilities.

As there was no type of data set available, one had to be constructed for this project. This data set consisted on labelled recordings of different subjects (with and without disabilities). These recordings were performed with a specific setup and the subjects were requested to do pre-defined movements.

In order to classify our subjects, two different types of features were developed, which were able to describe the trajectory of the subjects while they were performing different tasks. One of them focused more on the spatial information obtained through the recordings, and the other on the temporal information.

These features were estimated based on the 2D projection of the trajectory of facial landmarks. As the position of these important points were detected on each frame, they were stored onto a data matrix. However, there was a significant number of missing entries in this data matrix. This happened due to the fact that the current facial landmark detectors would fail when facing occlusions or were simply not accurate enough. Therefore, a large part of this work consisted on estimating the values of these missing entries, by assuming there was some structure behind this data. Overall, the data obtained by the proposed methods provided a good estimation of the points' of interest position.

After acquiring this data matrix, it was possible to encode it into our trajectons. As this step was completed, our new data was grouped using k-means clustering. Actually, some of the clusters obtained using this approach had already some meaning associated. There were some which were particular of subjects without any type of disability and other clusters correlated with motor impaired subjects.

Finally, we were able to classify our subjects based on the nearest neighbour algorithm, which was able to provide reasonable results. However, due to the low data quantity and the variability of the disabilities of the subjects analysed, the results could not be further improved.

Future Work

Overall, one of the greatest drawbacks in this project was the lack of quantitative baselines. To the extent of our knowledge no work facing this problem has been done before, so there were no benchmarks, nor ground truth, nor data sets available to properly compare our work with. Therefore, not only

we didn't have enough subjects with similar problems to better group them, but also most of the decisions had to be done based on a qualitative analysis, which might have led to possibly biased results. Nevertheless, the features presented in this thesis were quite promising. However, there is still room for improvement. More precisely, increasing the amount of subjects is imperative, as well as improving their labels.

Additionally, along this project several parameters had to be tuned in order to achieve better performance, but some of them may be further improved. Furthermore, it would also be beneficial to lose these small parameters and focus on an even more robust procedure.

Moreover, in order to increase the accuracy, a better camera could also be used. This way the points acquired in the initial phase and the ones tracked using the Lucas-Kanade algorithm would provide a result more reliable and with less error associated.

Other than classifying the subjects under study based on their motion patterns, one of he most direct applications to the work developed in this thesis consists of detecting intention. This could be achieved by continuously analysing the motion of the subject, and by comparing with other known ones, we could predict the intentions of the subject based on his movement.

Another interesting application could be an online classification of the subject. For instance in the feeding robot scenario, initially the subject can reach the cutlery with the food by himself. As he gets tired, the amplitude of his movements decrease, so ideally the system should detect this changes in performance and adapt the control of the robotic arm, so as to move according to the subject's capacities.

The methodology developed along this thesis is quite flexible, it does not necessarily need to be applied to the setup proposed. Actually, the same idea can be applied in other scenarios, in order to analyse other type of motion patterns.

For instance, still related with these subjects, a similar method could be used by physiotherapists in order to classify the patients based on their behaviour in different tasks.

Bibliography

[1] World Health Organization (WHO) - Disabilities. URL http://www.who.int/topics/disabilities/en/.

[2] WHO. World Report on Disability - Summary. *World Report on Disability 2011*, (WHO/NMH/VIP/11.01):1–23, 2011. ISSN 1353-8047.

[3] M. Oskoui, F. Coutinho, J. Dykeman, N. Jetté, and T. Pringsheim. An update on the prevalence of cerebral palsy: a systematic review and meta-analysis. *Developmental Medicine & Child Neurology*, 55(6):509–519. doi: 10.1111/dmcn.12080. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/dmcn.12080.

[4] S. Gulati and V. Sondhi. Cerebral Palsy: An Overview. *The Indian Journal of Pediatrics*, nov 2017. ISSN 0973-7693. doi: 10.1007/s12098-017-2475-1.

[5] W. L. Minear. A Classification of Cerabral Palsy. *Pediatrics*, 18(5):841–852, 1956. ISSN 0031-4005.

[6] C. Morris and D. Bartlett. Gross motor function classification system: impact and utility. *Developmental Medicine & Child Neurology*, 46(1):60–65. doi: 10.1111/j.1469-8749.2004.tb00436.x.

[7] D. Jeevanantham, E. Dyszuk, and D. Bartlett. The Manual Ability Classification System: A Scoping Review. *Pediatric Physical Therapy*, 27(3):236–241, 2015. ISSN 1538005X. doi: 10.1097/PEP.0000000000000151.

[8] L. Tschirren, S. Bauer, C. Hanser, P. Marsico, D. Sellers, and H. J. A. Hedel. The Eating and Drinking Ability Classification System: concurrent validity and reliability in children with cerebral palsy. *Developmental Medicine & Child Neurology*, 60(6):611–617. doi: 10.1111/dmcn.13751.

[9] Institute of Medicine. *The Second Fifty Years: Promoting Health and Preventing Disability*. The National Academies Press, Washington, DC, 1992. ISBN 978-0-309-04681-7. doi: 10.17226/1578.

[10] M. E. Mlinac and M. C. Feng. Assessment of activities of daily living, self-care, and independence. *Archives of Clinical Neuropsychology*, 31(6):506–516, 2016. doi: 10.1093/arclin/acw049.

[11] M. Topping. Early experience in the use of the 'Handy 1' robotic aid to eating. *Robotica*, 11(06): 525, 11 1993. ISSN 0263-5747. doi: 10.1017/S0263574700019366. URL http://www.journals.cambridge.org/abstract_S0263574700019366.

[12] Obi — Robotic feeding device designed for home care. URL https://meetobi.com/.

[13] C. J. Perera, T. D. Lalitharatne, and K. Kiguchi. EEG-controlled meal assistance robot with camera-based automatic mouth position tracking and mouth open detection. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1760–1765. IEEE, 5 2017. ISBN 9781509046331. doi: 10.1109/ICRA.2017.7989208. URL http://ieeexplore.ieee.org/document/7989208/.

[14] V. Petridis, B. Deb, and V. Syrris. Detection and identification of human actions using predictive modular neural networks. In *2009 17th Mediterranean Conference on Control and Automation*, pages 406–411, June 2009. doi: 10.1109/MED.2009.5164575.

[15] H. Zhou, L. Wang, and D. Suter. Human motion recognition using gaussian processes classification. In *2008 19th International Conference on Pattern Recognition*, pages 1–4, Dec 2008. doi: 10.1109/ICPR.2008.4761140.

[16] M. Shweta, K. Yewale, M. Pankaj, and K. Bharne. ARTIFICIAL NEURAL NETWORK APPROACH FOR HAND GESTURE RECOGNITION. 2018.

[17] S. Yokota, H. Hashimoto, Y. Ohyama, J. She, D. Chugo, and H. Kobayashi. Classification of body motion for human body motion interface. In *3rd International Conference on Human System Interaction*, pages 734–738, May 2010. doi: 10.1109/HSI.2010.5514486.

[18] J. Lee, J. Han, X. Li, and H. Gonzalez. Traclass: Trajectory classification using hierarchical region based and trajectory based clustering. *Proceedings of the VLDB Endowment*, 1(1):1081–1094, 1 2008. ISSN 2150-8097. doi: 10.14778/1453856.1453972.

[19] F. I. Bashir, A. A. Khokhar, and D. Schonfeld. Object trajectory-based activity classification and recognition using hidden markov models. *IEEE Transactions on Image Processing*, 16(7):1912–1919, July 2007. ISSN 1057-7149. doi: 10.1109/TIP.2007.898960.

[20] V. Syrris and V. Petridis. Statistical descriptors for human actions classification. In *2009 17th Mediterranean Conference on Control and Automation*, pages 412–415, June 2009. doi: 10.1109/MED.2009.5164576.

[21] J. Owens and A. Hunter. Application of the self-organising map to trajectory classification. In *Proceedings Third IEEE International Workshop on Visual Surveillance*, pages 77–83, July 2000. doi: 10.1109/VS.2000.856860.

[22] X. Xiao, H. Hu, and W. Wang. Trajectories-based motion neighborhood feature for human action recognition. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 4147–4151, Sept 2017. doi: 10.1109/ICIP.2017.8297063.

[23] A. Boubezoul, A. Koita, and D. Daucher. Vehicle trajectories classification using support vectors machines for failure trajectory prediction. In *2009 International Conference on Advances in Compu-*

tational Tools for Engineering Applications, pages 486–491, July 2009. doi: 10.1109/ACTEA.2009.5227873.

[24] M. Isaloo and Z. Azimifar. Anomaly detection on traffic videos based on trajectory simplification. In *2013 8th Iranian Conference on Machine Vision and Image Processing (MVIP)*, pages 200–203, Sept 2013. doi: 10.1109/IranianMVIP.2013.6779978.

[25] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana. Fast time series classification using numerosity reduction. In *In ICML'06*, pages 1033–1040, 2006.

[26] M. Mctear, Z. Callejas, and D. Griol. *The Conversational Interface*. 2016.

[27] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 524–531 vol. 2, June 2005. doi: 10.1109/CVPR.2005.16.

[28] Y. Zhou and T. S. Huang. 'bag of segments' for motion trajectory analysis. In *2008 15th IEEE International Conference on Image Processing*, pages 757–760, Oct 2008. doi: 10.1109/ICIP.2008.4711865.

[29] C. Li and F. Yang. Cluster-based dictionary learning and locality-constrained sparse reconstruction for trajectory classification. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1248–1252, March 2016. doi: 10.1109/ICASSP.2016.7471876.

[30] P. Viola and M. Jones. Robust real-time object detection. 2001.

[31] MIT — Technology Review. URL `https://www.technologyreview.com/s/535201/the-face-detection-algorithm-set-to-revolutionize-image-search/`.

[32] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct 2016. ISSN 1070-9908. doi: 10.1109/LSP.2016.2603342.

[33] T. Baltrušaitis, P. Robinson, and L. Morency. Openface: An open source facial behavior analysis toolkit. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10, March 2016. doi: 10.1109/WACV.2016.7477553.

[34] T. Baltrusaitis, P. Robinson, and L. Morency. Constrained local neural fields for robust facial landmark detection in the wild. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 354–361, Dec 2013. doi: 10.1109/ICCVW.2013.54.

[35] A. Zadeh, T. Baltrušaitis, and L.-P. Morency. Convolutional Experts Network for Facial Landmark Detection. pages 2519–2528, 2017. ISSN 21607516. doi: 10.1109/CVPRW.2017.256.

[36] OpenPose — OpenPose Output. URL `https://github.com/CMU-Perceptual-Computing-Lab/OpenPose/blob/master/doc/output.md`.

[37] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. 2017.

[38] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. 2016.

[39] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. 2017.

[40] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L. Morency. Openface 2.0: Facial behavior analysis toolkit. pages 59–66, May 2018. doi: 10.1109/FG.2018.00019.

[41] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004. ISSN 09205691. doi: 10.1023/B:VISI.0000011205.11775.fd.

[42] R. F. C. Guerreiro and P. M. Q. Aguiar. Estimation of rank deficient matrices from partial observations: two-step iterative algorithms, energy min. In *meth., Computer Vision and Pattern Recognition, Lecture Notes in Computer Science 2683*. Springer-Verlag, 2003.

[43] A. Eriksson and A. van den Hengel. Efficient computation of robust low-rank matrix approximations in the presence of missing data using the l lt;inf gt;1 lt;/inf gt; norm. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 771–778, June 2010. doi: 10.1109/CVPR.2010.5540139.

[44] K. Fragkiadaki, M. Salas, P. Arbeláez, and J. Malik. Grouping-based low-rank trajectory completion and 3d reconstruction. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, pages 55–63, Cambridge, MA, USA, 2014. MIT Press. URL `http://dl.acm.org/citation.cfm?id=2968826.2968833`.

[45] M. Marques and J. Costeira. Estimating 3D shape from degenerate sequences with missing data. *Computer Vision and Image Understanding*, 113(2):261–272, 2009. ISSN 10773142. doi: 10.1016/j.cviu.2008.09.004.

[46] K. Zhao and Z. Zhang. Successively alternate least square for low-rank matrix factorization with bounded missing data. *Computer Vision and Image Understanding*, 114(10):1084–1096, 2010. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2010.07.003. URL `http://www.sciencedirect.com/science/article/pii/S1077314210001505`.

[47] C. Tomasi and T. Kanade. Shape and motion from image streams: a factorization method. *Proceedings of the National Academy of Sciences*, 90(21):9795–9802, 1993. ISSN 0027-8424. doi: 10.1073/pnas.90.21.9795.

[48] C. J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):206–218, March 1997. ISSN 0162-8828. doi: 10.1109/34.584098.

[49] J. C. Gower and G. B. Dijksterhuis. *Procrustes Problems*. 2004. ISBN 0 19 851058 6 10 98765 4321.

[50] P. Matikainen, M. Hebert, and R. Sukthankar. Trajectons: Action Recognition Through the Motion Analysis of Tracked Features. *IEEE 12th International Conference on Computer Vision Workshops*, 2009.

Appendix A

Details of Tomasi-Kanade Algorithm

According to section 4.3, after computing the SVD of our data matrix we were left with $\hat{M}$ and $\hat{S}$.

Both $\hat{M}$ and $\hat{S}$ share the same dimensions as M and S. However, these might not be the best approximation as the decomposition is not unique. In other words, if we have any 3x3 invertible matrix, Q, we get other valid decomposition of $\hat{W}_c$ by multiplying M by Q and S by Q^{-1}, since:

$$(\hat{M}Q)(Q^{-1}\hat{S}) = \hat{M}(QQ^{-1})\hat{S} = \hat{M}\hat{S} = \hat{W}_c \tag{A.1}$$

Therefore it is necessary to constraint the final result. More precisely we should find the matrix Q that respects the following equality:

$$M = \hat{M}Q \tag{A.2}$$

$$S = Q^{-1}\hat{S} \tag{A.3}$$

One good way to find the final result relies on the rotational matrices' properties. Since, M should encode a rotational matrix, so $\hat{M}Q$ should also be the same. Therefore, we need to enforce that and try to find the best Q which allows the preservation of those properties. Which all comes down to solving the following quadratic system presented in equation (A.4).

$$\begin{aligned} i^f Q Q^T i^f &= 1 \\ j^f Q Q^T j^f &= 1 \\ i^f Q Q^T j^f &= 0 \end{aligned} \tag{A.4}$$

Having solved this system we are then left with the final result of S and M.

Appendix B

Reformulation of the Anisothropic Procrustes problem

In this appendix we reformulate the minimisation problem shown in equation (5.1) and simplify the expressions present in Algorithm 1.

Regarding the minimisation problem we had to change the formulation of the cost function, as shown in the following equalities.

$$\begin{aligned} RDS_{ref} &= S \\ RDS_{ref}S'_{ref} &= SS'_{ref} \\ \mathbb{I}RD &= SS'_{ref}(S_{ref}S'_{ref})^{-1} \end{aligned}$$

As it is possible to see the problem we have at hand now is in essence the same, but now we are trying to fit a 3×3 identity matrix, $\mathbb{I}$, onto $SS'_{ref}(S_{ref}S'_{ref})^{-1}$. In order to avoid confusion and simplify reading, $\mathbb{I}$ will be referred as X_1 and the expression $SS'_{ref}(S_{ref}S'_{ref})^{-1}$ will be replaced by X_2. Therefore, we arrive at the minimisation problem shown in equation (5.2).

Regarding the Algorithm 1, first we needed to initialise our variables. In case of D, this matrix was defined as a 3×3 identity matrix. Regarding R, this matrix was initialised by solving the common Procrustes problem. So, by applying the Procrustes method presented in section 4.4, we found the rotational matrix that would best overlap both X_1 onto X_2.

Since we are dealing with the identity matrix, X_1, and rotational matrix, R, we could use their basic properties, shown below, to simplify our problem.

$$\begin{aligned} RR^T &= \mathbb{I} \\ \mathbb{I}A &= A \\ \mathbb{I}^T\mathbb{I} &= \mathbb{I} \end{aligned} \tag{B.1}$$

As it was shown in Algorithm 1, D was updated using the following equation:

$$D = (\mathbb{I} \circ (R'X_1'X_2))(\mathbb{I} \circ (R'X_1'X_1R))^{-1} \tag{B.2}$$

Due to the the properties shown in the equations (B.1), then the update of D could be rewritten as:

$$D = \mathbb{I} \circ (R'X_2) \tag{B.3}$$

One can emphasise the fact that by multiplying element wise both the identity matrix with any other matrix A, in essence we are selecting the diagonal entries of A and leaving the remaining ones as 0.

In order to update R, an auxiliary matrix, Z, had to be used.

$$Z = D(X_2'X_1 + D'R'(\mu\mathbb{I} - X_1'X_1)) \tag{B.4}$$

As it is possible to see, the expression presented in (B.4) can also be simplified, due to the properties of the matrices involved.

$$Z = DX_2' \tag{B.5}$$

From this new matrix, Z, it was possible to determine its left (U) and right (V) singular vectors and update R by properly multiplying the two.