TOWARDS AN AUTOMATIC INTELLIGIBLE MONITORING OF BEHAVIORAL AND PHYSIOLOGICAL METRICS OF USER EXPERIENCE: HEAD POSE ESTIMATION AND FACIAL EXPRESSION RECOGNITION

by

KALLIATAKIS GRIGORIOS

B.A, Technological Educational Institute of Crete, 2013

A THESIS

submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

DEPARTMENT OF APPLIED INFORMATICS
AND MULTIMEDIA
SCHOOL OF APPLIED TECHNOLOGY
TECHNOLOGICAL EDUCATIONAL INSTITUTE OF CRETE

· 2015 ·

Approved by:
Major Professor
Vidakis Nikolaos

Abstract

In this thesis a model for building, visualizing and ultimately assessing behavioral and physiological metrics of user experience in terms of head pose changes and facial expression variations is presented. The motivation of this work is to produce comprehensible visual representations of two different sets of data, which are acquired using an affordable 3D sensing technology (Microsoft Kinect sensor), with a view to raise the cognitive level in terms of analyzing user metrics. Both head pose estimation and facial expression recognition have attracted a great deal of interest in literature due to recent advances in computer vision, human computer interfaces and human activity recognition systems. To that end, an approach build on discriminative random regression forests was followed in order to achieve fast, accurate and reliable estimation of head pose in uncontrolled environment In addition to that, emotion recognition via facial expressions (ERFE) was adopted in order to complete the process of recognizing four main expressions including happiness, anger, sadness and surprise. For this reason, the features of animation units (AUs), tracked by the Kinect sensor, are exploited. A lightweight data exchange format (JavaScript Object Notation-JSON) is utilized for collecting and storing the data from the aforementioned sets. Such mechanism can yield a platform for objective and effortless assessment of user behavioral actions and physiological attitude within the context of different applications, such as game development and evaluation.

Research is to see what everybody else has seen, and to think what nobody else has thought. …

Albert Szent-Gyorgyi

Table of Contents

List of Figures

List of Tables

List of Abbreviations

DOF	Degrees of Freedom
SVR	Support Vector Regressors
HRI	Human Robot Interaction
RBF	Radial Basis Function
AAP	Active Appearance Models
PCA	Principal Component Analysis
LSD	Local Slice Depth
LSO	Local Slice Orientation
DRRF	Discriminative Random Regression Forests
JSON	JavaScript Object Notation
XML	Extensible Markup Language
FER	Facial Expression Recognition
FERA	Facial Expression Recognition and Analysis
HCI	Human Computer Interaction
AUs	Action Units
ERFE	Emotion Recognition via Facial Expression
SVM	Support Vector Machine
SURF	Speeded Up Robust Features
NN	Neural Networks
k-NN	k-Nearest Neighbors
NB	Naïve Bayes
DMCMs	Differential Mean Curvature Maps
HOG	Histogram of Oriented Gradients
SIFT	Scale Invariant Feature Transform
EBS	Elastic Body Spline
DM	Deformable Facial Mesh
FELM	Facial Expression Label Map
AFM	Annotated Face Model

TPS	Thin Plate Splines
PDM	Point Distribution Model
D3	Data-driven Documents
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
SVG	Scalable Vector Graphics
DOM	Document Object Model

Acknowledgements

I would like to express my gratitude to my supervisors: Nikolaos Vidakis, Cedric Demonceaux and Georgios Triantafyllidis for their guidance and support. I am very thankful to all members of *interactive Software Technologies & System Engineering Laboratory* in Technological Educational Institute of Crete for their general help and constructive feedback at many stages of this work.

Special thanks must go to professor Georgios Triantafyllidis for his guidance, support and collaboration on many levels during the last couple of years.

Chapter 1 - Introduction

Automatic and effective estimation of head pose is a challenging problem of computer vision systems, since it is considered to be a key element of human behavior analysis. Many applications would benefit from automatic and robust head pose estimation systems such as face recognition, human activity analysis, human-computer interaction, robotic vision etc. For this reason, and because of its numerous applications, head pose estimation has drawn great attention from academia and a variety of techniques have been reported in the literature [1]. The field of facial expression analysis is still an enthusiastic issue in latest research works due to its various purposes and applications, such as the design of better human/machine interfaces, video gaming, computer generated animations and identification. It plays a key role in emotion recognition and thus contributes to the development of human-computer interaction systems [2].

With the recent technological advancements of depth sensors, it is possible to perform data collection in terms of head movement and changes in facial expressions for subsequent analysis. Providing an objective assessment and evaluation of those findings can lead to valuable conclusions regarding the overall experience of user in many applications (e.g. in the case of educationally-oriented tasks in terms of evaluating user training and ludology experience in serious games).In this context, efficient visualization of such data can play a major part in that kind of assessment by creating encodings of data into visual channels that people can view and understand comfortably. The process of visualization is suitable for externalizing the data and enable people to think and manipulate the data at a higher level. Undoubtedly, it is considered to be an essential tool for understanding such information. In addition, visualization can be used in several distinct ways to help tame the scale and complexity of the data so that it can be interpreted more easily.

The proposed scheme presented in this thesis consists of three distinctive parts as shown in Figure 1.1. First the 3D head pose estimation and facial expression events are separately obtained for different users sitting and moving their head without restriction in front of a Microsoft Kinect sensor for specified intervals. Then the data for every user are stored in a JSON file for offline usage in creating the 2D and 3D visualizations with scatter plot and 3D columns respectively for the head pose estimation events, while a 3D visualization is also manufactured for facial expressions. The case for employing scatterplots for multidimensional visualization lies in their relative simplicity in comparison to other multidimensional visualization techniques, familiarity among users, and their high visual clarity.

The principal objective of the proposed scheme is considered to be the acquisition of efficient and user-friendly visualizations in order to improve the understanding and the analysis of the captured data, which can be easily accessed through the web.

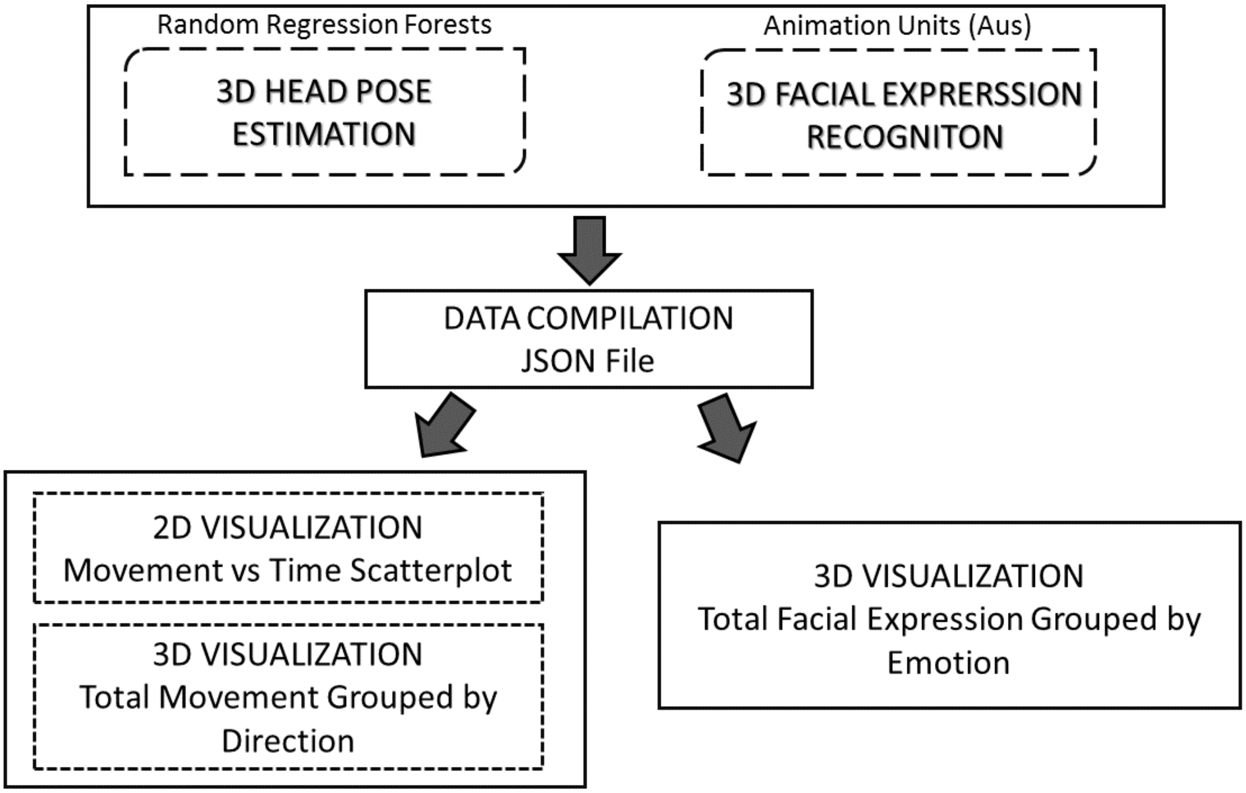

Random Regression Forests
3D HEAD POSE ESTIMATION
Animation Units (Aus)
3D FACIAL EXPRERSSION RECOGNITON
DATA COMPILATION
JSON File
2D VISUALIZATION
Movement vs Time Scatterplot
3D VISUALIZATION
Total Movement Grouped by Direction
3D VISUALIZATION
Total Facial Expression Grouped by Emotion

Figure 1.1 Proposed System Overview

Document Structure

- **Chapter 2-Head Pose Estimation:** A comprehensive review concerning the state-of-the-art methods for head pose estimation is presented. Furthermore, the method appointed for this thesis is extensively exposed. Finally, a 2D visualization for summarizing head pose estimation from the selected method is unveiled.
- **Chapter 3-Facial Expression Recognition:** A thorough review touching the state-of-the-art methods for facial expression recognition is presented. Furthermore, the method appointed for this thesis is broadly exposed.
- **Chapter 4-Data Compilation and Experiments:** In this chapter, the data compilation phase of our proposed scheme is presented alongside the experiments carried out for the necessary accuracy assessment.
- **Chapter 5-Web-based Data Visualization:** This chapter contains a brief descriptions of the libraries that were used for presenting the acquired data from the two frameworks on the web, before presenting the actual web-based visualizations which were made for user actions and expressions assessment.
- **Chapter 5-Conclusions and Future Work:** Consists of a summary and concluding remarks. Furthermore, future steps regarding this work will be discussed.

Chapter 2 - Head Pose Estimation

The ingenuity to estimate the head pose of another person is a common human skill that shows up from an early age and empowers people with the ability to quickly and conveniently interpret the orientation and movement of a human head. In a computer vision reference frame, head pose estimation is most commonly seen as the process of deducting the orientation of a person's head relative to the view of a camera, or more scrupulously relative to a global coordinate system. The human head is limited to three degrees of freedom (DOF) in pose, which can be indicated by *pitch*, *roll* and *yaw* angles as seen in Figure 2.1, while a sequence of processing phases is required in order to transform a pixel-based head representation into a high-level notion of direction. However, automatic and effective estimation of head pose is a demanding problem that has challenged computer vision systems for decades. Many applications would benefit from automatic and robust head pose estimation systems such as face recognition, human activity analysis, human-computer interaction, robotic vision etc. Therefore, due to its numerous applications, head pose estimation has drawn great attention from academia and a variety of techniques have been reported in the literature.

In this chapter, an overview of the most prominent state-of-the-art methods for continuous head pose estimation will be presented, mainly focusing on works over the last five years. After that, a specific method named Discriminative Random Regression Forests (DRRF), which was utilized for head pose estimation in this master thesis will be discussed in details. Finally a basic 2D scatterplot visualization will be presented in terms of users' head movement against the time passed.

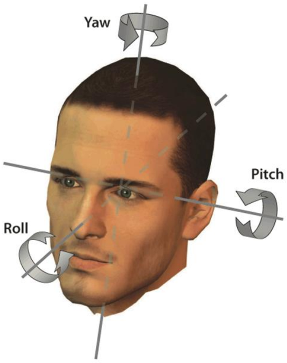


Figure 2.1 The three degrees-of-freedom of a human head

Related Work

Head pose estimation methods proposed before 2008 has been reviewed in [1] by Murphy-Chutorian. In this section up-to-date progress on head pose estimation is investigated, which are approximately classified into nine categories describing the conceptual approaches that have been used to estimate head pose, while each system was arranged by the fundamental approach that underlies its implementation as seen in Table 2-1.

Table 2-1 Taxonomy of Head Pose Estimation Methods

Approach	Representative Works	Representative Paper(s)
Appearance Template Methods ▪ Image Comparison ▪ Filtered Image Comparison	Mean Squared Error Normalized Cross-Correlation Gabor-Wavelets	[3] [4] [5]
Detector Arrays ▪ Machine Learning ▪ Neural Networks	SVM Adaboost Cascades Router Networks	[6] [7]
Nonlinear Regression Methods ▪ Regression Tools ▪ Neural Networks	SVR Feature-SVR MLP	[8] [9]
Manifold Embedding Methods ▪ Linear Subspaces ▪ Kernelized Subspaces ▪ Nonlinear Subspaces	PCA Pose-eigenspaces LEA	[10] [11] [12]
Flexible Models ▪ Feature Descriptors ▪ Active Appearance Models	Elastic Graph Matching ASM AAM	[13] [14]
Geometric Methods ▪ Facial Features	Planar & 3D Methods Projective Geometry Vanishing Point	[15] [16]

Tracking Methods - Feature Tracking - Model Tracking - Affine Transformation - Appearance-based Particle Filter	RANSAC Dynamic Templates Weighted Least Squares Adaptive Diffusion Dual-linear State Model	[17] [18]
3D Methods - Least Square Minimization - Plane fitting to 3D points - Head depth maps matching - Decision Forest	DRRF Depth rate constraint equation Ellipse around the head Particle Swarm Optimization	[15] [19] [20] [21]

Appearance Template Methods

Appearance template methods use image-based comparison measurements to match a view of a person’s head to a set of templates with corresponding pose labels. One recent example can be found in [3], where the problem of head pose classification from real world images (unconstrained environments) is being addressed in a template-based form. In their proposed methodology, each head pose is represented with a probabilistic and spatial template learned from facial codewords, which contain the probability density function for head pose class and anatomical labeling. These specific templates are motivated by the anatomical face regions (e.g. nose, mouth, ear and eyes). Therefore, pose information are retrieved not only from codewords but also from the inferred anatomical regions. Arbitrary partial occlusions are allowed by the Bayesian formulation which is being followed.

Detector Arrays

Detector arrays perform similarly to appearance templates in terms of operating directly on an image patch. Nevertheless, instead of comparing an image to an extensive set of single templates, the image is rather assessed by a detector, which was trained on plenty of images with a supervised learning algorithm. Li et al. in [6] presented a person-independent head pose estimation framework for gray-level images. In their work, two tasks are introduced and performed separately, multi-view face detection and pose angle estimation. Regarding the first task, face detection is achieved by training a tree structure that is composed by cascaded-Adaboost classifiers. For this purpose, Haar-like features are essentially extracted and then the Adaboost classifiers are trained and applied to localize the face. After that, the random forest algorithm for regression is followed to retrieve pose angles in yaw and pitch axes. At the root node of the tree, the so called, label feature is being tested while at other nodes a binary test of two different pixel's comparison is performed. Finally, the estimated angles are stored at leaf nodes. This method is applicable on complex environment with low resolution images.

Nonlinear Regression Methods

In nonlinear regression methods, the pose is estimated by learning a non-linear functional mapping from the image space to one or more directions. In cases where the dimensionality of the data can be reduced, Support Vector Regressors (SVRs) have enjoyed great success. A typical example of that can be found in [8], where Matthias Rätsch et al. adopt SVR for estimating the orientation of a human head. This resulting information is utilized in order to determine which trained classifier will be used. Therefore, two-adjustable in their complexity-stages are introduced, regression and classification, for an efficient course-to-fine approach in consideration of real-time performance on video streams something which is missing from other approaches. As a first stage, a full SVR is trained to estimate the yaw angle of the head pose with an average error of 1.30◦ on a range of ±90◦. After that, in order to confront the drawback of a very complex and slow SVR, the reduction is achieved by using Radial Basis Function, RBFkernels with a grayscale feature space and histogram equalization as normalization. High accuracy of the pose estimation is reached because the full SVR estimation is used at the last coarse-to-fine stage of the Evolutionary Regression Tree.

Manifold Embedding Methods

Each high-dimensional image sample lies on a low-dimensional continuous manifold forced by the acceptable pose variations. Head pose estimation in this method requires the manifold to be modeled and an embedding technique for projecting a new sample into the manifold. Then this low-dimensional embedding can be used for head pose estimation alongside other techniques such as regression in the embedded space. In [10] by integrating supervised Laplacian regularization and sparse regression into manifold learning, the local geometric structure is kept intact, while extracting features can be achieved more easily and in better form. In more details, manifold learning is forced to hold local geometry structure preservation in order to get more discriminative projection embedding by exploiting the Laplacian regularization term in the objective function. Dominant features are better described by casting the problem of learning projective function into a regression with L_1 norm regularizer. This method experimentally is beneficial for head pose angle estimation in 3D space.

Flexible Models

Flexible models follow a different approach compared to the four aforementioned methods. A non-rigid model is fit to the image in order to fall in with the facial structure of each individual. Training data with annotated facial features are necessary besides pose labels, but comparisons are made at the feature level rather than the global appearance model. In their human-robot interaction (HRI) application [13], Bidgoli et al. utilize a three dimensional real-time monocular head pose tracker, in which active appearance models (AAMs) are employed in order to extract facial features. Furthermore, for the sake of improving the texture model, two probabilistic approaches are preferred for principal component analysis (PCA) .Opposed to the typical assumption in AAM, the gradient matrix is suggested to be adapted with new images during model fitting of video sequences instead of being constant. They manage to achieve an enhancement between human and robot for controlling the robot's camera orientation.

Geometric Methods

These methods examine the human viewpoint of head pose to build upon indications, such as the deviation of the head from bilateral symmetry and the deviation of nose angle. In order to estimate pose, geometric approaches exploit the head shape and the strict configuration of local features. In their model Tang et al. [15] break down a range image into a set of simple slices that contain sufficient geometric cues, which can be used for precisely describing the poses of a subject. A generic scheme for designing new features for head pose estimation is introduced by this specific model. According to their model, crafting a new feature model for describing a slice is regarded as the first step. After that, a new set of features is created by combining all slices with a view to describe range images. Two innovative range image representations are utilized, Local Slice Depth (LSD) and Local Slice Orientation (LSO). The former is used for coarse estimation of head poses, while the latter can accomplish accurate results. In their implementation, color and range images captured by a Kinect sensor are used in order to localize and segment the facial region from the background. Then the real-time feature extraction as described takes place and finally random forests are used for learning a steady tie between slice feature descriptors and head pose parameters.

Tracking Methods

Tracking methods operate by following the respective motion of the head between consecutive video frames. A descent estimation of pose over time is achieved by utilizing smooth motion and temporal continuity constraints. A high level of accuracy may be achieved, but initialization from a known head position is a precondition. A method for 3D head pose recovery from video by using a 3D cross model to track head motion changes is presented in [17]. An initial front view of the head is required in order for a biologically-reasonable 3D cross model to be constructed, which will represent the head during the video frames. Eyebrow center is regarded as the center of the 3D cross model, while the tip of the nose and the center of the forehead are taking as the four end points of the model. Then the shape of the head is divided into two parts, horizontal and vertical. After that, the head tracking process takes place, in which the created 3D cross model is projected to the initial template in order to approximate the head. Finally, when the face is detected, both the initial reference template and the corresponding head pose are calculated. Based on the optical flow method, full head motion can be traced from input images. The effects of self-occlusion, head large motion and gradual illumination changes can be diminished by updating the model templates dynamically. Furthermore, when the head is hidden or moves out of scene, the model has the ability to be reinitialized automatically.

DRRF for Real Time Head Pose Estimation from Depth Data

Systems relying on 3D data have demonstrated very good results in terms of head pose estimation, compared to 2D systems that have to overcome vaguenesses in real time applications. This section briefly describes the method proposed by Fanelli et al [19], which is utilized in our work as it is regarded to be suitable for real time 3D head pose estimation, considering its robustness to the poor signal-to-noise ratio of current consumer depth cameras like Microsoft Kinect sensor. In this direction, regression forests are being extended in such a manner that depth patches belonging to a head can be discriminated and solely used for the prediction of the pose resulting in solving both the classification and regression problems respectively. While several works in the literature foresee the case which the head is the only object present in the field of view [20], the proposed method concerns depth images where other parts of the body might be visible as well and therefore need to be disjointed into image patches, which belong to the head and which do not. The system is able to perform on a frame-by-frame basis, while it runs in real time without the need of initialization.

Forests of randomly trained trees are less sensitive to over-fitting and generalize better than decision trees independently. In the proposed setup [22] depth patches are annotated with class label and a vector $\theta_\iota =\{\theta_\chi, \theta_\upsilon, \theta_\zeta, \theta_{y\alpha}, \theta_{pi}, \theta_{ro}\}$ containing the offset between the 3D points falling on the patch’s center and the head center location, plus the Euler rotation angles describing the head orientation. Randomness is imported in the training process, either in the set of training examples provided to each tree or in the set of tests usable for optimization at each node, or even in both. When the pair of classification measure $U_C(\{P|t^k\})$and regression measure $U_R(\{P|t^k\})$ are engaged, the aggregation of trees which simultaneously separate test data into positive cases (they represent part of the object of interest) are labeled as Discriminative Random Regression Forests (DRRF). This signifies that an extracted patch from a depth image is sent through all trees in the forest. The patch is evaluated at each node according to the stored binary test as defined in (1) and passed either to the right or left child until a leaf node is reached [23] as shown in Figure 2.2.

$$|F_1|^{-1} \sum_{q \in F_1} I(q) - |F_1|^{-1} \sum_{q \in F_2} I(q) > \tau, \tag{1}$$

Leaves store two kinds of information: The ratio of positive patches that reached them during training p(c=1|P) and the multivariate Gaussian distribution computed from the pose parameters of the positive patches. Figure 2.3 shows some processed frames regarding the two DOF (pitch and yaw). Starting from left to right, the first row estimations displayed are: *still*, *up*, *down*. The second row estimations are *left*, *right* correspondingly. All calculations derived from the difference between the exact previous frame and the current frame, at each iteration of the program. The green cylinder encodes both the estimated head center and direction of the face.

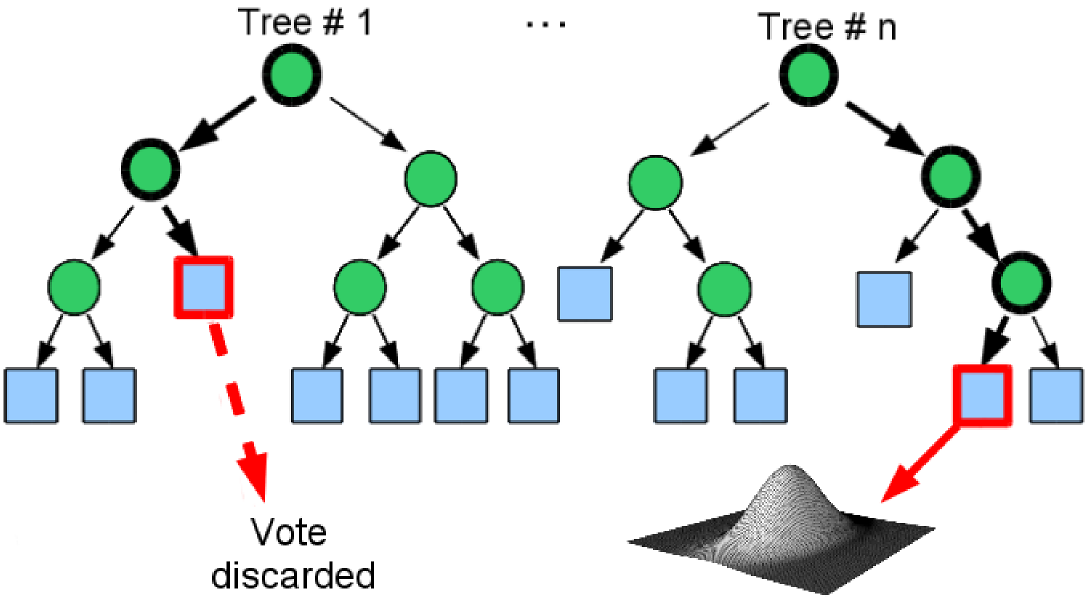


Figure 2.2 Example of DRRF : A patch is sent down to two trees, ending up in a non-head leaf in the first case, thus not producing a vote, and in a head leaf in the second case, extracting the multivariate Gaussian distribution stored at the leaf.

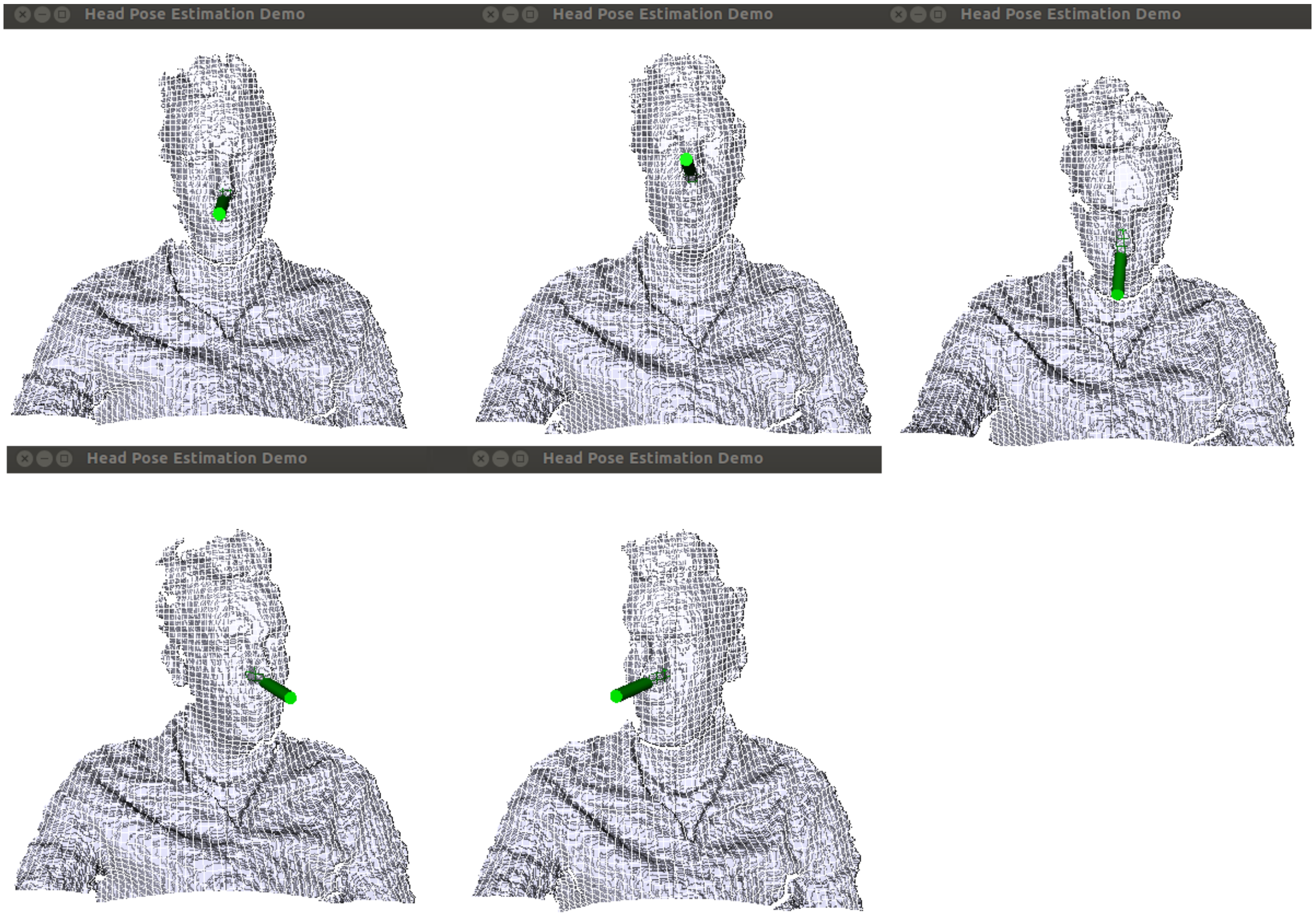

Head Pose Estimation Demo
Head Pose Estimation Demo
Head Pose Estimation Demo
Head Pose Estimation Demo
Head Pose Estimation Demo

Figure 2.3 Some processed frames regarding two DOF (pitch and yaw), as shown by the main application window for head pose estimation

Movement vs Time Web Visualization

In the final section of this chapter, a 2D scatterplot visualization was constructed, for presenting the users' movement vs time as shown in Figure 2.4. In more details, the x axis represents the time scale in seconds during which the tests take place (Figure 2.4 shows only a zoomed portion of the whole scatterplot graph), while each label in y axis symbolizes each different user who participates in the test. Four different arrows imitate the movement of the human's head in two DOF. Furthermore an additional feature is displayed when the mouse is hovering an arrow, showing the respective time the movement had occurred and the intensity, which is based on how large was the difference between the previous and the current frame as explained in Data Compilation section. Apart from those elements, a color fluctuation is also evident, serving as an intensity indicator for each movement (the closer to red color the arrow is, the higher the intensity of the movement). One can easily examine the players' movement that way, alongside their intensity which adds a different dimension to the knowledge gained from the visualization. The full version of the web-based visualization can be found at: http://83.212.117.19/Scatterplot_HeadPoseEstimation_MSCV/

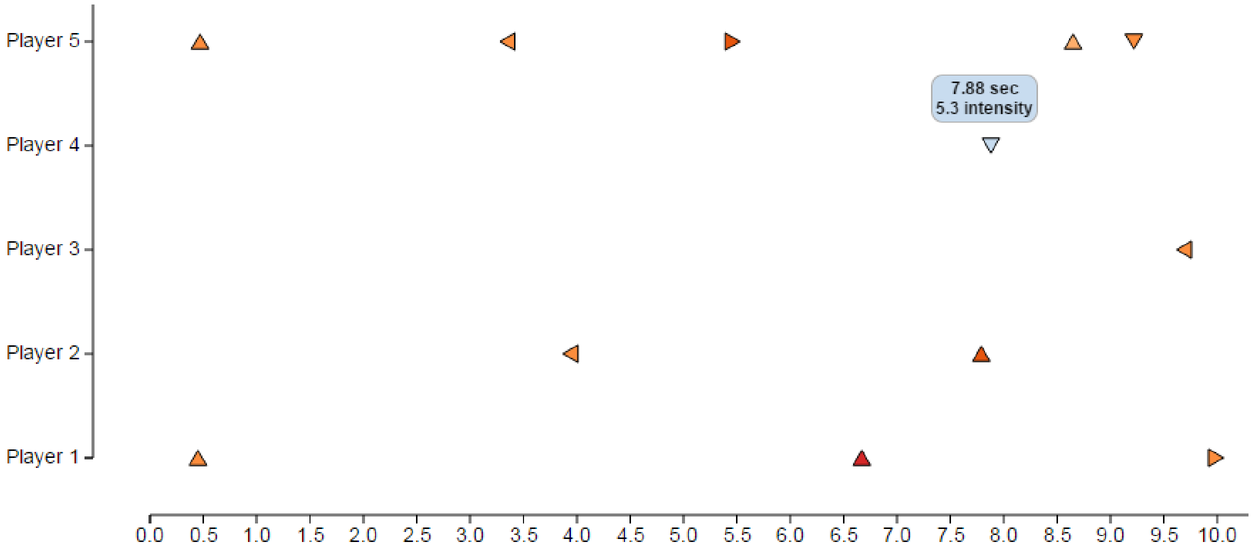


Figure 2.4 2D Scatterplot of Head Pose Estimation

Chapter 3 - Facial Expression Recognition

Facial expression is one of the most dominant, natural and instantaneous means for human beings to communicate their emotions and intentions. The reason for this lies in the ability of the human face to express emotion sooner than people verbalize or even realize their feelings. Facial expressions are generated by facial muscle contractions which lead to temporal facial deformations in facial geometry and texture. Humans may be able to observe and clarify faces and facial expressions in a scene with little or no effort, however development of an automated system that performs this task is still regarded as a rather difficult process. There are considerable relevant problems such as: detection of an image segment as a face, derivation of facial expression information and classification of the expression in emotion categories. Therefore, facial expression recognition (FER) and analysis (FERA) alongside discrete emotion detection, have been active topics in computer science for some time now, while many encouraging approaches have been reported. Due to its key role in emotion recognition, a contribution can easily be made to the development of human-computer interaction (HCI) systems by FER. Furthermore, face recognition systems would greatly benefit by prior knowledge on the facial motions and facial feature deformations provided by such systems. However, due to the subtlety, complexity and diversity of facial expressions and inter person facial differences, automatic FER remains a challenging task.

In this chapter, a brief overview of the most prominent existing methods for FER and FERA will be presented, focusing on works which exploit 3D information. After that, a specific method based on Action Units (AUs) coding, which was applied for emotion recognition via facial expression (ERFE) in this master thesis will be explained in details.

Related Work

Existing approaches in 3D FER can be typically divided into two categories as stated by Fang et al. in their survey [2]: feature-based and model-based. Feature based 3D FER methods concentrate on the extraction of facial features directly from the input scan. On the other hand, model-based approaches normally engage a generic face model as an intermediate for bringing input scans into correspondence by means of registration and/or dislocation. In this section state-of-the-art works related to 3D FER are investigated and summarized in Table 3-1. Must be noted that each system was arranged by the fundamental approach that underlies its implementation. Furthermore, in Table 3-2 key properties of those approaches are outlined.

Table 3-1 Taxonomy of 3D Facial Expression Recognition Methods

Approach	Representative Works	Representative Papers
Feature-based	▪ Local Depth Features ▪ SURF Features ▪ DMCMs & HOG Features ▪ SIFT Features	[24] [25] [26] [27]
Model-based	▪ Deformable 3D Model ▪ Vertex-level correspondence ▪ Bilinear Models ▪ Annotated Face Model	[28] [29] [30] [31]

Feature-based Facial Expression Recognition

In [24], a fully automatic facial expression recognition algorithm based on depth features extracted from local patches is presented. Only the 3D shape is investigated because most of the times facial expressions are encoded in facial geometry deformations rather than textures. As a first step, local patches must be defined without human intervention, therefore the nose tip and four eyes corners are detected automatically using a Haar detector and AdaBoost classifier and regarded as the fiducial landmark points. From those five points their relative distances are utilized in order to produce another 25 heuristic landmarks for local depth features to be extracted form patches around all the 30 landmarks, in order to represent facial expressions. This happens because those fiducial points are not representative enough on their own for extracting expression features. Then, as a result the depth features are projected to a lower dimensional subspace where feature selection is carried out by maximizing their relevance and minimizing their redundancy, for the reason that depth features contribute differently to each type of expression. Finally the selected features are fed to a Support Vector Machine (SVM) classifier for expression classification to take place. An average recognition rate of 83% was achieved by this method for the six prototypic facial expressions.

Azazi et al. in [25] are using Speeded Up Robust Features (SURF) in order to evaluate several state-of-the-art classification schemes such as SVM, Neural Networks (NN), k-Nearest Neighbors (k-NN) and Naïve Bayes (NB). In their framework, a pre-processing stage is introduced in order to map all the 3D textured face images of the BU-3DFE database into the 2D plane by utilizing conformal mapping. After that, seven main landmarks (eye corners, nose tip and mouth corners) are detected by structured output SVM. Based on those main landmarks, a set of 52 facial points are located and the features of those points are considered to be the optimal facial features in the textured mapped 2D images. Then, a selection algorithm is applied in order to identify the optimal feature set that could differentiate rationally between the universal facial expressions, resulting to a set of 52 features. For every image in the database, the 52 SURF descriptors are linked together in order to form a single descriptor for each image. Finally, those descriptors are fed to different classifiers both for training and testing purposes. RBF-SVM incomparably outperformed the other tested classifiers, followed by NB.

Another complete, fully automatic approach to 3D FER is presented by Lemaire et al. in [26]. An innovative facial representation namely Differential Mean Curvature Maps (DMCMs) is suggested for acquiring both global and local facial deformations, which regularly take place during facial expressions. The main idea of the proposed method is to represent 3D face models over a set of 2D maps in contemplation of taking advantage of the 2D-based image processing tools that exist. An integral computation is used for calculating the mean curvatures in order to extract the aforementioned DMCMs straight from depth images. Those maps represent curvature-like data and highlight the 3D surface topology at various scales. After the map extraction, each DMCM is normalized in an attempt of retaining only the informative parts of the face while discarding as many as possible model boundaries. The next step consists of describing DMCMs by utilizing Histogram of Oriented Gradients (HOG) algorithm. For this purpose, a regular grid is applied to several subdivisions of the decomposed normalized DMCM. Thereafter, each subdivision is described by HOG and finally the descriptors of every subdivision are concatenated in order to form a global descriptor. Lastly, the multi-class SVM algorithm is used both for learning and testing purposes. The 6 prototypical expressions were tested by this method, while an average recognition rate of 78.13 was achieved.

In [27] a set of selected Scale Invariant Feature Transform (SIFT) features for 3D FER is investigated by Berretti et al. Other recent works have shown that salient keypoints and local descriptors can effectively be used for describing 3D objects. Therefore, they are exploiting the local characteristics of the face by computing SIFT descriptors situated over a small set of facial landmarks identified on range images. SIFT algorithm originally was defined for 2D gray-scale images, so in order to bypass this limitation, range images that make use of the gray-scale of every pixel to represent the depth data of a scan are employed. Some steps are required to be performed in order for the facial landmarks to be obtained, as well as for the transformation of 3D face scans into range images. Hence, initially a subset including 20 landmarks on the mouth, two on the nose and five on the face contour were considered, while 85 additional landmarks were identified and used as keypoints. An average expression recognition rate around 77.54 % was achieved by this approach.

Model-based Facial Expression Recognition

An approach based on fiducial point controlled 3D facial model is presented by Tie et al in [28]. A physics-based transformation namely elastic body spline (EBS) is put into use over a deformable facial mesh (DM) for setting up a smooth wrap that reflects the control point matching to the dislocation of fiducial points. As a first step, facial regions in the input video sequence are detected by applying a local normalization technique. Thereafter 26 fiducial points are selected based on the anthropometric measurement achieving maximum movement of the facial components at the time expressions take place. EBS is applied for generating different facial expressions with a generic facial model from a neutral face. Just after the acquisition of the deformable facial features, a D-Isomap based method for emotion classification is exploited. For their experiments two different databases were used, RML emotion database and Mind Reading DVD database. Overall system performance that was achieved was around 90.93%.

In their paper [29] Rosato et al. introduce a method for automatically establishing vertex correspondences for feature registration and facial models classification to specific expressions. Two primary elements constitute this method, a) vertex correspondence establishment based on a conformal mapping and a generic model matching b) suggestive model labeling placed on primitive surface feature classification and sequential model tracking and classification. In their context, a vertex-level correspondence between two meshes is considered to be a mapping from each vertex of one mesh to the vertices of another and plays a key role in investigating the facial surface similarity and its dynamics. At a high-level of description, that mapping is actually a one-to-one correspondence establishment between the 3D mesh and another mesh with a shape restriction that was set as a guide. They employed a circle pattern conformal mapping in order to establish a 2D representation of a 3D mesh, this happens for simplifying reasons regarding the correspondence problem. A deformable template-based approach is applied to extract 22 feature points. Facial surface labeling takes place for describing the facial surface context (curvature, crest lines etc.) and its behavior. By labelling the surfaces of both 3D scans and the tracking model, they are able to determine each expression’s facial expression label map (FELM). The BU-3DFE database was also used for their experiments, while a recognition rate at 80.1% was achieved with this method.

Another attempt on discrete expression classification using 3D data from the human face can be found in [31]. Fang et al. divide their work in two parts, the first one where they present an improvement to the fitting of the Annotated Face Model (AFM) in a way that a dense point correspondence can be found in terms of position and semantics among static 3D face scans. The second part consists of an expression recognition framework on static 3D images. They improved the AFM on static 3D face scans by introducing Procrustes Analysis and thin plate splines (TPS) to the fitting process. Secondly, a key point selection (Spin image or MeshHOG) and filtering (RANSAC) mechanism are employed in order to accelerate the expressive 3D facial scans' registration in a sequence. Next an algorithm that exploits component-based point distribution model (PDM) is introduced for taking advantage of the dense point correspondence concerning the understanding of shape deformations from expressions. SVMs with a radial basic function kernel were selected as the classifier in their proposed method, while they achieved around 91.0% recognition rate.

Bilinear models for simultaneously addressing 3D face and FER is presented in [30]. An elastically deformable model algorithm which sets up correspondence among a set of faces is proposed while bilinear models are created, which decouple the identity and facial expression factors. An evaluation of the proposed technique took place on the BU-3DFE face database reaching an overall 90.5% FER rate. The six prototypic expressions are put into test by utilizing a bootstrap set of faces in order to tune an asymmetric bilinear model, which is integrated in a probabilistic framework. A simple linear system of equations is used for carrying out the establishment of point correspondences among faces, which are organized automatically by building a low-dimensional face eigen-space.

Table 3-2 Methods for 3D Facial Expression Analysis

Reference	Category	2D Used? [a]	Landmarks	Database	Expression Types	Performance (%) [b]
[24]	Feature	N	30 auto	BU-3DFE	6	83
[25]	Feature	Y	7 auto	BU-3DFE	6	79.36
[26]	Feature	Y	7 auto	BU-3DFE	6	78.13
[27]	Feature	N	20 auto	BU-3DFE	6	77.54
[28]	Model	N	26 semi-auto	RML, Mind Reading DVD	7	90.93
[29]	Model	Y	22 semi-auto	BU-3DFE	6	80.1 static 85.9 dynamic
[30]	Model	N	12 semi-auto	BU-3DFE BU-4DFE	6	91.0
[31]	Model	N	7 manual	BU-3DFE	6	90.5

a) Denotes whether the method makes use of the 2D texture associated with the 3D data; b) the average recognition rates are listed only for reference not for comparison due to different experiment settings.

Real-time Emotion Recognition via Facial Expressions

Features which are utilized for classifying human affective states, are commonly based on local spatial position or dislocation of explicit points and regions of the face, contradictory to audio-based approaches which use global statistics of the acoustic features. Recognition of facial action units (AU) is one of the two main streams in facial expression analysis. AUs are anatomically related to the reduction of specific facial muscles, 12 for upper face and 18 for lower face as stated in [32].The facial action coding system (FACS) is a system for human facial expression categorization, originally developed by Paul Ekman and Wallace Friesen in 1978 [33]. Action Units (AUs) in (FACS) describe the facial deformations in terms of visually detectable muscle actions. They are contemplated as signals to describe human faces and therefore provide a probability to derive expressions through a high-level decision making process. Since AUs come from the reduction of facial muscles and lead to deformation on facial landmark location, texture and facial surface, 3D faces serve benefit in recognizing AUs. A total of 44 AU can be derived from the face, while they and their combination can compose different facial expressions. Figure 3.1 (Nilesh Powar 2011 [34]) illustrates the 4 basic universal expressions which are investigated in this work, happiness, surprise, sadness, anger described by utilizing AUs combinations as shown by the corresponding labels. For example, considering one of the basic expressions "Happiness", is consisted of two separate units AU 6 (Check Raiser and Lid Compressor) and AU 12 (Lip Corner Puller).

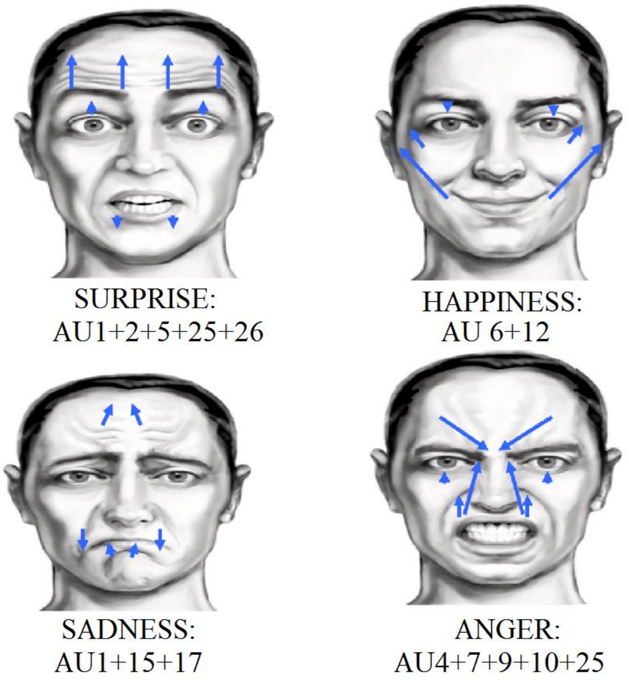


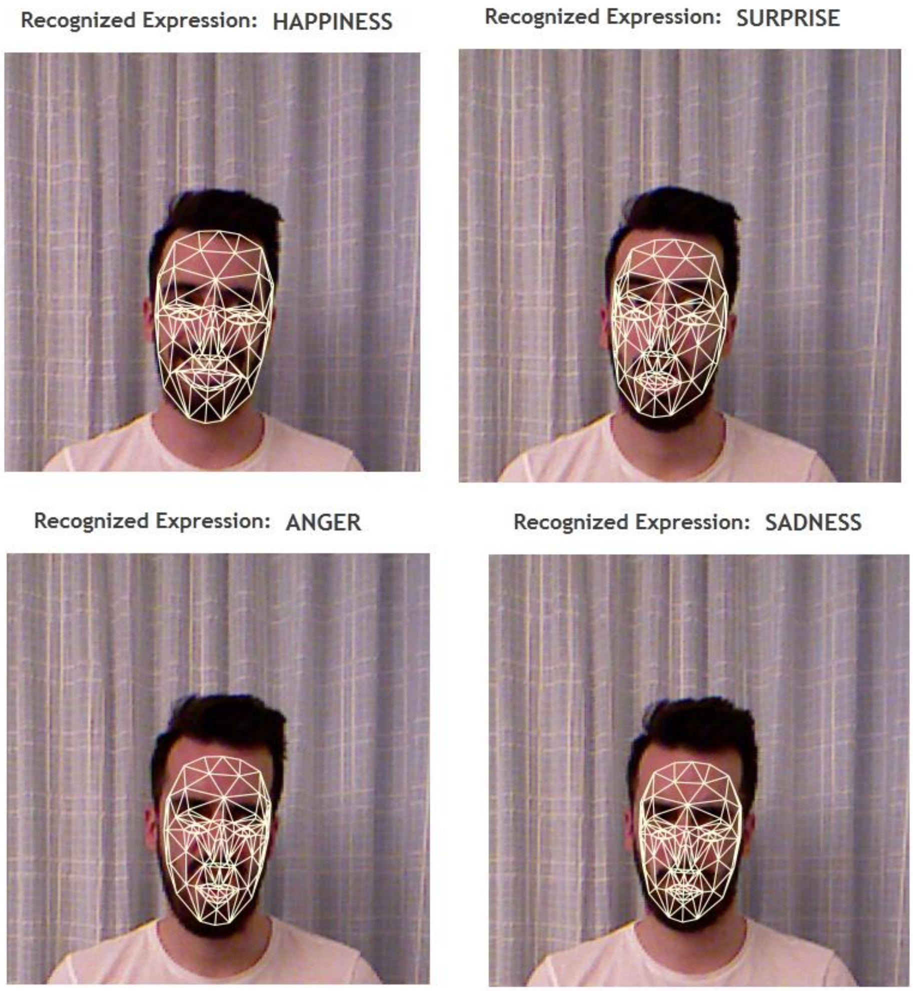
Figure 3.1 Four Basic Universal Expressions: Surprise, Happiness, Sadness, Anger

A similar to [35] approach was followed for real-time emotion recognition. Video sequences acquired from the Kinect sensor are regarded as input. Then face detection and feature extraction are performed on each frame of the stream. The Face Tracking SDK [36] , which is included in Kinect's Windows Developer toolkit, is used for tracking human faces with RGB and depth data captured from the sensor. Furthermore, facial animation units and 3D positions of semantic facial feature points can be computed by the face tracking engine, which can lead to emotion recognition via facial expressions.

Face tracking results are expressed in terms of weights of six animation units, which belong to a subset of what is defined in the Candide3 model [37]. Each AU, which are deltas from the neutral shape, is expressed as a numeric weight varying between −1 and +1, and the neutral states of AUs are normally assigned to 0. The AU's feature of each frame can be written in the form of a 6-element vector:

$$\overline{\alpha} = (A_1, A_2, A_3, A_4, A_5, A_6), \qquad (2)$$

where A1, A2, A3, A4, A5, and A6 refer to the weights of 'lip raiser', 'jaw lower', 'lip stretcher', 'brow lower', 'lip corner depressor', and 'brow raiser', respectively. For the purpose of this work four different emotions were tested: anger, happiness, sadness and surprise as shown in Figure 3.2. For example, (0.3, 0.1, 0.5, 0,−0.8, 0) corresponds to a happy face, which means showing teeth slightly, lip corner raised and stretched partly, and the brows are in the neutral position. Figure 3.3 demonstrates the FER application window. On the left image, only the Kinect stream is visible while the mesh mask is not loaded. When the face is located successfully, right image, the mesh is drawn and the recognized expression is displayed as text above the window.



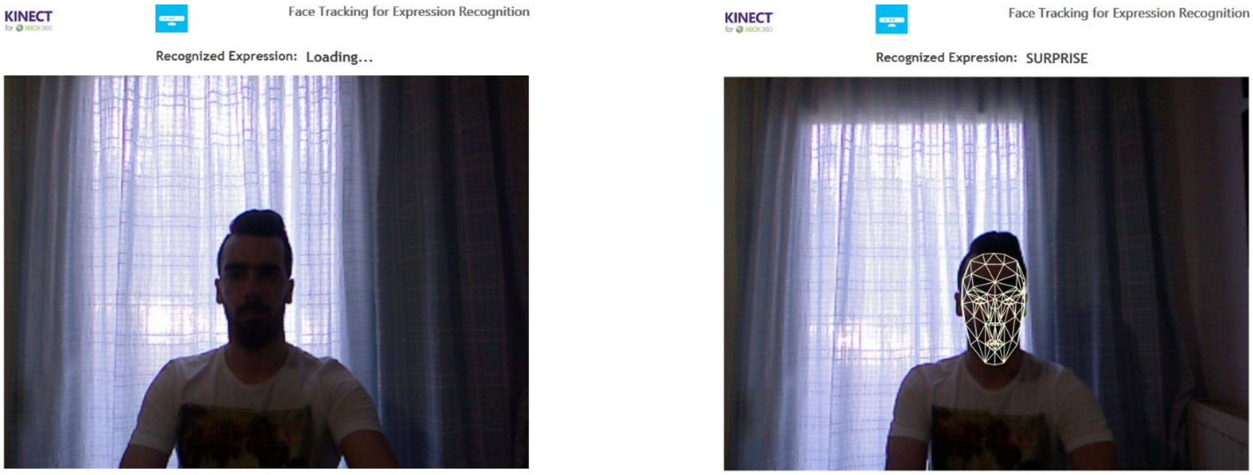
Recognized Expression: HAPPINESS

Recognized Expression: SURPRISE

Recognized Expression: ANGER

Recognized Expression: SADNESS

Figure 3.2 FER Results



Face Tracking for Expression Recognition

Recognized Expression: Loading...

Face Tracking for Expression Recognition

Recognized Expression: SURPRISE

Figure 3.3 Main application window for FER

Chapter 4 - Data Compilation and Experiments

With the recent technological advancements of depth sensors, it is possible to perform data collection in terms of head movement and changes in facial expressions for subsequent analysis. As stated previously, the principal objective of this work is considered to be the acquisition of efficient and user-friendly visualizations in order to improve the understanding and the analysis of the captured data, which can be accessed easily through the web. In order to achieve this goal, a mechanism for collecting and storing data in a way that can easily lead to web-based visualizations had to be engaged. A lightweight data exchange format, JavaScript Object Notation (JSON) was preferred for this task. In this chapter, the data compilation process based on the aforementioned frameworks is presented together with the datasets and the experiments carried out for that purpose.

Data Compilation

In this section, the data compilation process based on the aforementioned frameworks, is presented. Data representing the changes of a person's head direction, concerning two DOF, pitch and yaw, had to be obtained. To that end, some modifications were necessary to be made in the work presented in [19],in order for the current streaming frame of Microsoft Kinect sensor to be examined in contrast with the immediate previous (returned by the forest) for tracing the direction shift. A threshold was experimentally set around 5.5 for controlling the direction change as shown in (3) - (8) of a person's head in relation to the frames difference.

$$pitch\_difference = pitch_{t-1} - pitch_t \tag{3}$$

$$yaw\_difference = yaw_{t-1} - yaw_t \tag{4}$$

$$UP \quad = pitch\_difference > threshold \tag{5}$$

$$DOWN = pitch\_difference < threshold \tag{6}$$

$$LEFT \quad = yaw\_difference > threshold \tag{7}$$

$$RIGHT = yaw\_difference < threshold \tag{8}$$

Concerning the detection of emotions, boundaries for each Animation Unit had to be created in order to associate the vector obtained by the AU feature with the four simple emotions as shown in (9) – (12) in relation to (2).

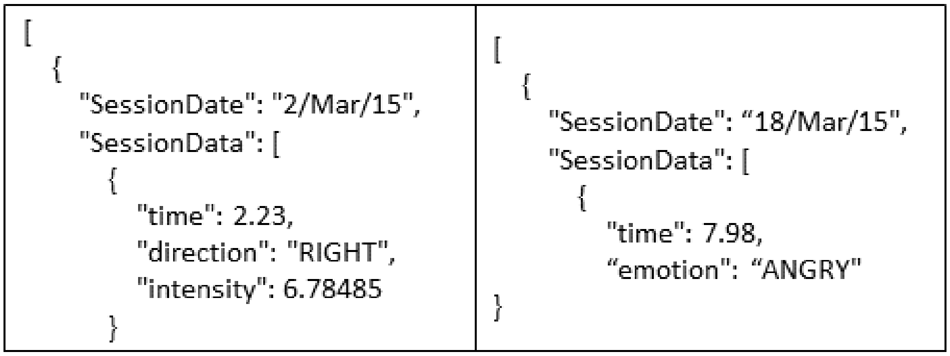
$$sadness = A_6 < 0 \ AND \ A_5 > 0 \qquad (9)$$

$$surprise = (A_2 < 0.25 \ OR \ A_2 > 0.25) AND \ A_4 < 0 \qquad (10)$$

$$happiness = A_3 > 0.4 \ OR \ A_5 < 0 \qquad (11)$$

$$anger = ((A_4 > 0 \ AND \ (A_2 > 0.25 \ OR \ A_2 < -0.25)) \text{ OR } (A_4 > 0 \ AND \ A_5 > 0)) \qquad (12)$$

Regarding the storage of the obtained data, JavaScript Object Notation (JSON) format was used mainly because of its lightweight nature, convenience in writing and reading and more importantly, as opposed to other formats such as XML, its usefulness in generating and parsing tasks in various Ajax applications as described in [38].A record in an array was created for each user session, while an extra array was inside it, preserving three variables: time, direction and intensity for each movement that was detected as shown in Figure 4.1.For facial expressions, a similar array was created but in this case only two variables were needed to be stored: time and emotion.



```
[
   {
      "SessionDate": "2/Mar/15",
      "SessionData": [
         {
            "time": 2.23,
            "direction": "RIGHT",
            "intensity": 6.78485
         }
```

```
[
   {
      "SessionDate": “18/Mar/15",
      "SessionData": [
         {
            "time": 7.98,
            “emotion": “ANGRY"
}
```

Figure 4.1 JSON structure

Experimental results and datasets

In this section the datasets which are necessary for assessing the selected frameworks alongside the actual results will be presented. Furthermore, the datasets can be found in JSON format in CHAPTER 1 - Chapter 1 -Appendix A - .In contemplation of assessing the validity of our modified version of the DRRF approach, we performed the following experiments in order to construct the ground truth data as well as collecting the experimental results concerning the two DOF, both in the JSON format as shown in Figure 4.1. First, the ground truth data had to be constructed, therefore one JSON file consisting of 10 different sessions, each one populated with specific movements and their corresponding time, was manually created. Concerning the collection of the actual experimental results, 10 subjects (each subject indicates a new session) were asked to move their head in explicit direction and time intervals. Finally, the obtained results were put against the pre-assembled ground truth data. The experiments are controlled by a number of parameters. Some parameters were fixed intuitively during the establishment stage of the experiments, for example a threshold was set in order to split actual changes of the pose from negligible ones that can occur when a user moves his head in an uncontrolled environment.

Two well-known evaluation measures were used, recall and accuracy for our case. In general the term positive is used for the identified cases while the term negative for the rejected ones. Therefore:

- True positive (TP) = correctly identified
- False positive (FP) = incorrectly identified
- True negative (TN) = correctly rejected
- False negative (FN) = incorrectly rejected

Recall or sensitivity (as known in psychology) is the proportion of real positive cases that are correctly predicted as positive (13), as stated by David Powers [39].

$$Recall = Sensitivity = tpr = \frac{TP}{TP + FN} \qquad (13)$$

Accuracy is the proportion of true results, both true positives and true negatives, among the total number of cases investigated (14).

$$Accuracy = \frac{TP + TN}{P + N} \qquad (14)$$

An overall accuracy of 83.5% and an overall recall of 96.9 % was achieved by our method regarding all the 10 tests that were carried out. Table 4-1 shows the experimental results in terms of recall and accuracy for every direction that was tested. The head pose method run by 30 fps at a computer with an Intel Core Duo CPU @ 3.00GHz.

Table 4-1 Head Pose Experimental Results

Direction	Recall %	Accuracy %
UP	87.5	70
DOWN	100	91.6
LEFT	100	90
RIGHT	100	84.2
TOTAL	**96.9**	**83.95**

Concerning the FER framework, the same approach was followed. At the beginning, the ground truth data had to be constructed, therefore one JSON file consisting of 10 different sessions, each one populated with specific facial expressions and their corresponding time, was done by hand. In the matter of collecting the actual experimental results, 10 subjects (each subject indicates a new session) were asked to make specific facial expressions, looking towards the direction of the Kinect, and time intervals. Finally, the obtained results were put against the pre-assembled ground truth data. The experiments are controlled by a number of parameters similar to head pose estimation framework. Some parameters were fixed intuitively during the establishment stage of the experiments, for example neutral facial expression difference between current and previous frame must be higher than 0.2 in order to be considered as normal change, otherwise it is regarded as neutral. This happens in order to prevent small changes in AUs from one frame to another playing a part in classifying emotions.

An overall accuracy of 76.58% and an overall recall of 95.6 % was achieved by the aforementioned FER method regarding all the 10 tests that were carried out. Table 4-2 shows the experimental results in terms of recall and accuracy for every one of the four emotions that were tested. The FER method run by 30 fps at a computer with an Intel Core Duo CPU @ 3.00GHz.

Table 4-2 FER Experimental Results

Emotion	Recall %	Accuracy %
HAPPINESS	92.3	80
SADNESS	100	81.8
ANGER	90	90
SURPRISE	100	54.5
TOTAL	**95.6**	**76.58**

It is evident from the results that our modified head pose estimation framework achieves accuracy close to and at some points even better results compared to other state-of-the-art methods. Must be noted that those experiments were conducted in uncontrolled environment in terms of lighting and ambient movement. This method achieves real-time performance comfortably even in those conditions. However, results concerning the FER framework, seem to be slightly behind in terms of accuracy. This is not a surprise, as no pre-processing steps took place. Furthermore, the simplicity level of the whole framework allows the system to perform real-time but demonstrates some drawbacks when changes between neighboring expressions take place, such as happiness and surprise (mouth slightly open in both of them).

Chapter 5 - Web-based Data Visualization

Although many different approaches have been proposed in the literature to solve both problems of head pose estimation and facial expression recognition, very few focus on how those data can be presented in order to deliver a useful meaning easily in terms of different context applications, such as serious games. To that end, we reached the decision of interpreting those data through the convenience of a website. In this section, the final and most decisive step, in terms of data comprehension in the proposed scheme, is presented alongside the actual results that can be found directly on the web. In the following subsections a JavaScript library, named D3.js, for manipulating documents based on data is briefly exposed while we look at another JavaScript library used in our work, named Highcharts.

D^3: Data-Driven Documents

Data-Driven Documents (D3) is a novel representation-transparent concept for web-based visualizations. This JavaScript library assists users at bringing data to life using varied technologies such as HyperText Markup Language (HTML) for page content, cascading style sheets (CSS) for aesthetics, JavaScript for interaction, scalable vector graphics (SVG) for vector graphics and so on. As claimed by Michael Bostock et al in [40] D3’s emphasis on web standards provides full capabilities of modern browsers while it combines powerful visualization components and a data-driven approach to a shared representation of the page called the document object model (DOM). However D3 should not be considered as a traditional visualization framework mainly because rather than introducing a novel graphical grammar, D3 solves the problem of efficient document manipulation based on data. Therefore, D3’s fundamental contribution is a visualization kernel, closer to other document transformers like jQuery [41], CSS, rather than a framework.

Highcharts

Highcharts is a charting library written in pure JavaScript which suggests an easy way of adding interactive charts to web applications. Currently many different chart types are supported by this library, such as box plot, pie charts, column charts etc. Many of these can be combined in one chart. This library was first released in late 2009, more details can be found in [42]. One big advantage of this library lies in being packed with adapters, which means that it does not rely on one particular framework, but instead is pluggable to different frameworks. The default framework implementation of Highcharts uses jQuery, hence the only requirement for users is to load the jQuery library before Highcharts.

Total Players’ Movement Web Visualization

This visualization consists of a 3D column diagram, which illustrates the aggregation of all users’ movements grouped by direction every two seconds as shown in Figure 5.1. The four different directions are imitated by four different colors. In one hand, x axis represents the time scale which is divided every two seconds until the end of the test. On the other hand, y axis displays the number of movements for all the users that take part in the tests. Furthermore, when hovering above a column, the number of the corresponding direction summary is displayed. In this fashion, the dominant direction amongst all users every time interval is effortlessly assumed. Moreover, not so evenly distributed movements, see for example columns between 14-16 seconds in Figure 6, can lead into practical conclusions taking into account the nature of the test as well. Thus, in the context of a serious game, the movement to the right from 9 out of 10 users between 14 and 16 seconds, gives the supervisors an indication about something which distracts the players and makes them change their head pose. The full version of the web-based visualization can be found at http://83.212.117.19/3D_HeadPoseEstimation_MSCV/.

Furthermore, this kind of visualization gives the opportunity for all charts separately or even all of them to be downloaded and saved locally as portable network graphics (PNG) images, as well as other well-known image formats. Figure 5.2 and Figure 5.3 illustrate the players’ movement only in terms of one DOF, pitch and yaw respectively.

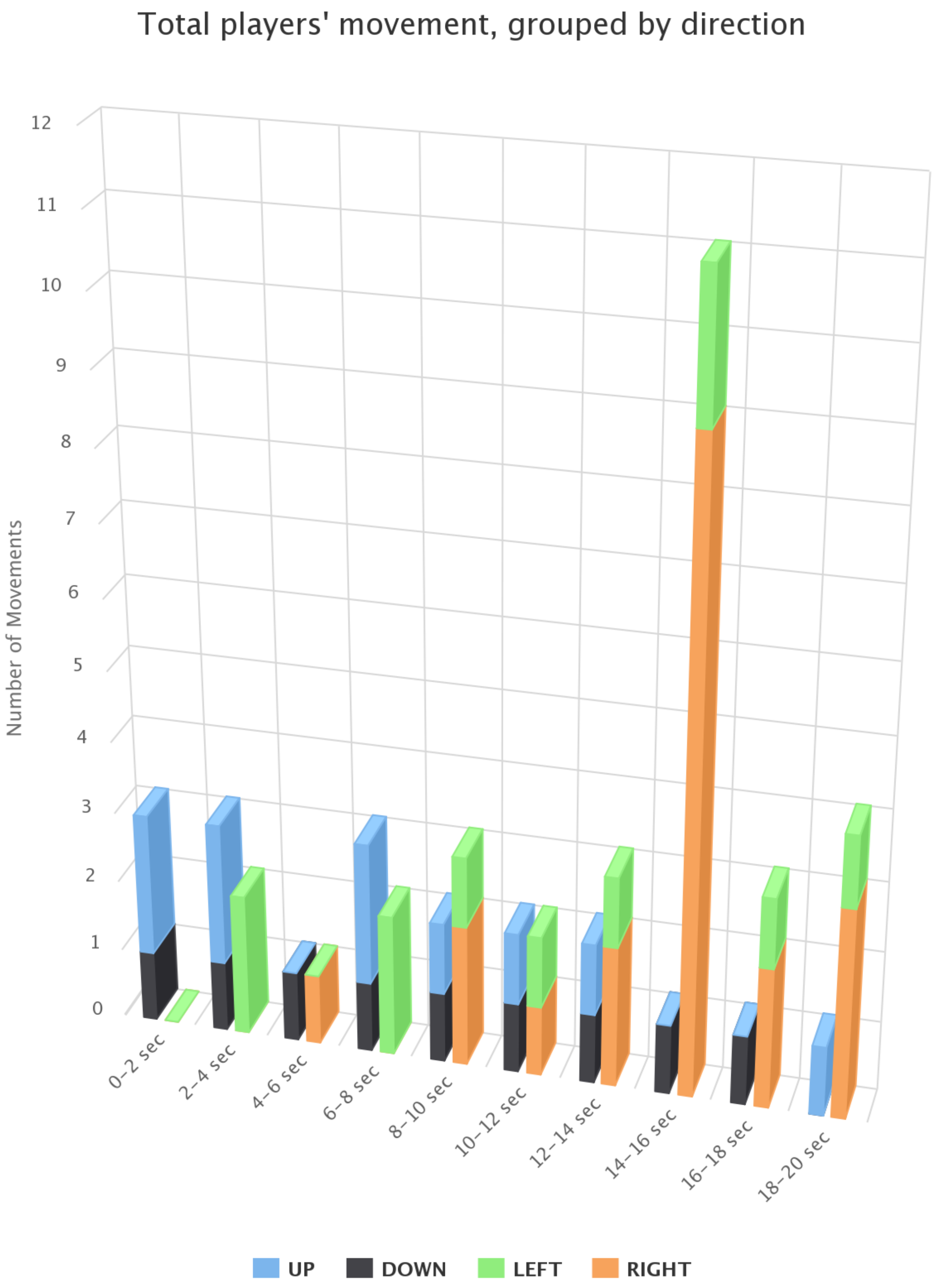


Figure 5.1 3D Column Visualization of Total Head Pose Estimation

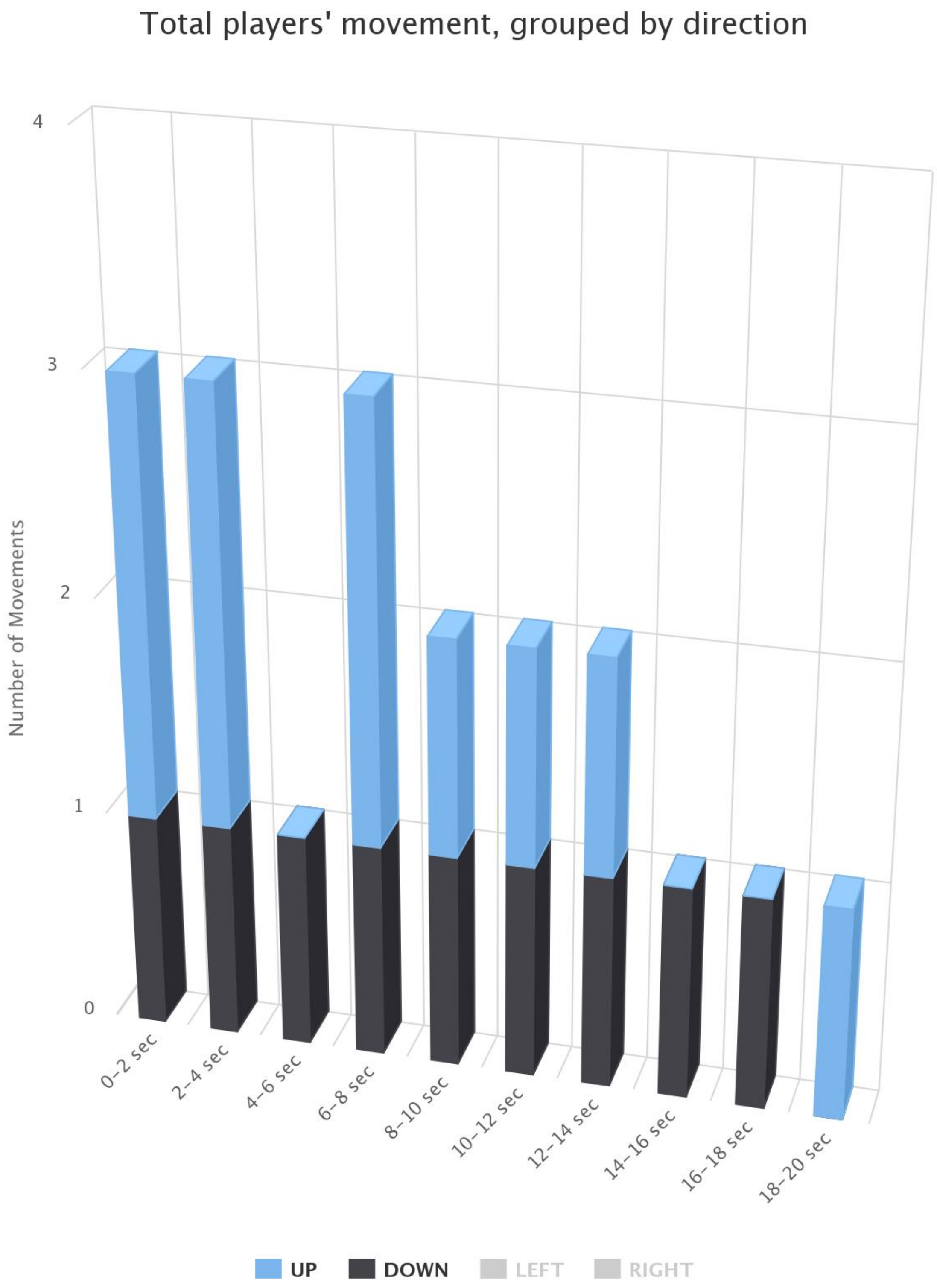


Figure 5.2 3D Column Visualization of Pitch Head Pose Estimation

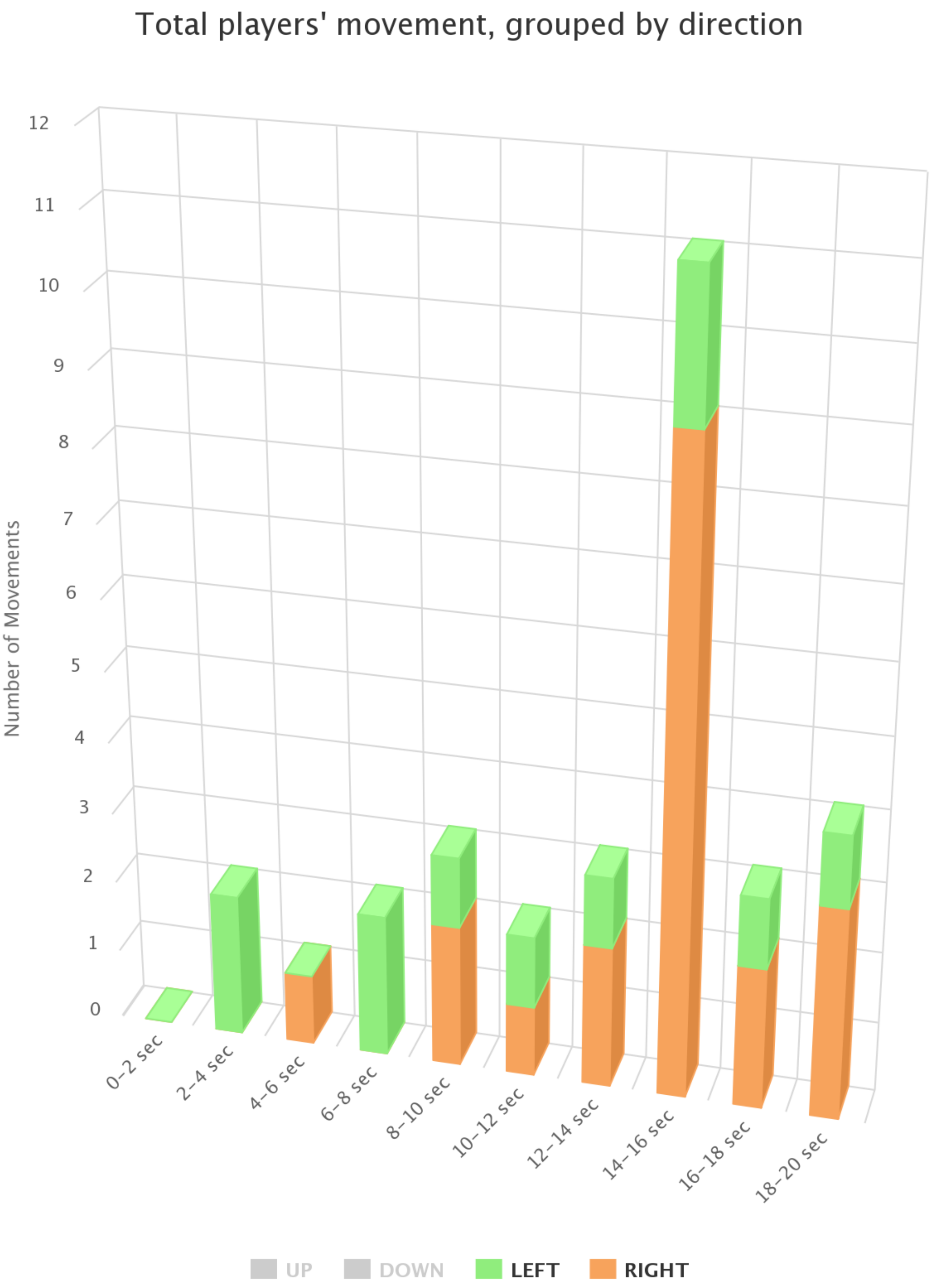


Figure 5.3 3D Column Visualization of Yaw Head Pose Estimation

Total Players' Movement Web Visualization

The visualization regarding the recognized emotions via facial expressions is built in the same fashion as the previous one. In this case, facial expressions are grouped by the recognized emotions. The four different emotions are represented by four different colors. In one hand, x axis represents the time scale which is divided every two seconds until the end of the test. On the other hand, y axis displays the number of recognized emotions for all the users that take part in the tests. Furthermore, when hovering above a column, the number of the corresponding emotion summary is displayed. As seen in Figure 5.4 the most dominant emotion in time intervals of 0-2, 4-6 and 18-20 is anger, which can lead to the conclusion that those serious game players' are feeling in such a way because of their low score or below par achievements. Must be noted that the exactly opposite emotions such as happiness and sadness are displayed in the same bar, while different bars are used for the other two emotions as seen in Figure 5.5 and Figure 5.6 respectively. The full version of the web-based visualization can be found at http://83.212.117.19/3D_FER_MSCV/.

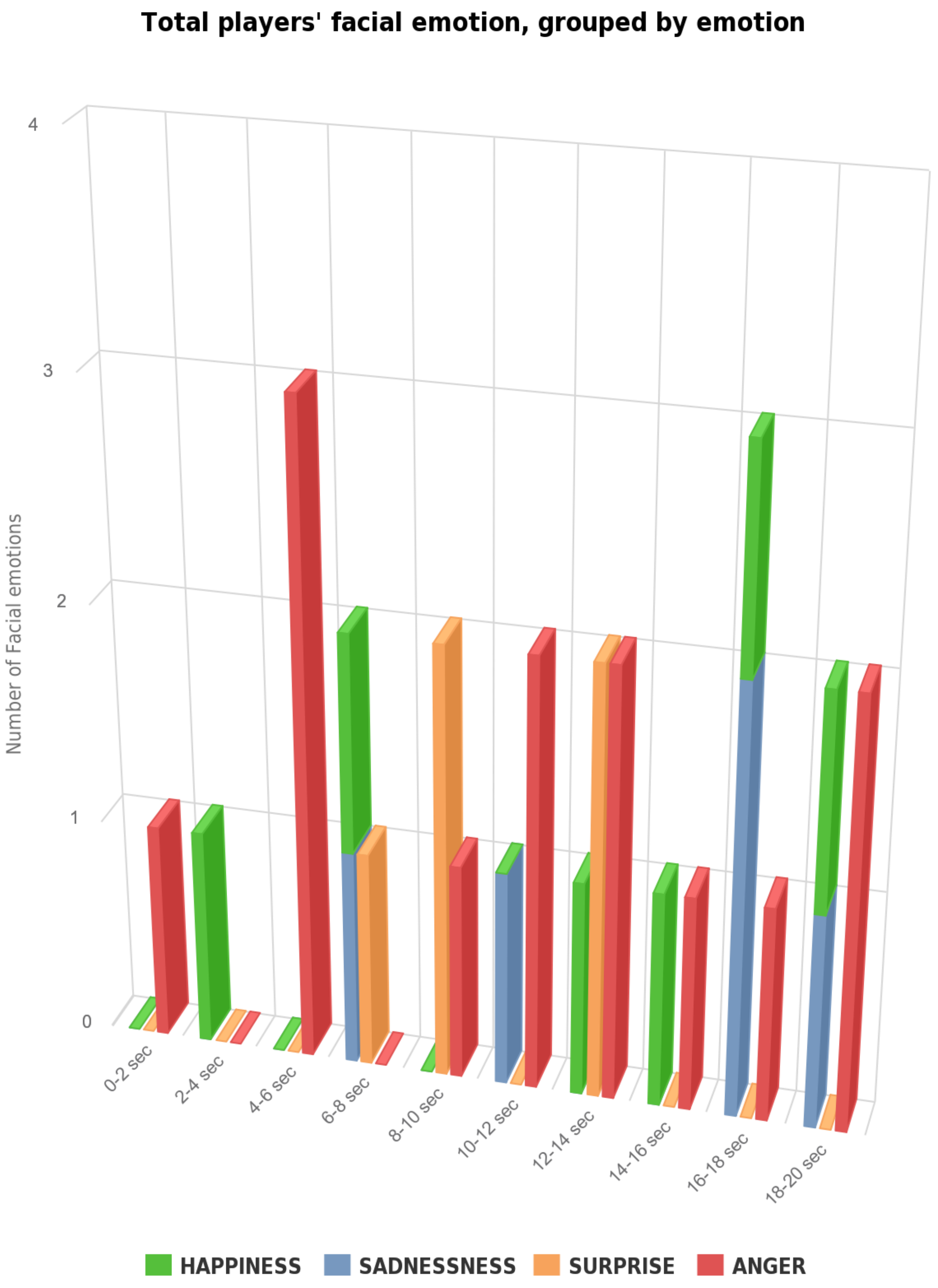


Figure 5.4 3D Column Visualization of Total FER

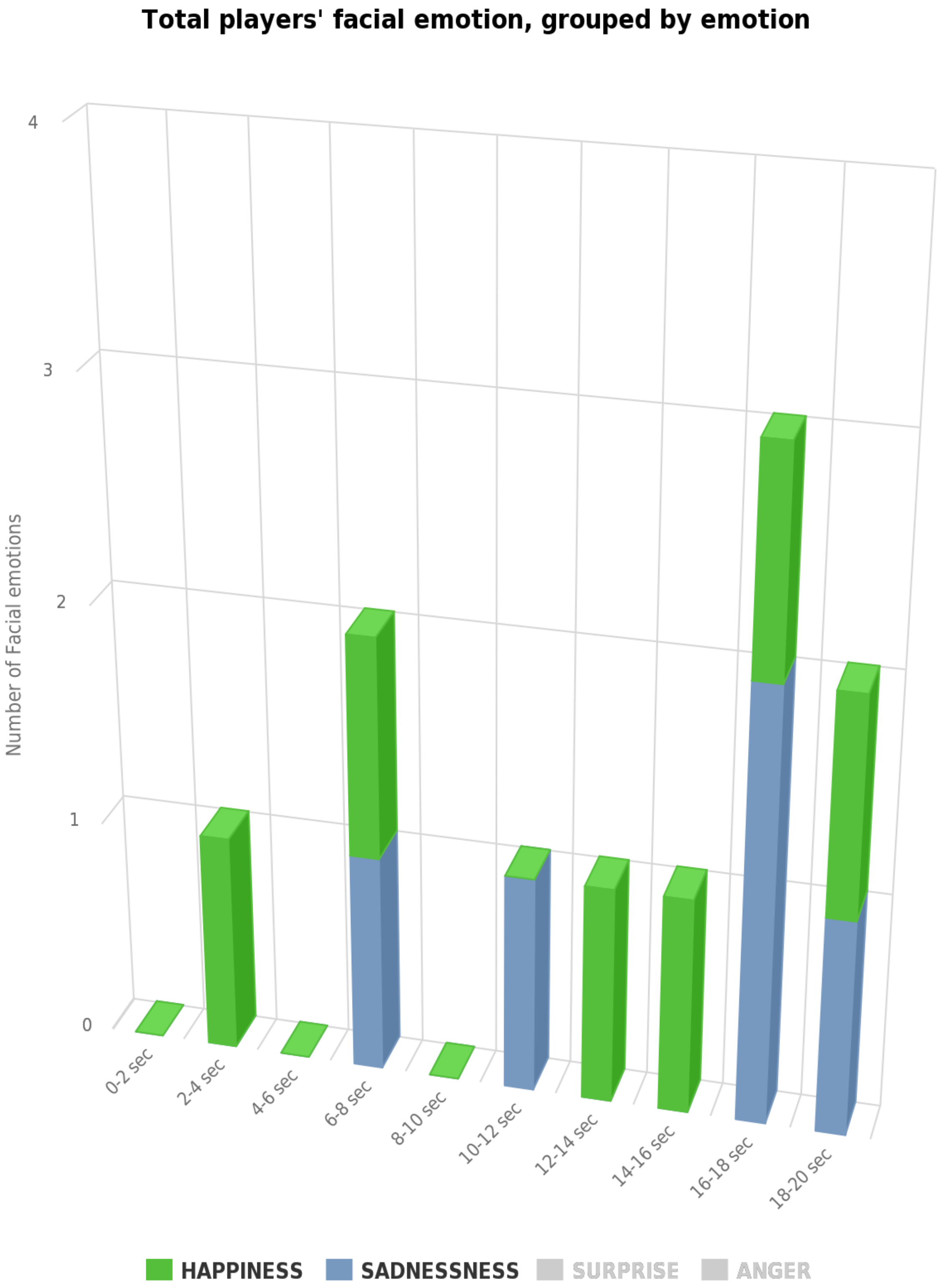


Figure 5.5 3D Column Visualization of facial expressions "HAPINESS" and "SADNESS"

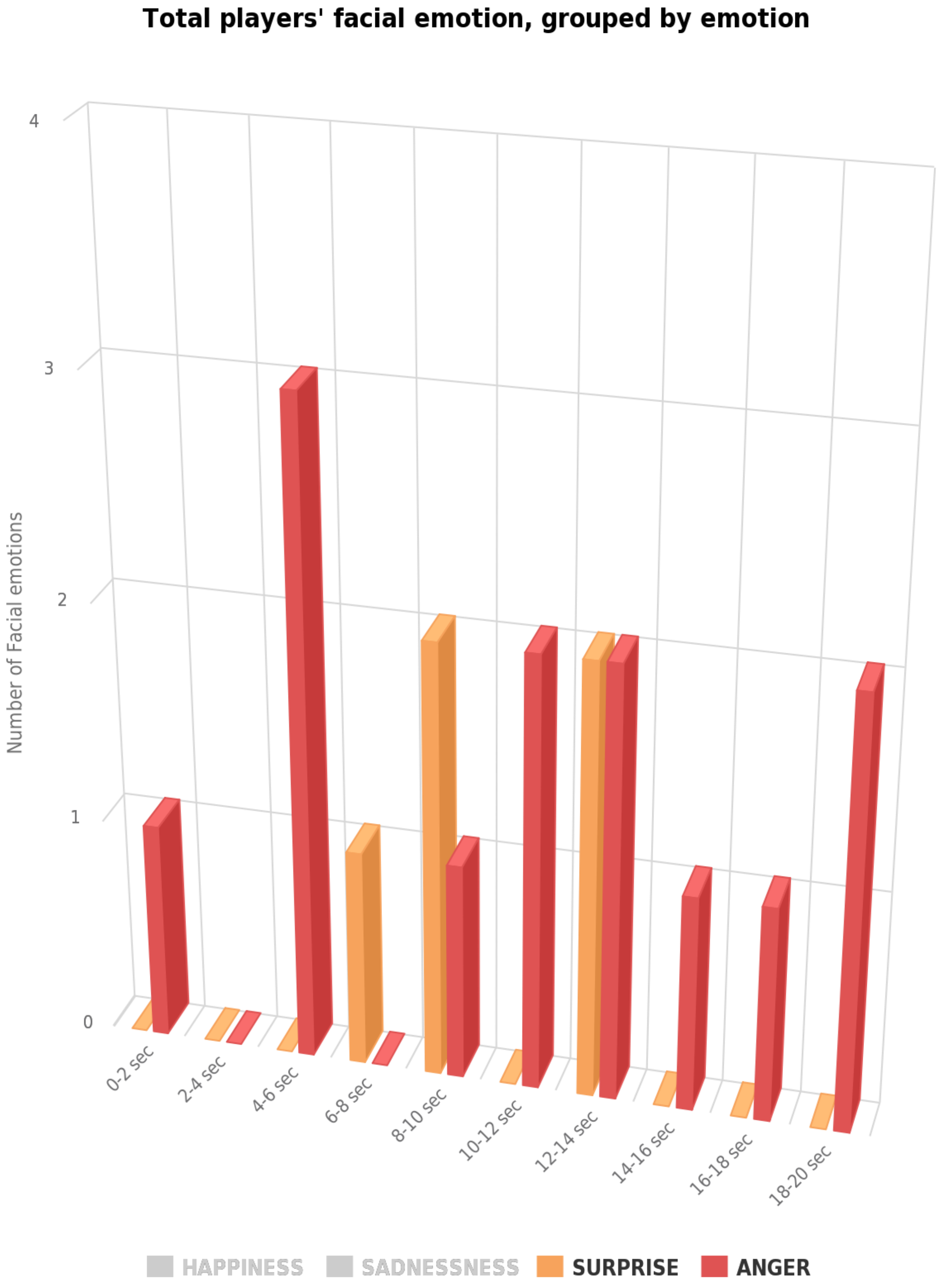


Figure 5.6 3D Column Visualization of facial expressions "SURPRISE" and "ANGER"

Chapter 6 - Conclusions and Future Work

This thesis introduces a promising method for efficiently aggregating behavioral and physiological metrics of user experience regarding head pose estimation and facial expression recognition from depth data. After a lengthy overview of state-of-the-art techniques on both scientific fields, Discriminative Random Regression Forests and features of animation units were cautiously chosen for head pose estimation and facial expression recognition respectively, primarily because of their documented superiority against other methods of the literature in terms of real-time performance capacity and clarity of implementation process. In the matter of storing the metrics data from the two sets, a lightweight data exchange format was exploited. After that web-based visualizations were made for objective and accessible assessment of user actions within the context of different applications. As an illustration following such an approach, the proposed system will be able to estimate users experience, mood and emotion during serious games sessions. Furthermore, these data can be used afterwards for evaluating players' educational, training and ludology experience and affecting the context of the serious game, acting as a feedback mechanism.

Although an encouraging method is proposed in this thesis for monitoring and assessing behavioral and physiological metrics of user experience, we believe our work will inspire more ideas on this specific area of objective evaluation. More efforts are needed in order to improve the overall accuracy of the system. In more details, results relevant to head pose estimation are produced with high accuracy. However facial expression outcome is not as satisfactory as we would have liked. Therefore in order to implement a complete system for monitoring those metrics, a fusion of the aforementioned methods is required. We look to extend our work in the following ways:

- Real-time connection between the two frameworks
- Fusion of the two frameworks by utilizing Bayes Rule, so a combination of the high accurate head pose estimation will improve facial expression framework's performance as well
- Larger database creation
- FER framework adaptation in order to include the absent two expressions, fear and disgust

Chapter 7 - Bibliography

[1] E. M. Chutorian and M. M. Trivedi, "Head Pose Estimation in Computer Vision: A Survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* pp. 607-626, 2009.

[2] T. Fang, Z. X., O. Ocegueda, S. Shah and I. Kakadiaris, "3D facial expression recognition: A perspective on promises and challenges," in *Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, Santa Barbara, CA, IEEE, 2011, pp. 603-610.

[3] M. Demirkus, B. Oreshkin, J. J. Clark and T. Arbel, "Spatial and probabilistic codebook template based head pose estimation from unconstrained environments," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, Brussels, IEEE, 2011, pp. 573-576.

[4] D. Kim, J. Park and A. C. Kak, "Estimating head pose with an RGBD sensor: A comparison of appearance-based and pose-based local subspace methods," in *Image Processing (ICIP), 2013 20th IEEE International Conference on*, Melbourne, VIC, IEEE, 2013, pp. 3637-3641.

[5] N. Alioua, A. Amine, M. Rziza, A. Bensrhair and D. Aboutajdine, "Head pose estimation based on steerable filters and likelihood parametrized function," in *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European*, Marrakech, IEEE, 2013, pp. 1-5.

[6] Y. Li, S. Wang and X. Ding, "Person-independent head pose estimation based on random forest regression," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, Hong Kong, IEEE, 2010, pp. 1521-1524.

[7] Q. Zhao, W. Xu, Y. Wang and X. Shi, "Using head poses to control a virtual robot walking

in a virtualmaze," *Optics Communications,* vol. 295, pp. 84-91, 2013.

[8] M. Ratsch, P. Quick, P. Huber, T. Frank and T. Vetter, "Wavelet Reduced Support Vector Regression for Efficient and Robust Head Pose Estimation," in *Computer and Robot Vision (CRV), 2012 Ninth Conference on,* Toronto, ON, IEEE, 2012, pp. 260-267.

[9] N. Meins, S. Magg and S. Wermter, "Neural Hopfield-ensemble for multi-class head pose detection," in *Neural Networks (IJCNN), The 2013 International Joint Conference on,* Dallas, IEEE, 2013, pp. 1-8.

[10] Q. Wang, Y. Wu, Y. Shen, Y. Liu and Y. Lei, "Supervised sparse manifold regression for head pose estimation in 3D space," *Signal Processing,* vol. 112, p. 34–42, 2014.

[11] C. Wang and X. Song, "Robust head pose estimation via supervised manifold learning," *Neural Networks,* vol. 53, pp. 15-25, 2014.

[12] D. Huang, M. Storer, F. De la Torre and H. Bischof, "upervised Local Subspace Learning for Continuous Head Pose Estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* Providence, IEEE , 2011, pp. 2921 - 2928.

[13] N. M. Bidgoli, A. A. Raie and M. Naraghi, "Probabilistic principal component analysis for texture modelling of adaptive active appearance models and its application for head pose estimation," *Computer Vision, IET,* vol. 9, no. 1, pp. 51-62, 2015.

[14] B. Ma, X. Chai and T. Wang, "A novel feature descriptor based on biologically inspired feature for head pose estimation," *Neurocomputing,* vol. 115, pp. 1-10, 2012.

[15] Y. Tang, Z. Sun and T. Tan, "Slice representation of range data for head pose estimation," *Computer Vision and Image Understanding,* vol. 128, no. null, pp. 18-35, 2014.

[16] S. Gurbuz, E. Oztop and N. Inoue, "Model free head pose estimation using stereovision," *Pattern Recognition,,* vol. 45, no. 1, pp. 33-42, 2012.

[17] Y. Xu, J. Zeng and Y. Sun, "Head Pose Recovery Using 3D Cross Model," in *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference on*, Nanchang, Jiangxi, IEEE, 2012, pp. 63-66.

[18] T. Siriteerakul, Y. Sato and V. Boonjing, "Estimating change in head pose from low resolution video using LBP-based tracking," in *Intelligent Signal Processing and Communications Systems (ISPACS), 2011 International Symposium on*, Chiang Mai, IEEE, 2011, pp. 1-6.

[19] G. Fanelli, T. Weise, J. Gall and L. Van Gool, "Real Time Head Pose Estimation from Consumer Depth Cameras," in *Pattern Recognition*, Frankfurt/Main, Germany, Springer Berlin Heidelberg, 2011, pp. 101-110.

[20] G. Fanelli, J. Gall and L. Van Gool, "Real time head pose estimation with random regression forests," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, IEEE, 2011, pp. 617-624.

[21] T.-z. Qiao and S.-l. Dai, "Depth data filtering for real-time head pose estimation with Kinect," in *Image and Signal Processing (CISP), 2013 6th International Congress on}*, Hangzhou, IEEE, 2013, pp. 953-958.

[22] G. Fanelli, J. Gall and L. Van Gool, "Real time 3d head pose estimation: recent achievements and future challenges," in *Communications Control and Signal Processing (ISCCSP), 2012 5th International Symposium on*, Rome, IEEE, 2012, pp. 1-4.

[23] G. Fanelli, M. Dantone, J. Gall, A. Fossati and L. Gool, "Random forests for real time 3D face analysis," *International Journal of Computer Vision*, vol. 101 , no. 3, pp. 437-458 , 2013 .

[24] M. Xue, A. Mian, W. Liu and L. Li, "Fully automatic 3D facial expression recognition using local depth features," in *Applications of Computer Vision (WACV), 2014 IEEE*

Winter Conference on, Steamboat Springs, CO, IEEE, 2014, pp. 1096-1103.

[25] A. Azazi, S. L. Lutfi and I. Venkat, "Analysis and evaluation of SURF descriptors for automatic 3D facial expression recognition using different classifiers," in *Information and Communication Technologies (WICT), 2014 Fourth World Congress on*, Bandar Hilir, IEEE, 2014, pp. 23-28.

[26] P. Lemaire, M. Ardabilian, L. Chen and M. Daoudi, "Fully automatic 3D facial expression recognition using differential mean curvature maps and histograms of oriented gradients," in *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, Shanghai, IEEE, 2013, pp. 1-7.

[27] S. Berretti, A. D. Bimbo, P. Pala, B. B. Amor and M. Daoudi, "A Set of Selected SIFT Features for 3D Facial Expression Recognition," in *Pattern Recognition (ICPR), 2010 20th International Conference on*, Istanbul, IEEE, 2010, pp. 4125-4128.

[28] Y. Tie and L. Guan, "Human emotion recognition using a deformable 3D facial expression model," in *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on*, Seoul, Korea (South), IEEE, 2012, pp. 1115-1118.

[29] M. Rosato, X. Chen and L. Yin, "Automatic Registration of Vertex Correspondences for 3D Facial Expression Analysis," in *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on*, Arlington, VA, IEEE, 2008, pp. 1-7.

[30] I. Mpiperis, S. Malassiotis and M. G. Strintzis, "Bilinear Models for 3-D Face and Facial Expression Recognition," *Information Forensics and Security, IEEE Transactions on*, vol. 3, no. 3, pp. 498-511, 2008.

[31] T. Fang, X. Zhao, O. Ocegueda, S. K. Shah and I. A. Kakadiaris, "3D/4D Facial Expression Analysis: An Advanced Annotated Face Model Approach," *Image and Vision Computing*, vol. 30, no. 10, pp. 738--749, 2012.

[32] Y.-l. Tian, T. Kanade and J. F. Cohn, "Recognizing Action Units for Facial Expression Analysis," *IEEE Trans. Pattern Anal. Mach. Intell.,* pp. 97--115, 2001.

[33] P. Ekman and W. Friesen, "Facial action coding system: a technique for the measurement of facial movement," 1978.

[34] N. U. Powar, J. D. Foytik and V. K. Asari, "Facial Expression Analysis using 2D and 3D Features," in *Aerospace and Electronics Conference (NAECON), Proceedings of the 2011 IEEE National*, Dayton, IEEE, 2011, pp. 73-78.

[35] Q.-r. Mao, X.-y. Pan, Y.-z. Zhan and X.-j. Shen, "UsingKinect for real-time emotion recognition via facial expressions," *Frontiers of Information Technology & Electronic Engineering,* vol. 16, no. 4, pp. 272-282, 2015.

[36] Microsoft, "Kinect SDK Documentation – Face Tracking," [Online]. Available: http://msdn.microsoft.com/enus/. [Accessed April 2015].

[37] J. Ahlberg, "Candide – A parameterized face," [Online]. Available: http://www.icg.isy.liu.se/candide/. [Accessed April 2015].

[38] B. Lin , Y. Chen , X. Chen and Y. Yu, "Comparison between JSON and XML in Applications Based on AJAX," in *Computer Science Service System (CSSS), 2012 International Conference on*, Nanjing, IEEE, 2012, pp. 1174-1177.

[39] D. M. Powers, "Evaluation Evaluation a Monte Carlo study," *European Conference on Artificial Intelligence,* 2015.

[40] M. Bostock, V. Ogievetsky and J. Heer, "D3: Data-Driven Documents," in *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011.

[41] "jQuery," [Online]. Available: http://jquery.com/. [Accessed April 2015].

[42] Kuan and Joe, "Learning Highcharts," 2012.

[43] T. Fang, X. Zhao, O. Ocegueda, S. Shah and . I. Kakadiaris, "3D facial expression recognition: A perspective on promises and challenges," in *Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, Santa Barbara, CA, IEEE, 2011, pp. 603-610.

Appendix A - Head Pose Estimation Experimental Results

```
[
  {
    "SessionDate":"08/May/15",
    "SessionData":[
      {
        "time":6.67,
        "direction":"UP",
        "intensity":9.38992
      },
      {
        "time":9.99,
        "direction":"RIGHT",
        "intensity":6.64818
      },
      {
        "time":14.27,
        "direction":"RIGHT",
        "intensity":5.72149
      },
      {
        "time":14.84,
        "direction":"LEFT",
        "intensity":5.78074
      },
      {
         "time":17.98,
         "direction":"LEFT",
         "intensity":5.23937
      }
   ]
},
{
   "SessionDate":"08/May/15",
   "SessionData":[
      {
         "time":3.94,
         "direction":"LEFT",
         "intensity":6.74131
      },
      {
         "time":7.79,
         "direction":"UP",
         "intensity":8.06202
      },
      {
         "time":14.93,
         "direction":"RIGHT",
         "intensity":7.99714
      },
      {
         "time":15.88,
         "direction":"LEFT",
         "intensity":5.9314
      },
      {
         "time":18.56,
         "direction":"RIGHT",
         "intensity":14.4218
      }
   ]
```

```
},
{
   "SessionDate":"08/May/15",
   "SessionData":[
      {
         "time":9.69,
         "direction":"LEFT",
         "intensity":5.97083
      },
      {
         "time":14.46,
         "direction":"RIGHT",
         "intensity":5.66509
      },
      {
         "time":14.64,
         "direction":"RIGHT",
         "intensity":6.26326
      }
   ]
},
{
   "SessionDate":"08/May/15",
   "SessionData":[
      {
         "time":7.88,
         "direction":"DOWN",
         "intensity":5.27548
      },
      {
         "time":11.02,
         "direction":"RIGHT",
         "intensity":11.4221
      },
      {
         "time":14.64,
         "direction":"RIGHT",
          "intensity":6.42895
        }
      ]
    },
    {
      "SessionDate":"08/May/15",
      "SessionData":[
        {
          "time":0.47,
          "direction":"UP",
          "intensity":6.53054
        },
        {
          "time":3.35,
          "direction":"LEFT",
          "intensity":6.55545
        },
        {
          "time":5.46,
          "direction":"RIGHT",
          "intensity":8.7874
        },
        {
          "time":8.65,
          "direction":"UP",
          "intensity":5.00297
        },
        {
          "time":9.22,
          "direction":"DOWN",
          "intensity":6.41506
        },
        {
          "time":12.54,
          "direction":"RIGHT",
          "intensity":8.82215
        },
            {
               "time":15.33,
               "direction":"RIGHT",
               "intensity":5.79061
            }
         ]
      },
      {
         "SessionDate":"19/May/15",
         "SessionData":[
            {
               "time":0.88,
               "direction":"DOWN",
               "intensity":6.51841
            },
            {
               "time":5.77,
               "direction":"DOWN",
               "intensity":9.76992
            },
            {
               "time":14.27,
               "direction":"RIGHT",
               "intensity":5.72149
            },
            {
               "time":16.08,
               "direction":"RIGHT",
               "intensity":5.77937
            }
         ]
      },
      {
         "SessionDate":"19/May/15",
         "SessionData":[
            {
               "time":3.34,
            "direction":"UP",
            "intensity":6.89431
          },
          {
            "time":6.29,
            "direction":"LEFT",
            "intensity":8.26002
          },
          {
            "time":8.91,
            "direction":"RIGHT",
            "intensity":7.77465
          },
          {
            "time":12.08,
            "direction":"DOWN",
            "intensity":5.6139
          },
          {
            "time":15.65,
            "direction":"DOWN",
            "intensity":14.8211
          },
          {
            "time":18.05,
            "direction":"RIGHT",
            "intensity":11.6789
          }
        ]
      },
      {
        "SessionDate":"19/May/15",
        "SessionData":[
          {
            "time":6.69,
            "direction":"LEFT",
            "intensity":5.8874
```

```
         },
         {
            "time":10.86,
            "direction":"LEFT",
            "intensity":5.9656
         },
         {
            "time":12.03,
            "direction":"UP",
            "intensity":6.8905
         },
         {
            "time":15.64,
            "direction":"RIGHT",
            "intensity":6.45326
         },
         {
            "time":18.96,
            "direction":"RIGHT",
            "intensity":12.45654
         }
      ]
   },
   {
      "SessionDate":"19/May/15",
      "SessionData":[
         {
            "time":3.98,
            "direction":"DOWN",
            "intensity":9.43248
         },
         {
            "time":11.52,
            "direction":"UP",
            "intensity":11.2321
         },
         {
```

```
         "time":13.02,
         "direction":"RIGHT",
         "intensity":11.4221
      },
      {
         "time":16.64,
         "direction":"RIGHT",
         "intensity":8.769
      },
      {
         "time":19.84,
         "direction":"LEFT",
         "intensity":6.4321
      }
   ]
},
{
   "SessionDate":"19/May/15",
   "SessionData":[
      {
         "time":3.47,
         "direction":"UP",
         "intensity":6.53054
      },
      {
         "time":10.35,
         "direction":"DOWN",
         "intensity":7.4568
      },
      {
         "time":12.06,
         "direction":"LEFT",
         "intensity":8.0094
      },
      {
         "time":14.56,
         "direction":"RIGHT",
        "intensity":6.00297
      },
      {
        "time":16.11,
        "direction":"DOWN",
        "intensity":6.51403
      },
      {
        "time":18.14,
        "direction":"UP",
        "intensity":12.82215
      }
    ]
  }
]
```

Appendix B - FER Experimental Results

```
[
  {
    "SessionDate": "11/05/15",
    "SessionData": [
      {
        "time": 9.94,
        "emotion": "HAPPINESS"
      },
      {
        "time": 11.22,
        "emotion": "HAPPINESS"
      },
      {
        "time": 13.76,
        "emotion": "SADNESS"
      },
      {
        "time": 18.1,
        "emotion": "HAPPINESS"
      }
    ]
  },
  {
    "SessionDate": "12/05/15",
    "SessionData": [
      {
        "time": 2.12,
        "emotion": "SURPRISE"
      },
      {
        "time": 4.66,
        "emotion": "HAPPINESS"
      },
      {
        "time": 10.06,
      "emotion": "ANGER"
    },
    {
      "time": 18.76,
      "emotion": "HAPPINESS"
    }
  ]
},
{
  "SessionDate": "13/05/15",
  "SessionData": [
    {
      "time": 14.33,
      "emotion": "SADNSESS"
    }
  ]
},
{
  "SessionDate": "14/05/15",
  "SessionData": [
    {
      "time": 1.98,
      "emotion": "HAPPINESS"
    },
    {
      "time": 5.03,
      "emotion": "SADNSESS"
    },
    {
      "time": 12.12,
      "emotion": "HAPPINESS"
    },
    {
      "time": 14.17,
      "emotion": "SURPRISE"
    },
    {
```

```
   "time": 16.83,
   "emotion": "ANGER"
  }
 ]
},
{
 "SessionDate": "15/05/15",
 "SessionData": [
  {
   "time": 6.22,
   "emotion": "SADNSESS"
  },
  {
   "time": 7.13,
   "emotion": "ANGER"
  }
 ]
},
{
 "SessionDate": "16/05/15",
 "SessionData": [
  {
   "time": 5.08,
   "emotion": "HAPPINESS"
  },
  {
   "time": 7.11,
   "emotion": "SURPRISE"
  },
  {
   "time": 9.9,
   "emotion": "SADNESS"
  },
  {
   "time": 13.27,
   "emotion": "SADNESS"
  },
        {
          "time": 16.93,
          "emotion": "ANGER"
        },
        {
          "time": 19.93,
          "emotion": "ANGER"
        }
      ]
    },
    {
      "SessionDate": "17/05/15",
      "SessionData": [
        {
          "time": 6.28,
          "emotion": "SADNESS"
        },
        {
          "time": 16.08,
          "emotion": "HAPPINESS"
        }
      ]
    },
    {
      "SessionDate": "18/05/15",
      "SessionData": [
        {
          "time": 9.12,
          "emotion": "SADNESS"
        }
      ]
    },
    {
      "SessionDate": "19/05/15",
      "SessionData": [
        {
          "time": 12.12,
        "emotion": "HAPPINESS"
      },
      {
        "time": 16.28,
        "emotion": "SURPRISE"
      },
      {
        "time": 19.2,
        "emotion": "SURPRISE"
      }
    ]
  },
  {
    "SessionDate": "20/05/15",
    "SessionData": [
      {
        "time": 5.66,
        "emotion": "HAPPINESS"
      },
      {
        "time": 10.38,
        "emotion": "HAPPINESS"
      },
      {
        "time": 12.9,
        "emotion": "SURPRISE"
      },
      {
        "time": 15.03,
        "emotion": "HAPPINESS"
      }
    ]
  }
]
```